SIMUL8

Manual and Simulation Guide
To install **SIMUL8** on your computer please see the installation guide at the back of this document (page 227).

To get started we recommend you read the section “Building your first **SIMUL8** model”

**Visual Thinking’s mission** is to enable Visual Interactive Modeling to be used by anyone who is making decisions, or thinking, about improvements in any system they manage. In pursuit of this we are creating and marketing very high quality software products which are easy to use, very carefully researched, and priced for widespread use.
Contact Information:

Worldwide Support:
Email: info@VisualT.com

Use the interactive web based support conference: http://www.VisualT.com/conf/
This gives you direct access to a library of previous questions and answers direct from the support team - and also direct from other users. The conference is completely open - you can discuss all aspect of simulation (and SIMUL8) with the support team and other users.

USA and Canada:
Visual Thinking International Limited
55A Port Street
Mississauga, Ontario
Canada L5H 4P3
Tel: +1 800 878 3373
Fax: +1 905 279 1662

Japan:
Design Technologies Inc.
Towa Nihonbashi-Horidome Building,
3-15 1 Chome Nihonbashi- Horidomecho, Chuo-KU,
Tokyo, Japan.
Tel: +81 3 3667 1110
Fax: +81 3 3668 3100

World Wide:
Visual Thinking International Limited
141 St James Road, Glasgow, UK G4 0LT
Tel: +44 141 552 6888
Fax: +44 141 552 3886
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>13</td>
</tr>
<tr>
<td>SIMULATION - INTRODUCTION</td>
<td>13</td>
</tr>
<tr>
<td>Definition</td>
<td>14</td>
</tr>
<tr>
<td>SIMUL8 - INTRODUCTION</td>
<td>15</td>
</tr>
<tr>
<td>The Process</td>
<td>16</td>
</tr>
<tr>
<td>BUILDING YOUR FIRST SIMUL8 MODEL</td>
<td>19</td>
</tr>
<tr>
<td>OVERVIEW OF USING SIMUL8 FOR BUILDING LARGE MODELS</td>
<td>25</td>
</tr>
<tr>
<td>Main Concepts</td>
<td>25</td>
</tr>
<tr>
<td>Display</td>
<td>26</td>
</tr>
<tr>
<td>Tailored Interactive Aspects</td>
<td>27</td>
</tr>
<tr>
<td>Models with similar parts</td>
<td>27</td>
</tr>
<tr>
<td>Models within models (Sub-Simulation Windows)</td>
<td>28</td>
</tr>
<tr>
<td>Model Structure</td>
<td>28</td>
</tr>
<tr>
<td>External Data</td>
<td>28</td>
</tr>
<tr>
<td>Zoom</td>
<td>29</td>
</tr>
<tr>
<td>Business Process Re-engineering</td>
<td>29</td>
</tr>
<tr>
<td>Documenting Models</td>
<td>29</td>
</tr>
<tr>
<td>Visual Logic</td>
<td>29</td>
</tr>
<tr>
<td>Information Store</td>
<td>30</td>
</tr>
<tr>
<td>What Next</td>
<td>30</td>
</tr>
<tr>
<td>SIMULATION: THE TECHNIQUE - A GUIDE</td>
<td>33</td>
</tr>
<tr>
<td>(a) Listen</td>
<td>33</td>
</tr>
<tr>
<td>(b) Level of Detail?</td>
<td>34</td>
</tr>
<tr>
<td>(c) How Broad?</td>
<td>34</td>
</tr>
<tr>
<td>(d) Build Rough &amp; Ready Model, then Refine it</td>
<td>34</td>
</tr>
<tr>
<td>(e) Work Closely with your Client</td>
<td>35</td>
</tr>
<tr>
<td>(f) Running your Model</td>
<td>36</td>
</tr>
<tr>
<td>(g) Verification</td>
<td>36</td>
</tr>
<tr>
<td>(h) Warm Up Time</td>
<td>37</td>
</tr>
<tr>
<td>(i) Visual Exploration</td>
<td>40</td>
</tr>
<tr>
<td>(j) Accuracy of Results</td>
<td>41</td>
</tr>
<tr>
<td>(k) How much of this do you need?</td>
<td>49</td>
</tr>
<tr>
<td>Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>REFERENCE SECTION - THE SIMUL8 FACILITIES</td>
<td>51</td>
</tr>
<tr>
<td>The Main Concepts</td>
<td>51</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Structure of your model</td>
<td>51</td>
</tr>
<tr>
<td>Important Terminology</td>
<td>52</td>
</tr>
<tr>
<td>Simulation Window</td>
<td>52</td>
</tr>
<tr>
<td>Simulation Objects</td>
<td>53</td>
</tr>
<tr>
<td>Work Items</td>
<td>54</td>
</tr>
<tr>
<td>Important Images on the Simulation Window</td>
<td>54</td>
</tr>
<tr>
<td>Work Centers</td>
<td>57</td>
</tr>
<tr>
<td>Storage Bins</td>
<td>59</td>
</tr>
<tr>
<td>Conveyors</td>
<td>61</td>
</tr>
<tr>
<td>Work Entry Points</td>
<td>61</td>
</tr>
<tr>
<td>Work Exit Points</td>
<td>62</td>
</tr>
<tr>
<td>Resources</td>
<td>63</td>
</tr>
<tr>
<td>Resource Shift Patterns</td>
<td>63</td>
</tr>
<tr>
<td>Pooled Resources</td>
<td>64</td>
</tr>
<tr>
<td>Multiple Resources at a Work Center</td>
<td>64</td>
</tr>
<tr>
<td>Converting Resources into Different Types</td>
<td>65</td>
</tr>
<tr>
<td>When are resources released?</td>
<td>65</td>
</tr>
<tr>
<td>When are resources required?</td>
<td>66</td>
</tr>
<tr>
<td>Invisible objects</td>
<td>66</td>
</tr>
<tr>
<td>Object List Dialog Box</td>
<td>66</td>
</tr>
<tr>
<td>Deleting Objects</td>
<td>67</td>
</tr>
<tr>
<td>Quick Replicate</td>
<td>68</td>
</tr>
<tr>
<td>Hierarchical Modeling</td>
<td>69</td>
</tr>
<tr>
<td>Sub-Simulation Windows</td>
<td>69</td>
</tr>
<tr>
<td>Removing Sub-Simulation Windows</td>
<td>70</td>
</tr>
<tr>
<td>Closing a sub-window</td>
<td>70</td>
</tr>
<tr>
<td>Important Control Features</td>
<td>72</td>
</tr>
<tr>
<td>Select, Move and Copy Objects on Screen</td>
<td>72</td>
</tr>
<tr>
<td>Move</td>
<td>73</td>
</tr>
<tr>
<td>Copy</td>
<td>73</td>
</tr>
<tr>
<td>Aligning Objects on the display</td>
<td>75</td>
</tr>
<tr>
<td>UNDO</td>
<td>75</td>
</tr>
<tr>
<td>Redo</td>
<td>76</td>
</tr>
<tr>
<td>Enable/Disable UNDO</td>
<td>76</td>
</tr>
<tr>
<td>Find</td>
<td>76</td>
</tr>
<tr>
<td>Preferences</td>
<td>77</td>
</tr>
<tr>
<td>The Simulation Time</td>
<td>78</td>
</tr>
<tr>
<td>Clock</td>
<td>78</td>
</tr>
<tr>
<td>Results Collection Period</td>
<td>78</td>
</tr>
<tr>
<td>Warm Up Period</td>
<td>78</td>
</tr>
<tr>
<td>High Speed Clock</td>
<td>79</td>
</tr>
<tr>
<td>Routing</td>
<td>80</td>
</tr>
</tbody>
</table>
Drawing the Routing ................................................................. 80
Deleting a route arrow ................................................................. 81
Switch Route Lines on/off .............................................................. 81
Routing In to a Work Center .......................................................... 82
Priority in determining the Work Item to select for a Work Center .... 82
Collect work items ........................................................................ 82
Expired Only .................................................................................. 83
Shelf Life ....................................................................................... 83
Interruptible Work Centers............................................................. 84
Attribute batching ......................................................................... 84
Batching by type ............................................................................ 85
Routing Out of a Work Center ......................................................... 85
Changing the work item during a task at a work center ................. 86
List of Destinations ......................................................................... 86
Circulate Routing Rule ................................................................. 87
Priority Routing (First Destination = First Choice) ....................... 87
Uniform Routing Rule .................................................................. 87
Percent Routing Rule .................................................................... 87
Percentage Automatically = 100 .................................................... 87
Shortest Queue ............................................................................. 88
Preference Routing ........................................................................ 88
TRAVELING TIME ...................................................................... 89
ATTRIBUTES .............................................................................. 91
Adding attributes to Work Items .................................................. 91
Attributes Containing Numbers .................................................... 92
Attributes Containing Text ............................................................ 92
Attributes List ............................................................................... 92
Attribute Actions .......................................................................... 92
List of Action Attributes .............................................................. 93
Add to the list of Action Attributes................................................ 93
Remove Action .............................................................................. 94
Attribute Value .............................................................................. 94
Attribute Text - Setting ............................................................... 94
Attributes control Work Item Images ............................................ 95
Special Attributes .......................................................................... 96
Priority attribute ........................................................................... 96
Due attribute .................................................................................. 96
Quantity Attribute ......................................................................... 96
ROUTING USING ATTRIBUTES .................................................... 96
ANIMATION ................................................................................ 98
PRIORITY ...................................................................................... 100
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prioritizing Work</td>
<td>100</td>
</tr>
<tr>
<td>Prioritizing Work Centers</td>
<td>100</td>
</tr>
<tr>
<td><strong>Efficiency and Breakdowns</strong></td>
<td>102</td>
</tr>
<tr>
<td>Distributions used for Efficiency</td>
<td>102</td>
</tr>
<tr>
<td>Detail / Auto Efficiency</td>
<td>102</td>
</tr>
<tr>
<td>Stoppages - special cases</td>
<td>102</td>
</tr>
<tr>
<td><strong>Random Numbers</strong></td>
<td>104</td>
</tr>
<tr>
<td>Where random numbers are used</td>
<td>104</td>
</tr>
<tr>
<td>Auto Change</td>
<td>104</td>
</tr>
<tr>
<td>Antithetic</td>
<td>105</td>
</tr>
<tr>
<td>Saving Random Number Streams</td>
<td>105</td>
</tr>
<tr>
<td><strong>Timing and Distributions</strong></td>
<td>106</td>
</tr>
<tr>
<td>Distribution Parameters</td>
<td>106</td>
</tr>
<tr>
<td>Distribution Type</td>
<td>107</td>
</tr>
<tr>
<td>New Distribution</td>
<td>107</td>
</tr>
<tr>
<td>Distribution Detail</td>
<td>107</td>
</tr>
<tr>
<td>Standard Distributions</td>
<td>108</td>
</tr>
<tr>
<td>Fixed Distribution</td>
<td>109</td>
</tr>
<tr>
<td>Uniform Distributions</td>
<td>109</td>
</tr>
<tr>
<td>Named Distributions</td>
<td>109</td>
</tr>
<tr>
<td>Ask VBA Check Box</td>
<td>110</td>
</tr>
<tr>
<td>Time Dependent Distributions</td>
<td>110</td>
</tr>
<tr>
<td>Re-Sampling Ovrun's between time slots</td>
<td>111</td>
</tr>
<tr>
<td>External Distributions</td>
<td>112</td>
</tr>
<tr>
<td>External Distributions to collect data from Excel</td>
<td>112</td>
</tr>
<tr>
<td>Probability Profile Distributions</td>
<td>114</td>
</tr>
<tr>
<td>Attribute Based Distributions</td>
<td>116</td>
</tr>
<tr>
<td>Combination Distributions</td>
<td>116</td>
</tr>
<tr>
<td>Time Absolute Distributions</td>
<td>117</td>
</tr>
<tr>
<td><strong>Shift Patterns</strong></td>
<td>118</td>
</tr>
<tr>
<td><strong>High Volume (BPR and FMCG) Applications</strong></td>
<td>120</td>
</tr>
<tr>
<td><strong>Groups</strong></td>
<td>122</td>
</tr>
<tr>
<td><strong>Results</strong></td>
<td>124</td>
</tr>
<tr>
<td>Results Summary Window</td>
<td>124</td>
</tr>
<tr>
<td>Pie Charts</td>
<td>127</td>
</tr>
<tr>
<td>Histograms</td>
<td>127</td>
</tr>
<tr>
<td>Time graphs</td>
<td>127</td>
</tr>
<tr>
<td>TIME GRAPH button on the tool bar</td>
<td>128</td>
</tr>
<tr>
<td>Combining Time Graphs</td>
<td>128</td>
</tr>
<tr>
<td>Exporting Time Graph Results Data to Excel</td>
<td>128</td>
</tr>
<tr>
<td><strong>Trials</strong></td>
<td>129</td>
</tr>
<tr>
<td><strong>Additional Graphics</strong></td>
<td>130</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Diagnostic Facilities</td>
<td>131</td>
</tr>
<tr>
<td>Contents Button</td>
<td>131</td>
</tr>
<tr>
<td>Simulation Monitor</td>
<td>131</td>
</tr>
<tr>
<td>SIMUL8 Assistant</td>
<td>133</td>
</tr>
<tr>
<td>SIMUL8 Information Store</td>
<td>134</td>
</tr>
<tr>
<td>VISUAL LOGIC EXTENSION USER GUIDE</td>
<td>136</td>
</tr>
<tr>
<td>VISUAL LOGIC INTRODUCTION</td>
<td>136</td>
</tr>
<tr>
<td>VISUAL LOGIC STRUCTURE</td>
<td>138</td>
</tr>
<tr>
<td>The structure of Visual Logic in your simulation model</td>
<td>138</td>
</tr>
<tr>
<td>The structure of Visual Logic Itself</td>
<td>141</td>
</tr>
<tr>
<td>VISUAL LOGIC BASICS</td>
<td>144</td>
</tr>
<tr>
<td>VL Blocks</td>
<td>144</td>
</tr>
<tr>
<td>Current Work Item</td>
<td>145</td>
</tr>
<tr>
<td>BUILDING VISUAL LOGIC</td>
<td>148</td>
</tr>
<tr>
<td>VISUAL LOGIC COMMANDS BY CATEGORY</td>
<td>159</td>
</tr>
<tr>
<td>Model related commands</td>
<td>159</td>
</tr>
<tr>
<td>Object Related Commands</td>
<td>159</td>
</tr>
<tr>
<td>VISUAL LOGIC COMMAND DETAILS</td>
<td>161</td>
</tr>
<tr>
<td>Add Work Item to Queue command</td>
<td>162</td>
</tr>
<tr>
<td>Beep</td>
<td>162</td>
</tr>
<tr>
<td>BreakDown</td>
<td>162</td>
</tr>
<tr>
<td>BreakRestart</td>
<td>163</td>
</tr>
<tr>
<td>Delete Work Item</td>
<td>164</td>
</tr>
<tr>
<td>Empty Simulation Object</td>
<td>164</td>
</tr>
<tr>
<td>If...ElseIF...Else</td>
<td>167</td>
</tr>
<tr>
<td>Loop</td>
<td>168</td>
</tr>
<tr>
<td>Move Work Item To</td>
<td>168</td>
</tr>
<tr>
<td>Percent Route Out Adjust to 100</td>
<td>169</td>
</tr>
<tr>
<td>Reset Clock</td>
<td>169</td>
</tr>
<tr>
<td>RunModel</td>
<td>169</td>
</tr>
<tr>
<td>Save Model</td>
<td>170</td>
</tr>
<tr>
<td>Select Current Work Item</td>
<td>170</td>
</tr>
<tr>
<td>Select Work Item on Conveyor</td>
<td>171</td>
</tr>
<tr>
<td>SET command</td>
<td>172</td>
</tr>
<tr>
<td>Set Collect Number</td>
<td>172</td>
</tr>
<tr>
<td>Set Route Out Percent</td>
<td>173</td>
</tr>
<tr>
<td>Set Route Out Discipline</td>
<td>174</td>
</tr>
<tr>
<td>Set Distribution Parameters</td>
<td>175</td>
</tr>
<tr>
<td>Set Resources Required</td>
<td>175</td>
</tr>
<tr>
<td>Set Object Image</td>
<td>176</td>
</tr>
</tbody>
</table>
### SIMUL8 User's Manual

Set Prob-Profile Distrib Column ................................................................. 176  
Set Work Item Image ................................................................................. 177  
Signal VBA ................................................................................................. 177  
Simulation Speed ...................................................................................... 178  
StopModel ................................................................................................. 178  
Until ............................................................................................................ 178  
While .......................................................................................................... 179

**SIMUL8 EXTENSIONS** ................................................................................. 180

**THE SIMUL8 / EXCEL EXTENSION** ............................................................. 181  
How It Works .............................................................................................. 181  
How to get started with the Excel Extension ........................................... 182  
Functions which interrogate and control your model ............................... 184  
Detailed Description and Examples of Excel Interface Library ............... 188  
AddWItoQueue .......................................................................................... 188  
AdjustRoutePercent ................................................................................ 188  
AttribValue ................................................................................................. 189  
AveSize ....................................................................................................... 189  
AveTimeIn .................................................................................................... 190  
AveTimeInByCategory .............................................................................. 190  
BreakDown ................................................................................................. 191  
BreakRestart .............................................................................................. 191  
ClearAllResourcesRequired ..................................................................... 192  
CollectComplete ...................................................................................... 193  
ConnectSIMUL8 ....................................................................................... 193  
EmptyQueue .............................................................................................. 193  
ExitSIMUL8 ............................................................................................... 194  
Get_WI_Held_Resource_Count ................................................................. 194  
MinAveMax ................................................................................................. 195  
ModelSpeed ............................................................................................... 195  
QueueSize ................................................................................................... 196  
ReSetModel ............................................................................................... 196  
RunModel ................................................................................................. 197  
SaveModel ................................................................................................. 198  
SelectWI_in_Object .................................................................................. 198  
SetAttributeBatchSize .......................................................................... 199  
SetAttribValue .......................................................................................... 200  
SetCollect ................................................................................................. 200  
SetConveyor .............................................................................................. 200  
SetDistribution ....................................................................................... 201  
SetMaxLevel ............................................................................................. 202  
SetPriority ............................................................................................... 202
SetResourcesRequired ............................................................................................................. 203
SetRouteMode .......................................................................................................................... 203
SetRoutePercent ....................................................................................................................... 204
SetWarmUpPeriod ..................................................................................................................... 205
SetWITLength ............................................................................................................................ 205
SimTime .................................................................................................................................... 205
StepModel .................................................................................................................................. 206
StopModel ................................................................................................................................. 206
StopTask .................................................................................................................................... 207
Utilization ................................................................................................................................ 208
Signals (automatic) from SIMUL8 to Excel ................................................................................ 209
DISTRIB: distribution name ........................................................................................................ 209
ENDRUN: ................................................................................................................................. 209
LOADED: model name ................................................................................................................ 209
RESET: ..................................................................................................................................... 210
ROUTE_IN_AFTER: work center name ...................................................................................... 210
ROUTE_IN_BEFORE: work center name ..................................................................................... 210
SET: attribute name .................................................................................................................. 211
WARM: ..................................................................................................................................... 211
Other Signals .............................................................................................................................. 211
Signals from Visual Logic .......................................................................................................... 211
Time Based Signals ................................................................................................................... 212
External Distributions - Signals ................................................................................................ 213
EXE Programs ............................................................................................................................ 214
Object Names from Excel/VB ..................................................................................................... 214
Hiding Visual Basic from the User ............................................................................................ 215
Interfaces to other languages .................................................................................................... 215
"Current" Work Item ................................................................................................................ 218
Using Visual Basic with EXCEL ............................................................................................... 219
Visual Basic Project Name ......................................................................................................... 220
Simple Importing of Data from Visual Basic ............................................................................. 221
A Free Service from Visual Thinking ....................................................................................... 224

TECHNICAL EXAMPLES ........................................................................................................ 225

INSTALLATION GUIDE ......................................................................................................... 227

STANDARD WINDOWS INSTALLATION ............................................................................... 227
NETWORK INSTALLATION .................................................................................................... 228
Introduction

SIMUL8 is a high power, but very easy to use, simulation package.

This SIMUL8 User’s Manual is not just about using SIMUL8 - its also about how to go about doing whole simulation studies.

First, lets take a look at what simulation is, then we will consider how SIMUL8 helps us do simulation.

Simulation - Introduction

Before we suggest a formal definition of simulation- lets look at the main benefits which it gives.

Simulation lets you explore an electronic model of the project you manage - whether the project is a factory, or a hospital, or an administrative center, or whatever. The type of model it provides is time-based, and takes into account all the resources and constraints involved, and the way all these things interact with each other as time passes.

This means that the model really can match reality - so something you try in the model will behave the same way as it would in the real organization.

With simulation you can quickly try out your ideas at a fraction of the cost of trying them on the real organization. And, because you can try ideas quickly, you can have many more ideas, and gain many insights, into how to run the organization more effectively.
Simulation is also a communication tool. As one of our first customers said: “We don’t want to just find the right solution, we want to use it to sell that solution to every one of our staff in our 200 branches”.

When you click the RUN button in a simulation model you see the work you do (products, patients, paper work etc.) move around the organization. The clock in the corner of the screen tells you what the equivalent time would be in the real system, and the software automatically collects performance measures as the modeled resources work - so that, you not only see visually what will happen, you can also get accurate numerical results to prove your case.

**Definition**

Simulation is the process of building and using a time based visual model which emulates every significant step that occurs in a process and every significant interaction between resources in a process so as to gain insight about the impact of potential decisions on that process. The model shows you visually what will happen in the process if you make changes to it and it records performance measures of your system under different scenarios.

Actually different people (working in different fields) define simulation in different ways. For example, in the aircraft industry simulation means flight simulators which help a pilot train to fly a new type of aircraft.

However, we will consider simulation in the business decision making sense described in the definition paragraph above.
SIMUL8 - Introduction

SIMUL8 is different from many other packages because its design is based on some careful research rather than a desire to exploit a potential market. Our main objective in launching SIMUL8 is to get simulation used so that it can benefit industry and commerce in the way simulation should.

One of the main impacts of the above statement is that the way you use SIMUL8 is not based on programming or statistical data, but on drawing your organization (with the computer’s mouse) on the screen, and only filling in numerical information where you need it.

However, you can still use SIMUL8 for traditional hard number crunching simulation. It is built with the expectation that this is the likely finale of most simulation studies. But our experience (and we have a good deal of simulation experience) is that what people need in a simulation study is:

1. A very quick model to enable them to start thinking about the structure of a situation.

2. To be able to build in numerical information as and when it becomes available but only to the level of detail which is appropriate for the level of investigation required.

3. As the model is used more, and the people concerned start to home-in on detailed plans for changes to their organization, to be able to cope with any level of complexity.

4. To be able to obtain easily understood, but rigorous, performance results from the simulation package.
One and Two above recognize that simulation is a communication and thinking tool, not just an analytical tool.

The Process

You build your SIMUL8 model by drawing it on the screen. You place work centers (machines or desks or people or other facilities which do work) on the screen by clicking buttons and dragging the objects to places on the screen which match approximately with the real organization. You specify the route which work normally takes though the organization by dragging arrows between the objects.

The model will run as soon as you have done this.

Other objects are provided to make it easy to represent the way things really work. (For examples we provide “Storage bins (queues)”, “Conveyors”, “Floating Resources” etc. etc.).

Then you click on objects on the screen if you want to put more detail into them.

Initially this detail consists of timing information, routing information etc. But later, if needed, you can control each of the objects from external sources like spreadsheets or the Visual Basic/EXCEL.

The two-way interface with Visual Basic is the facility which means SIMUL8 can both be very easy to use, and unlimited in its flexibility and power. Because SIMUL8 links to Visual Basic and because Visual Basic is a full programming environment which is now the macro language used in standard packages like EXCEL, it means SIMUL8 can be extended by you, using the same tool as you would use if you were to add macros to EXCEL. Experience so far with SIMUL8 would lead us to think that Visual Basic will not be used with many SIMUL8 models - but it is there if you need it so SIMUL8 will never limit you.
The next section of this manual takes you step-by-step through building a very simple model.
Building your first SIMUL8 model

With SIMUL8 you build your simulation model simply by drawing it on the screen.

(You can follow this chapter in an on-line tutorial - Click HELP / GETTING STARTED in SIMUL8)

We suggest you start with a model which looks like the one in the picture overleaf, and then extend it.

However, before we start, a short overview of some concepts:

When drawing your model you are drawing onto the screen a number of simulation objects. These are things like work-centers (which perform work) and queues. These are joined up by routing-arrows which describe the default route which work-items take as they move through your simulation model. Work-Items are simply things which you work on in your organization (like products, or people, or paper work).

Note: The pictures here assume you have chosen the “SERVICE INDUSTRY” option during installation - but the model will work the same which ever option you have chosen - just the icons will look a little different.
Lets try building this simulation model:

Click the 'Work enters operation' button (\[\text{\[\text{\[\text{\[\image]}}\]}}\] ) and place the icon that appears somewhere on the screen.

Next click a 'Storage' button and do the same.  

Then place just one work center on the screen using the :\[\text{\[\text{\[\text{\[\image]}}\]}}\] button (we will copy it in a moment to make the second one).

And finally somewhere where work leaves the system with the :\[\text{\[\text{\[\text{\[\image]}}\]}}\] button

\textit{SIMUL8} has automatically linked them together for you (the arrows). It does this for simple models only. Normally you put the links in (or take them out) by dragging the mouse- See page 80.

You can run the model now if you like.

However if you want to put a second work center in (like in the picture above) PRESS and HOLD the "CTRL" on the keyboard and then drag "Work Center 1" to
a new location on the screen with the mouse. (This is similar to the way you can copy files in File-Manager).

Now click the RUN button: (the button shows a consultant running to get a report in the last post!).

Click the RUN button a second time to stop the model.

If you don’t tell it otherwise your model will run for one week (a 40 hour week). At the end of the week you can look at some results (measures) on how well the system has performed.

For example, click on the “work complete” icon:

and then on the RESULTS button which will have appeared.

This will show you how long the items of work which went through your model took to get through the system (on average):

Now, all this is dependent on the timing information which is in the model (and which so far SIMUL8 has made assumptions about without asking you!)
SIMUL8 takes this approach because it lets you build a model fast when all you are concerned about is using the model as a discussion vehicle to get the structure of a situation understood.

When it comes to accurate analysis, we need to put some numbers in.

If you still have the results window on the screen, click the CLOSE or OK buttons to get back to just the main simulation window.

Click on one of the work centers:

```
0
```

You will now see a dialog box which gives you access to an enormous number of parameters and settings which can be used to configure how work is done.

For now, just change the number in the box labeled “Average” from 10.00 to 20.00.

Click the OK button,

Now click the RUN button.

Once the model has run, take another look at the RESULTS inside the “WORK COMPLETE” object.

The work took longer (on average) to get through the system - this is to be expected - but simulation has told you HOW MUCH LONGER!

Have a look at some more results.....how long was the queue on average?

Go back to the main simulation window (click CLOSE and OK until you get there).
Now click on the STORAGE BIN:

and click its RESULTS button. Then take a look at its TIME GRAPH:
(Click in the results dialog)

It will look something like this:

![Graph](image)

An average of only 0.3 jobs queuing - but a worst case peak of 4.

You should take some care about the way you interpret the results and performance measures. You should read the section of this manual on “Simulation: The Technique - a Guide” to understand how to use results which come from simulation models.

However, first we suggest you now have a browse through the “Reference Section - The SIMUL8 facilities” part of the manual.
Overview of using SIMUL8 for building large models

The purpose of this section is to give some guidance and insight into the facilities which can be used when building rather more majestic models than the one mentioned in the previous section!

SIMUL8 contains a wealth of facilities which are described in detail in the help file and the reference section of this manual.

The current section gives an overview of a few of these facilities that the experienced simulation user might want to know about before tackling significant models.

Main Concepts
(This discussion is repeated in more detail at the start of the reference section).

A SIMUL8 model consists of “Objects” (things like queues and work centers) on the screen with a default structure (routing) between them and “Work Items” which flow around the model - these “work items” are the “jobs” which are done in your organization.

Work items come in different varieties (if you want). All models have a default “Work Item Type” - but you can create as many more types as you like.

Work Items can have attributes (like “illness”, “age”, “history” if your work items are patients in a hospital). Each individual work item can have different values for each of its attributes. Values of attributes can be changed and used by work centers.
An important type of “Object” is a “Resource”. These are “floating resources” which can be used at work centers. For example, if work centers are machines, they might need resources called people to operate them. Resources do not have to be available for work all the time - you can describe shift patterns to which they can work. Work centers do not have to use a fixed number of resources to work - they can be set to work faster if more resources are available.

All objects and work item types can be given their own name. We recommend the first thing you do when you start building a new model is to rename the default work item type to something more relevant to your environment. (“Work Item Type 1” is not a good name for “Customers”!).

**Display**

**SIMUL8** lets you build a model which is heavily customized to your own situation.

Most of the examples shown in this manual use standard facilities (and specifically just the standard facilities with the “SERVICE INDUSTRY” version). However, you can easily create your own images to replace or enhance those provided as standard.

There are many example images provided in the **SIMUL8** directory on your computer.

Objects do not actually have to be displayed - you can make them invisible if you want. This often done if you want to hide a complex part of your model which might not be important to a particular aspect you want to review today. (Although in a large model you are likely to use sub-windows to do this). It is also used to hide a “synthetic work center” - one which controls routing but does not physically exist in the real factory (it is just a control rule).

You can add to your model’s display lines, text and graphics which are not part of the actual structure of simulation logic, but instead are there to add to the realism of the model, so that it looks like the situation being modeled.
The default display shows only the numbers of work items within each object and the work items as they travel between objects, but you can select to display much more information about each object, and the work items themselves as they travel around the model. To achieve this, click on the relevant object, then the GRAPHICS button which will have appeared.

**Tailored Interactive Aspects**

If you want to set your model up as part of a regularly used system in an organization, it is sometimes better to create your own interactive facilities specifically tailored to the environment concerned. You can do this by creating dialog boxes of your own (using Excel/Visual Basic) and controlling the SIMUL8 model from Excel/Visual Basic. There are examples provided which show how to do this.

**Models with similar parts**

If your model is likely to consist of a number of similar segments (for example 5, more or less similar, production lines, or a group of 4 operating theaters with broadly similar procedures) then our experience with SIMUL8 has been that it is quickest to build one of these sections first and then simply copy this section as many times as required. The on-screen copy facility lets you drag a box around many objects and then simply drag the box (with CTRL pressed) to copy them and their structure.

Similarly (but on a smaller scale) if you have several work centers which work the same way, it is worth setting up one, and then copying it to create the others.

Named Distributions are useful for speeding up the creation of models (and later changing them). A “named distribution” is simply a statistical distribution (say Normal with a mean of 12.4 and standard deviation of 2.1) which has a name (say “Machine Cycle Time”). All your work centers which are machines of this type can then use this distribution - and if you need to change the speed of your machines you can simply change the parameters of the named distribution.
Models within models (Sub-Simulation Windows)

A feature which is particularly useful for large models is being able to shrink part or parts of the model into sub-windows. You can drag a box around part of the model and the click the right mouse button and “make sub window”. All the objects in the box going to a new simulation window which can then be closed and it becomes a single icon on the screen. You can have sub windows within sub windows - to any level of depth, and have any combination open or iconic, to watch any aspects of the model you like.

This means you can create a model of a whole organization, with multiple factories, each containing multiple product lines, each containing multiple cells, and view two cells in different factories and the impact on an organizational variable at the same time, if this helps you in modeling your organization.

If (during your project) responsibility for one of the sub-sections changes to a different department you can simply drag one sub-window into another on the screen.

Model Structure

With SIMUL8 you build a default structure (routing for work items) into your model, however, your work items can take different routes. This is controlled using the Routing In and Routing Out facilities which tell work items where to go when they enter or leave a work centers. The routing in and out dialogs also let you control a wealth of other features which affect the work center’s choice about work to do and what to do with it.

External Data

Although the interface to Excel/Visual Basic is in SIMUL8 to let you control the logic of your model you can also use it to simply provide external data to your model. This lets you quickly use data that is in a spreadsheet (or can be got into a spreadsheet) in your model. The facility works by letting you create what SIMUL8 uses as a “distribution” but which is actually data in EXCEL. The data which SIMUL8 uses (where there might normally be a statistical distribution) actually comes from a column of cells in the spreadsheet.
Zoom
The simulation window (or any of the sub-windows) can be zoomed so that you can easily get an overview, or look at detail, when required.

Business Process Re-engineering
Work centers normally work on jobs “one at a time” modeling each work item as a completely unique item in the model. Every one can be individually distinguished so that you can model any situation in a great deal of accuracy. However, some situations are better modeled with a slightly different approach. BPR applications (where you want to model high volumes of transactions) do not need to model every individual piece of work. It is usually better to categorize the work and think in terms of number of items in each category. The High Volume switches in SIMUL8 let you automatically switch individual parts of your models into this mode when you need. (Fast Moving Consumer Goods (FMCG) applications can also benefit from use of this facility.

Documenting Models
Document your SIMUL8 model by using MEMO buttons in object dialogs and by adding notes to the model itself (File / Model Details).

You can create a formatted text file which documents the entire state of the model using File / Documentation. This file can be inserted into word processing documents (as part of your report) or printed for detailed checking of the model.

Visual Logic
Visual Logic (VL) is SIMUL8’s logic building environment. In a model of significant complexity you will want to add your own rules for deciding how to process work. VL lets you add very detailed logic to control the operation of your model. It is naturally structured at a number of levels to allow you to quickly access the logic used in a particular part of the model and also to easily see the logic of the rules you have entered. A guide to Visual Logic starts on page 136.
Information Store

The Information Store holds SIMUL8 “Global Data Items”. These are variables which can be used in Visual Logic. They can be single numbers or “spreadsheets” (a matrix of data).

Create your own Global Data Items by typing a new name in the name box and clicking the ADD button. Rename an existing item by clicking it in the list (its name will appear in the name box) then edit its name in the name box.

You can inspect and change the current value of a data item in the Value box.

When the SIMUL8 RESET button is pressed the data item's current value will be set to the value in the On Reset box.

As Spreadsheets have many values you need to click the VIEW button to see the values in a spreadsheet. Spreadsheet values are not changed on RESET.

In programming terms it works just like an array, but visually it is like a spreadsheet.

To change values in a spreadsheet simple type the new values.

To copy/paste the entire spreadsheet to/from other SIMUL8 spreadsheets or other packages use the COPY/PASTE buttons in the spreadsheet dialog. (Using CTRL C and CTRL V will copy/paste the currently highlighted cell).

What Next

This is designed as only as a very brief overview of a few aspects of SIMUL8. We strongly urge you to have a browse through the reference section of this manual (or the SIMUL8 Windows help file) to find out more about the SIMUL8 facilities before you embark on building a large model.
Simulation: The Technique - a Guide

The main purpose of this section is to enable you to understand and make sensible use of the results which you get out of SIMUL8. However, it would be wrong to convey to you the idea that simulation starts with building the model and ends as soon as you get the results.

For this reason, this section covers rather more than just getting the results out of the model. It attempts to convey some guidelines on the whole process of conducting a simulation study. You will however find specific sub-sections (h & j) which do deal with the results of model runs and how to interpret them.

The rest of this section is divided into a series of steps, but in practice simulation is not done in a series of steps. Simulation is much more of an iterative process, cycling around until the problem is dealt with. Nevertheless, here we have described it as a series of phases to enable you to see the elements which you need to include.

This is written from the point of view of someone undertaking a consultancy role. Some of the steps are obviously not necessary if you are building a model for your own use - but it is still perhaps interesting to consider them because the existence of these steps highlights one of the major benefits of building models for yourself!

(a) **Listen**

Listen to your client to find out what issues concern them and how their factory (or what ever the system is) operates. Their thoughts on what concerns them will give you a good idea of what variables are important to model in their factory.
At this stage you might, of course, decide that simulation is not the way to help the client! - but, assuming you do not reach this conclusion.....

(b) **Level of Detail?**

Think about the level of detail that it would be appropriate to build into the model. For example, do you need to include every part of every operation at each machine in the factory, or is it more appropriate to simply model the output of whole departments and how they interact with each other? Putting too much detail in a model means it costs more of your time to build and can also cloud the issues which the decision maker is interested in.

(c) **How Broad?**

Think about how wide to make the model. Can you limit the boundaries of the model to the factory itself, or do you need to also include the systems which affect supply of raw materials to the factory?

(d) **Build Rough & Ready Model, then Refine it**

As quickly as possible build a rough and ready simulation model (it doesn't matter if some data is missing or if it does not run properly) and take it to show your client. Experience has shown that even static drawings on a computer screen can do good job of getting your client to understand (i) more about what it is they will get from you and (ii) what aspects of the conversation, (a) above, have been misunderstood and (iii) what aspect of their worries are different now that a consultant has put some structure on them.

This phase (d) is much more important than it may seem. Experience is that the consultant's first idea of what simulation model is appropriate is never much like the simulation model at the 'end' of the study. This is not because the consultant was necessarily wrong, but because the client's needs evolve throughout the process.
It doesn't matter how rough and ready the model was because you now have to start again!

(e) **Work Closely with your Client**

Having built a more robust model you now need to start working more closely with your client. (In fact, with the types of software becoming available now, it is possible for all the stages above to be part of one session with the client).

It is especially important to work with the client because doing simulation is like a journey of discovery with numerous dead ends. You will try many experiments and learn a lot about the way the factory operates and could operate under different circumstances. If your client is not engaged in this journey of discovery then they will not understand how you have reached the end result. This means that implementation of anything other than an obvious solution could be difficult.

We will look more at how you build models in the section below on simulation technology, but it is important to realize that there are two key, but separate, skills in this model building process.

1) Being able to use the software to build the model.

2) Being able to design the display for a model which highlights the important issues to you and the client without clouding the issues with information overload.

The second of these is not easy to learn but as you read the rest of these tutorial you'll get a feel for its importance and what is likely to affect its success.

The phase above (e) breaks down into many sub-steps - each of which are described next. This phase continues until the client feels they know what is best to do: In other words until the problem is "solved".

---

35
(f) Running your Model
You will already have been running your model quite a bit by now (by ‘running’ we mean setting the clock going and see the products moving around the factory) and you may have gained many insights into how the factory works and what changes could be made to improve it. However, it is very important to ensure certain key steps are taken to ensure validity of these discoveries.

(g) Verification
The first of these steps is to verify that the model is running in the same way that the real world would work in the same circumstances. In theory there is a need to do this every time you make a change to the model, but clearly this would be impractical. If the simulation is a model of an existing factory (or other system which actually exists) then this verification stage is just a matter of ensuring the model behaves as the real factory currently does. So you can put some actually customer orders (for example) into the model and see if they are handled in the same way and in the same time as they are in the real factory.

The only way to verify a model of a new factory (which does not yet exist) is to carefully experiment with the model, watching (both the animated screen and the performance measures) to see if the changes you see are reasonable in the light of changes you make to the input to the model (such as demand, numbers and speeds of machines etc.). There is a methodological problem here because clearly if these changes were known, or expected, there would be no need to build and use the simulation model in the first place. Nevertheless, careful examination of the model's behavior will allow you to see and remove most mistakes or misunderstandings in the model building.

It is very easy to omit this stage in the excitement and urgency of trying to help your client find a solution. Omitting it is also one of the easiest ways to loose a client's faith in your work when they see the model show some result which is not only intuitively wrong but proves to be caused by a mistake!
(h) **Warm Up Time**

You may need to find a suitable 'run-in' or 'warm-up' time for your model. Models of factories usually need a warm-up time because your model is likely to start empty (no products at any of the machines), but real factories do not start each week without any work-in-progress! A situation where you would not need a warm-up time in a simulation model might be when you are simulating the start up of a new factory, where you happen to be interested in its first day's operations, or where you are simulating a situation which starts each day empty: like a shop, or a car park, or an airport. However, even in these circumstances you would probably want to separate off the results you get during the early part of the day from the results under peak conditions because knowing the 'average' performance of an operation which never works under 'average' conditions is probably not worthwhile.

Most simulation packages contain facilities to set a 'warm-up' time during which results are either not collected, or which can be separated off from the main results collection time.

But how to you decide how long to make the warm-up time? There is a simple procedure for this.

1) Decide what measures of performance are of interest to you. For example you might be interested in the number of products leaving the factory, or the utilization of a machine or the length of a queue. If there is more than one then you might want to carry out this procedure for some, or all, of the measures which interest you and use the longest warm-up time indicated. Lets just assume you are interested in the number of products leaving the factory.

2) Run the model for a short period of time. The amount of time you run the model for is rather dependent on what you are simulating but if the total time you wanted to run the simulation for was, say, a week (see below for how to decide this) then you might make this "short period of time" an hour. It needs to be an amount of time in which you might
expect to get a few products out of the factory. Anyway, having run the model for, say, 60 minutes, record the number of products which have emerged from the factory. *This will almost certainly be zero for the first 60 minutes!* This is because of the nature of the need for a warm-up time. No products will have reached the end of the factory yet!

3) Run the model for another 60 minutes (or whatever time you have chosen) and record how many products left the factory in the second 60 minute period. i.e. the number which emerged *between 60 and 120 minutes - not the total at the end of 120*. Keep doing this and you should end up with a table looking something like this:

<table>
<thead>
<tr>
<th>Time</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>180</td>
<td>5</td>
</tr>
<tr>
<td>240</td>
<td>4</td>
</tr>
<tr>
<td>300</td>
<td>8</td>
</tr>
<tr>
<td>360</td>
<td>23</td>
</tr>
<tr>
<td>420</td>
<td>15</td>
</tr>
<tr>
<td>480</td>
<td>22</td>
</tr>
<tr>
<td>540</td>
<td>21</td>
</tr>
<tr>
<td>600</td>
<td>16</td>
</tr>
<tr>
<td>660</td>
<td>21</td>
</tr>
</tbody>
</table>

which you could plot in a graph to make it easy to see what is happening:
If you plot the graph as you collect the numbers you will be able to see when you have gone far enough. In the example above we can see (by the time we get to simulated time 660) that there is a fairly distinct difference between the figures prior to time 360 and the figures after time 360. So before time 360 the simulation model was still 'warming up'.

Obviously you will not expect a completely steady line after time 360 because simulation matches reality - and you cannot expect completely steady output from a real factory. The simulation is using random numbers (mentioned above) to emulate the variability in the real world (e.g. for the time a machine takes to weld each box), so you would expect some variability in output. There might be some variability even if the simulation model contained no randomness. This will be discussed later.

This example has been deliberately made clearer than often occurs. Quite often the performance measure in which you are interested never looks at-all stable. Sometimes during the warm-up time the measure you are interested in is very high, sometimes it is very stable - and becomes unstable once the model has warmed up.
You will have to make a judgment when you look at the figures (and a graph) about how much warm-up time to allow. The principle is simply to ensure that the model is not in some atypical start-up state caused by the model starting empty.

4) Once you have determined a warm-up time it is worth adding about 20% to it as a safety margin and then using this warm-up time for all your analysis of the model unless the structure of the model, or the data in it, gets changed significantly. Strictly speaking you ought to recheck the warm up time every time you change anything in the model (even the random number streams) - but in practice this would take more time than is usually available! The only harm in making the warm-up time too long is that it can take up a lot of your time when you are running the model.

Significant changes to the model might affect how long it would take to warm-up, so you might need to repeat the above process.

(i) **Visual Exploration**

This is the most important step in undertaking simulation.

Traditionally the next stage in using simulation was to design and perform a series of experiments with the model to see how the factory could be improved. Details of how to do this are in section (j) below. However, a much more important step in doing simulation (and one which only became possible when simulation became Visual and Interactive) is to explore the model visually with your client. This enables both you and your client the opportunity to gain an understanding of how the different parts of the model interact with each other, how the performance measures can be affected and to gain insights into how the factory can be changed and improved.
This 'Visual exploration' is done by running the model and watching the products move around on the computer screen. You can, for example, see where bottlenecks build up, you can see under what circumstances these are worst. By changing the model and watching to see the effects of these changes you can gain a knowledge and understanding of the way the system behaves. You can get to know its 'character'.

Of course, exploring a model visually like this is not a rigorous or scientific testing procedure and should always be followed by some careful tests (see (j) below) but it is much better at giving insights into how to improve the factory (or other system) than a series of controlled experiments.

(j) Accuracy of Results
The procedure described here can be handled for you, fully automatically by SIMUL8, but it is described here so that you can fully understand the principles involved.

Once some insights have been gained it is important to test the resulting ideas, especially if there are a number of competing ideas and it is difficult to see visually which is best. Remember that a simulation (usually) contains random numbers and if you are simulating a week's production when you explore the model visually, you may be seeing results which apply only to one week (perhaps a lucky week when few of the machines broke down!). A different week might give you slightly (or very) different results.

This procedure described here gives you a step-by-step way to ensure your results are valid. Even if you do not feel a need to go as far as calculating the statistics, you should do the first part and run the simulation model with a number of different set of random numbers.

When using most simulation packages, if you set the time clock back to zero and re-run the model you will see exactly the same things happen on the screen, in the same sequence as the last time you ran it (despite the fact that the model
contains random numbers to emulate ‘real life’). This is because simulation packages use ‘pseudo random numbers’ which are generated mathematically and simply appear to be random. Each time the random numbers are re-started, the same sequence of numbers will be generated. This is very useful because it means you can re-watch a model several times to understand exactly what is happening, without the issue being clouded by the random numbers changing each time.

All simulation packages allow you to change the random numbers so you can also see what happens when the random numbers are different. They do this by allowing you to set the ‘stream’ of random numbers which will be used. Most packages have many thousands of ‘streams’ of random numbers built into them so there is no limit on the number of different weeks you can simulate in the factory.

But wait a moment - why are we only simulating a week's production? Why not a year, or an hour? What is the right length of time to simulate? The answer is simple. Simulate an amount of time which makes sense to your client in terms of the performance measure you are using. For example if you simulate the factory for a year and report to your client that you expect the factory to produce 14,500 boxes in a year this might be useful information in itself but might hide the information that any given week's production might be as low as zero or as high as 500. Conversely, your client may be unconcerned by information about the output in any given hour. Choose a time which makes sense to the client. The decision about what time to choose will become clearer when we see (next) what we will do with the information we get.

Lets assume we choose to simulate the factory for one week at a time and with each random number stream we get a different number of products completed by the end of the week. This information can be really useful to your client for making decision with - but it has to be presented in a useful and valid way for it to be of genuine use. This section is all about how to do this.
Once we get to this stage of a simulation study we are normally doing these runs and recording the results because we want to compare two or more alternative decisions which we cannot easily distinguish between visually. Let's assume we have two alternatives and we have run the simulation 30 times for each of these alternatives. For each run (of a week plus warm-up time) we have used different random numbers and obtained the following output figures:
Number of products leaving the factory in 30 different weeks
(30 different random number streams).

<table>
<thead>
<tr>
<th>Alternative</th>
<th>One</th>
<th>Two</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>293</td>
<td>261</td>
</tr>
<tr>
<td></td>
<td>274</td>
<td>264</td>
</tr>
<tr>
<td></td>
<td>196</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>254</td>
<td>287</td>
</tr>
<tr>
<td></td>
<td>203</td>
<td>287</td>
</tr>
<tr>
<td></td>
<td>228</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>234</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>228</td>
<td>257</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>210</td>
<td>334</td>
</tr>
<tr>
<td></td>
<td>234</td>
<td>242</td>
</tr>
<tr>
<td></td>
<td>216</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>218</td>
<td>301</td>
</tr>
<tr>
<td></td>
<td>315</td>
<td>292</td>
</tr>
<tr>
<td></td>
<td>239</td>
<td>288</td>
</tr>
<tr>
<td></td>
<td>246</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>189</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>229</td>
<td>253</td>
</tr>
<tr>
<td></td>
<td>259</td>
<td>265</td>
</tr>
<tr>
<td></td>
<td>273</td>
<td>239</td>
</tr>
<tr>
<td></td>
<td>265</td>
<td>298</td>
</tr>
<tr>
<td></td>
<td>248</td>
<td>223</td>
</tr>
<tr>
<td></td>
<td>280</td>
<td>319</td>
</tr>
<tr>
<td></td>
<td>232</td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>212</td>
<td>260</td>
</tr>
<tr>
<td></td>
<td>238</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td>287</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>254</td>
<td>298</td>
</tr>
</tbody>
</table>

The results for alternative one are as low as 189 and as high as 315 with the figures for alternative two being 188 and 334. Which is best? The averages for the two alternatives are 245.8 and 270.5, so perhaps alternative two is best.
There is so much overlap between the two, perhaps there is not really any difference between the two alternatives. Can your client really justify spending the extra £1 million that alternative two will cost? What if the next 30 weeks (random number streams) average out the other way around? What will the genuine long term average really be?

We could do more many more runs of the models but in practice running a simulation model of a complex factory for even one week can take a long time, and your client may not be able to wait.

A simple statistical procedure can help here. With limited time to do a limited number of simulation runs we cannot given our client a dead-accurate long term average, but if we could tell our client what range we expected the long term average to be inside this might be good enough.

In addition to the information we have so far we could calculate the standard deviation of the individual results:

<table>
<thead>
<tr>
<th>Alternative</th>
<th>One</th>
<th>Two</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average:</td>
<td>245.8</td>
<td>270.5</td>
</tr>
<tr>
<td>Standard Deviation:</td>
<td>33.1</td>
<td>33.3</td>
</tr>
</tbody>
</table>

But this only tells us the variability of the individual results. We are interested in how safe our average figure is. If we had time to do another batch of 30 runs, would its average be different? If we did 30 batches of 30 runs what would the variability of the averages be?

Fortunately for us there is a statistical relationship which applies here which provides a method of predicting the standard deviation of the averages from the standard deviation of the individual results - which we have just calculated.
Standard Deviation of Averages = Standard Deviation of Results / √Number of Results

In our case:

Standard Deviation of Averages = 33 / √30 = 6

We can now calculate confidence limits for our average to report to our client.

As 95% of normally distributed numbers are within 2 standard deviations of their average then:

95% confidence limits for our average = Average +/- 2 x 6
= 245.8 +/- 12
= 233.8 to 257.8 for alternative one
and
= 270.5 +/- 12
= 258.5 to 282.5 for alternative two

This means that we can say we are 95% confident that the true average for alternative one lies in the range 233.8 to 257.8 and also that we are 95% confident that the true average for alternative two lies in the range 258.5 to 282.5. So (again with 95% confidence) alternative two will produce more boxes in the long term (but not every week!)
A picture of this can be useful:

![Diagram showing two ranges with a shaded area]

Showing this type of picture to your client can help understanding especially if you change the numeric production count information into profit or cost. The above picture shows that it is *just* possible to distinguish (statistically) between these two alternatives.

Of course the picture might show this:

![Diagram showing two ranges with an overlap]

An overlap between the two ranges in which case:

1) You **cannot** distinguish between the two alternatives. It is not reasonable to report any difference between them to your client because the average for alternative one could be as high as 257 and the average for alternative two could be as low as 254.
2) You could do more runs because (take a look at the way the range is calculated) the larger the number of runs the smaller the predicted standard deviation of the averages: so the smaller the range. Of course this does not mean the two ranges will definitely separate if you do more and more runs - it simply means your figures are getting more and more accurate. You may move towards something like this:

![Diagram showing ranges and averages]

If this happened it would show there is no difference between the alternatives.

It is rarely necessary to do as many as thirty runs. If you do five runs and then do these calculations you will get an idea as to whether five is enough or whether you will need to do more. Even as few as five can be enough - see the note on statistical theory below.

To summarize the procedure:

1) Do five runs of each of your alternatives, each run using a different random number stream.

2) Calculate the average and standard deviation for each of your alternatives.

3) Calculate the 95% confidence limit ranges for each alternative = Average +/- 2 x Standard Deviation / \(\sqrt{\text{Number of Result}}\)
4) Draw these up on a picture like the one above to see which alternatives can be distinguished from each other, possibly changing the data to something more meaningful full to your client (like £ or $).

5) If ranges overlap do more runs to see if they separate.

A bit of statistical theory: This procedure only works if we know the averages are going to be 'normally' distributed. If we have at least 25 runs to calculate our average then this will be the case but fortunately for simulation users this is usually the case for smaller numbers of runs because the average will be normally distributed if the individual run results are normally distributed. These are (usually) normally distributed because they are, themselves, the combination of a large number (more than 25) individual events (the times between products leaving the factory). A statistical theory called 'central limit theorem' says that any number which is the total (or average) of a large number of other numbers will be normally distributed, whatever the distribution of the original numbers.

(k) How much of this do you need?

The final point here is that you might not get very far through the above phases when doing a simulation study. One of the most important things to realize about simulation in practice is that much of the benefit of simulation comes from the model building phase of the study. The act of building a simulation model causes you to have to ask your client many questions so that you can gain sufficient understanding of the process to build a model. This questioning process has the benefit of structuring in both your and the client's mind many thoughts about the process which had not previously been pulled together. The simulation model then acts as an understandable piece of documentation about the process. It is a 'document' which is extremely easy for most people to understand (being a moving picture of the process).

We realized this quite early on in the days of Visual Interactive Simulation when we noticed that clients were very keen for us to build models for them but often before we were ready to formalize the experimentation process they wanted the
model changed. The act of building the model itself was sufficient in some cases to enable the clients to see a new way of operating the plant which they wanted to see tested in the model.

The consequence of this that you will spend most of your time in the early steps of the above process and only dip into the more technical steps occasionally. It also means that the model building stage needs to be relatively easy and quick if you are going to be able to help your client and if your client is to be involved in this process.

**Conclusion**

**SIMUL8** has been designed specifically to match with the above recommended way of conducting simulation studies. In particular it recognizes that a crucial aspect of simulation studies is being able to build and change models very quickly - so that the thinking process (the process of gaining insights into how a system can be improved) is not interrupted.
Reference Section - The SIMUL8 facilities

The Main Concepts

Structure of your model

A SIMUL8 Simulation model is made up of OBJECTS in a structure. Work Items (work to be done in the operation you are modeling) travel through the structure with objects controlling their flow.

The structure is represented on the screen by a number of objects and lines joining these objects.

A great deal of detail can be placed behind this structure. This makes it possible to use SIMUL8 at a strategic level, and at a detailed tactical level, as required.

See the “Building your first SIMUL8 model” for a first view of a model structure.
Important Terminology

Two fundamental concepts underlie most model building in **SIMUL8**:

Objects and Work Items

However, before we look at these we will describe where you will see them.

**Simulation Window**

The simulation window is the main representation of your simulation model within **SIMUL8**.

The contents of the window are kept constantly up-to-date so that you can see what happens in your model as the simulation runs. From this you gain insights into how the ideas you have about your system work and how they can be further improved.

You create your model in this window by drawing a picture of the decision system you want to model.

Each icon in the simulation window represents an object in the simulation model (or an icon can also represent a simulation sub-window because your model can exist at a number of different levels).

To inspect the detailed parameters of any object on the screen, or to look inside any of the sub-windows, simply click on the object. Inspecting the object also lets you look at its performance results (collected as the simulation model ran).

Moving in the simulation window you will see the work items which are flowing around your model.
Simulation Objects

An **OBJECT** in **SIMUL8** is something in the simulation model which you normally draw on the screen and which is normally involved in the work done in the system you are modeling.

The **SIMUL8** simulation objects are:
- Work Centers
- Conveyors
- Storage Bins (Queues)
- Resources (People and other items required to do work at Work Centers)
- Work Entry Points
- Work Exit Points

Typically the work done in your system will be simulated by it going through a number of objects which are on the screen, but they can be invisible, and some types of object (e.g. resources) are "used" at other objects (e.g. at work centers).

All simulation objects have names (for example “Fitters”, Work Center 1”, “Xray Department” etc.).

The items which flow through the **SIMUL8** model are called Work Items.
Work Items

A Work Item is the work which is done in the organization being simulated. Examples are:

Patients in a hospital.
Invoices in an Accounts department
Products in a factory.

Individual work items flow through the work centers (and other objects) in a simulation model.

Each work item is of a "type" known as a "Work Item Type".

Work item types (and therefore all the work items of that type) may have "attributes" attached to them. All work items of the same type have the same attributes attached to them **BUT** the values in the attributes for each individual work item may differ.

For example: a work item type called "Patient" may have an attribute called "illness" which may be set to 1 for some patients (for "minor"), 2 for some patients (for "serious") and 3 for others ("life threatening").

There is a standard image which is used to display a work item on the screen (the **SIMUL8** #8 ball) but each work item type can have its own default image, and individual work items can have their own individual images changed as they flow through work centers.

**Important Images on the Simulation Window**

Note: (The icons shown here are the default icons in the “service industry setup” of **SIMUL8**. You can also design and draw any icons of your
own design in **SIMUL8**, or use any of the other images supplied on the installation disks).

**Work Enters Model Object**

This object is a "list of work to be done" in the model. Usually it simply describes the average time between items of work entering the model. By it can, for example, contain list of work stored in a spreadsheet.

**Storage Bin (Queue) Object**

This icon represent a queue in the model (in the simple example here it is shown as a static box, but it is more usual to display it as a queue - showing the work items in the queue as they move through the simulation model). It can be changed using the graphics button in the storage bin’s dialog.

**Work Center Object**

This icon represents the most important type of object in most **SIMUL8** models - the places where work is performed.

**Work Exits Object**

This is where work is declared as "finished".
Work Flow Arrows

These indicate the default path for items of work moving through the simulation model.

Draw them by dragging the mouse from one object to the next while holding down the **SHIFT** key on the keyboard. (You can also use Route Drawing Mode to draw them without the SHIFT key - see page 80).

Contents Count

This number indicates the number of work items in this object.

The value is updated continuously as the model runs.

It can be hidden using the GRAPHICS buttons in the object dialog boxes.
Work Centers

A Work Center is a place where work takes place on Work Items.

Work done at work centers usually takes up time and sometimes requires the availability of resources. At a work center a work item may be transformed in some way (perhaps by changing one or more of its attributes).

After the work is done the work item may be sent on to another work center, or one of a number of different work centers depending on routing rules which can be specified.

To create a work center click the create work center button and then click where you want the work center on the simulation window.

To move a work center: drag it on the simulation window.

To inspect or change the work center's parameters: click on the work center on the simulation window.

To copy a work center: drag it on the simulation window with the CTRL key pressed down.

To link a feeding object to it, drag from that object to the work center with the SHIFT key pressed down.

To link from the object to another, drag from it to the object it feeds with the SHIFT key pressed down.

As with all other simulation objects, if you want to access its parameters, simply click on the work center on the screen. When you have clicked on the work center
you will be shown the work center’s dialog box. This contains some of the parameters, and buttons which access the many other parameters available.

Each individual work center can be documented using the memo button. The image used to display the work center on the screen can be selected from disk or drawn using the graphics button. The image used can be different when the work center is in different states.

The time a work center takes to perform its tasks is determined by the settings in the Timing panel. If the work center does not work at this rate continuously (because of breakdowns or other in-efficiencies) specify this using the Efficiency button. (See page 102).

Work centers compete with each other for resources which are in short supply. Click the resource button to select which resources each work center needs. Each work center can be given a priority for obtaining its desired resources.

The routing-in and routing-out buttons control how the work center selects work items it gathers from feeding simulation objects and how it should feed complete work items onto others.

If the work center is relatively simple (for example it is not "Interruptible", or of "High Volume" or does not make use of any other special features) you can quickly find out how your model will perform with different numbers of this work center in your operation by using the quick replicate facility (see page 68).

To see a list of the actual work items in the work center at any moment in the simulation run click the contents button. This lets you also see the values of the work items' attributes. Actions let the work center change the values of attributes of work items as they finish their work at the work center.

The high volume check box changes the way the work center operates into SIMUL8’s high volume mode for handling BPR and FMCG type applications. (See page 120).
Storage Bins
A storage bin is a place where work to be done can wait until appropriate resources or work centers are available.

A storage bin is one of the fundamental objects which make up the structure of your simulation model.

Storage bins (queues) can be displayed as a fixed icon (with or without a count of the work items within it and its name), or as an animated queue, or as a “tank”. An animated queue shows every item in the queue (using the icon which has been chosen for displaying that type of work item). A tank shows the queue as a tank of liquid. The image on the screen fills up as work items build up in the queue.

For purposes of animation, storage bins are assumed to be "close to" the work centers they feed. This means the work items take no time to travel from a queue to a work center. However, work items may, take time to travel from a work center to a storage bin. (For information about “traveling time” see page 89).

Work Items can be allowed to "expire" while in a storage bin. For example if people enter a shop and are willing to queue for up to 5 minutes, set the SHELF LIFE in the storage bin dialog to 5 minutes. (See page 83).

When simulating some processes (e.g. an oven) you might want to use a storage bin to act as a simple work center but one which can contain many work items. However you would want to set a minimum time that the work items must stay in the storage. Use the Min Wait Time box to set this minimum.

Some times you may not want your model to start its run completely empty of work items. In this case you may place work items into storage bins.
Click START-UP in the Storage Bins' dialog. You can specify which type of work items this storage bin is to be initialized with and how many of the work items should be placed here.

Sometimes the particular logic of your simulation model will mean that you do not want these storage bins to be part of your normal running model. In this case you can make them invisible (set the invisible check box in the storage graphics dialog). If you need to set up attributes of the work items which are initially in your model when it starts to run click the ACTIONS button in the startup dialog. These attribute actions will be used to set the value of the attributes as the work items are placed in the storage bin (while the model is being reset).

If the queue has limited capacity (for example it is a mechanical buffer), type the maximum number of items into the capacity box in the storage bin dialog (reached by simply clicking on the storage bin on the simulation window).

Normally work items join the back of a queue and do not leave until they reach the front. Exceptions are:

- When items whose shelf life has expired are removed by an "expired only" work center.

- When items do not join the back of the queue because the queue is prioritized.

The storage area dialog box allows a prioritization option. If checked you are given the opportunity to select an attribute, the value of which is used to prioritize the work items at this storage bin. The higher the value of a work item's attribute, the higher its priority (the nearer the front of the queue it will be placed).

Zero is the lowest possible priority (negative values have the same priority as zero). Work items without the selected attribute are treated a zero priority.
This attribute (which can have any name) should not be confused with the "priority attribute" (See page 96)

**Conveyors**

Conveyors are a means of transporting work items around the system you are simulating (see also page 89).

Conveyors can be used in ways similar to Work Centers but they also have similarities to Storage Bins.

Conveyors can be displayed as standard icons, file based icons/bitmaps or as animations showing the location of each work item on the conveyor. Conveyors are animated in the same way as storage bins except that in moving the animated display with the mouse the two handles on the screen represent the front (destination) and end (source) of the conveyor, rather than the first and second items in the queue.

The number of work items which can be on a conveyor at any one time is determined by the length of the conveyor and the size of the work item.

The speed of a conveyor, combined with its length affects how long it takes a work item to travel from the start to the end of a conveyor.

A conveyor actively pushes work which reaches its end onto the next available simulation object. Use the Routing Out option in the conveyor dialog to affect how the conveyor decides where to send work items.

**Work Entry Points**

A work entry point is a place where work to be done appears in your model for the first time.
You can have as many "work entry points" as you like. Each can feed work in using different statistical distributions (if required). Each can feed in Work Item Types of different types (if required).

To create a "work enters point", click the button and then click the mouse at the location you would like it to appear in the simulation window.

Connect it to other objects using routing arrows.

If you want to set the values of attributes on your work items as they enter the model use actions at the work entry point.

**Blocked Work Entry Points:**
Note that work items will be "lost" (i.e. thrown away) if they cannot immediately proceed from the Work Enters Point to the next object on the screen. A warning will be given when this happens unless you have switched off these warnings using the check box in the work entry point dialog. It is usually best to connect Work Entry Points to Storage Bins first, otherwise, if two work items arrive in quick succession, and the work center you have routed them to cannot accept them (if it is busy) then the work item will be discarded, and will not enter your model. Sometimes you might want this (if, for example, you want to simulate customers being lost because a phone system is busy).

**Batching**
This controls the number of items which arrive at one time. If an "arrival" occurs at time 10:30 and the number of items set to leave the "Work Enters System Point" is "Fixed" 20, then 20 items of the specified work item type will arrive at 10:30.

The number of items in a batch can be determined by a distribution.

**Work Exit Points**
A Work Exit Point is a place where work that is complete (or otherwise "finished") leaves your model.
At the point in time when each work item leaves, data is recorded about how long it has spent in the model (from the time when it entered through a "Work Enters Point ").

**Resources**

"Resources" are items in the simulation model which are required at work centers in order for the work center to work on a work item.

Work centers cannot start work until both a work item is available and the specified resources are also available.

Resources are shared between all the work centers which used them. They are also know as "shared" or "floating" resources because they "float" between the work centers which require them.

Examples of "resources" are:
- Labor
- Special Fitments for machines
- Beds in a hospital ward

**Resource Shift Patterns**

Resources can be available for only part of the time a simulation model is running if you make use of shift patterns to describe their availability. Use the resource shift availability dialog to describe how many of the resource are available during each shift.

Click on a shift name, then click on the availability number box and change this number to the amount of the resource you want to be available during this named shift.
To adjust the times when shifts are "on", add more shifts, or change their names use the Shift Patterns dialog from Clock / Shifts in the main menu. (See page 118).

Pooled Resources

Resources can be pooled so that (for example) Fitters can fit and Operators can operate but either Operators or Fitters can polish. A pooled resource is one which does not really exist but instead is a combination of other resources.

If task A can be performed by people A and task B can be performed by people B and task C can be performed by either A or B people then create a pooled resource called C and used the Pooled check box in the resource dialog and button to set that either A or B can be used whenever a work center is seeking resource type C.

To make use of a resource at a work center, in the work center dialog, click the resource button and ADD the resource to the list of resources required at that work center.

Multiple Resources at a Work Center

To specify that more than one of a particular resource is required for a task at a work center, click the DETAIL button in this resource selection dialog.

In the Min box specify the minimum number required (this defaults to 1 and there is no need to use this dialog if you simply need one resource of each type specified at the work center). If it is possible to reduce the time taken to perform the task by using additional resources then type the maximum number which can be used in the Max box. The time take to perform the task will be automatically adjusted as follows.
Actual time = Time with Min Number * (Min Number/Actual Number of Resources Obtained).

Actual Number of Resources Obtained is always as many of the resource type as SIMUL8 finds free at the time the job starts (subject to the number in the Max box).
Time with Min Number is the time sampled from the work center's Timing box.

Converting Resources into Different Types

Resources can be released as a different "resource".

This makes it possible to indicate particular conditions to parts of the simulation model.

For example: If a doctor always completes two activities in sequence ("operate" and "clean up") then the operating theater work center could require a resource "doctor" but release this resource as "doctor A". A separate work center (perhaps called "cleansing") could require a resource "doctor A" and release a resource "doctor" - who is then ready to perform another operation.

When are resources released?

Resources are normally released by a work center as soon as the task at the work center is complete.

However, if a work center is blocked this means that the resource is freed up before the work center is available to do other work. In cases when resources are being used to control a sequence of tasks at one real life work center which is being simulated with a number of SIMUL8 work centers this might be undesirable. So this 'immediate release' facility can be switched off for particular work centers.
Click the RESOURCE button in the work center dialog and unclick the check box for 'Release resources as soon as task complete'

**When are resources required?**

Resources must be at a work center before the work center can start work and work items will not normally move to a work center before it can start work.

However, if "collect routing" is used to collect a number of work items into a work center for a task, then it is possible to de-select the default "require resources before collecting any work items" check box in the Resources dialog box. Reach this dialog box from the work center dialog box.

**Invisible objects**

Any object in your simulation model can be made invisible (so it is not seen when the model runs). You can do this so that you can remove information from the screen which is not important to see as the model runs. (See also: Simulation Sub-Windows, which let you place sections of your model in different windows, and only view those aspects which are relevant to the current decision).

For example you might include a work center to, say, control routing which does not really exist in the situation you are modeling (perhaps because the work items are intelligent and make their own choice about where to go!).

**Object List Dialog Box**

For fast access to an object just click it on the simulation window.

The Object List Dialog Box is a list of some or all the objects which form the structure of your simulation model. Access this directly from the main menu (OBJECTS and then any of the types or one of the search options) or also from many other dialog boxes which have options to link to other objects.
It is usually displayed so that you can select the particular object you need.

It can include the
- Work Centers
- Conveyors
- Storage Bins (Queues)
- Resources (People and other items required to do work at Work Centers)
- Work Entry Points
- Work Exit Points

It can also include all the invisible objects too.

Don’t use this list to create new objects: To create a new object click the appropriate button on the tool bar and then click on the simulation window where you would like the object to appear.

This list can also be used to delete objects. You can select a number of objects in the list at one time (click on them with SHIFT or CONTROL pressed on the keyboard - just like in File-Manager) and then delete them all by clicking the DELETE button.

You can get to see the details of an object more quickly by simply clicking on it on the simulation window.

**Deleting Objects**

Objects can be deleted using the above “Objects List” but can also be deleted by clicking on them, and then clicking the ERASE button in the dialog box which appears, or by dragging a box around them and pressing the delete key on the keyboard.
Quick Replicate

This lets you quickly try alternative numbers of work centers which perform the same tasks.

Note that using this facility prevents access to many other work center facilities (e.g. break downs and utilization results information).

It is ONLY designed as a quick way to assess required capacity in an organization. In most cases you should simply copy work centers directly on the screen or use the Groups facility to represent multiple copies of the same work center (see page 122).

To use quick replicate click the REPLICATE button in the work center dialog. If this button is grayed out then it is not available (because you are already using one of the facilities at this work center which cannot be validly modeled by using quick replicate). In this case use the duplication wizard (page 123).
Hierarchical Modeling

Sub-Simulation Windows
The simulation window does not itself have to show all the detail in your model.

With a complex model you may want to summaries your model at different levels. Achieve this by creating sub-windows and sub-windows within subwindows.....to any level of depth you require.

When a sub-window is closed its contents are represented by a single icon on its parent window.

Create a sub-window by dragging a box around the objects you would like in the sub-window and clicking the RIGHT mouse button.

Choose: MAKE SUB-WINDOW.

All the objects inside the box are moved to the new subwindow and are replaced on the parent window by a single icon.

If you close the subwindow the only representation of the objects in the subwindow is the single icon on its parent:

You can open a subwindow by clicking on its icon.

The default icon used to display a simulation window can be changed (click on the window’s top left icon and choose DETAILS in the menu.) Any image can be used (or you can create a new image or load an icon or bitmap from disk). You can also use this dialog to change the sub window’s title.
When the mouse is on top of an icon for a sub-window it changes to be an arrow with a window, to make it easy to distinguish icons for windows from icons for simulation objects.

You can move a sub-window so that it has a different parent window by simply dragging its icon on top of a different currently open simulation window.

**Removing Sub-Simulation Windows**

These windows can be removed by clicking the close box (just once) to bring up the system menu and selecting the DELETE SUB WINDOWS menu item.

Deleting a sub window places all the objects which are in the window onto the window's parent window (in the same location as they appear on the screen before the sub-window was deleted).

You cannot delete a sub-window if it currently has one or more sub-windows itself. Delete these first. This is so that the objects which are moved up a level know where to appear on the screen.

You cannot delete a sub-window unless its parent is currently open. Open its parent first. This is because the icons in the lower level would not know where to appear in the parent level.

**Closing a sub-window**

If you double-click a sub-simulation window's close box (or top left icon), or choose the CLOSE menu item, or minimize a sub-simulation window the window will disappear and you will only see the icon which represents all the objects on its parent window.
To actually delete a sub-window (and return all its objects to the parent window) see: Removing a sub-simulation window.
Important Control Features

This section details important features in SIMUL8 which aid your building of models. We assume here that you are familiar with using WINDOWS and with the features which are common to most windows packages. If not we strongly suggested you run the excellent Windows tutorial which comes with windows. (From Program Manager, or from SIMUL8, click the mouse on the HELP menu item and then on the WINDOWS TUTORIAL menu item.)

Select, Move and Copy Objects on Screen

To see the detail of a simulation object on the screen, just click on it.

If you want to move an object, drag it with the mouse.

To copy an object, drag the object to where you would like its copy to appear with the CONTROL key pressed on the keyboard (this is the standard WINDOWS method of indicating you want to COPY something).

You can copy or move many objects at the same time if you select them all first.

Select a number of objects by dragging a box around them.

You can then drag the box to move the objects in it (or copy them by also holding down CONTROL).

Select additional objects by either clicking them with the CONTROL key pressed (just like you would in WINDOWS file manager) or by dragging more boxes around objects with CONTROL pressed. (As long as the start of the drag is outside an existing selection, you can go on making as many selections as you like.)
Then just one drag (from within a selection) will move or copy as many objects as you like.

**Move**
Moving an object changes its place on the simulation window. It remains connected to the same objects as before.

**Copy**
Copying an object produces a new object which is connected to the same objects as the object from which it was copied. The new object also has the same parameters (e.g. breakdowns, maximum size etc.) as the object from which it was copied.

When a number of objects are copied, the new objects are connected to each other, where appropriate, to maintain the same structure as existed in the original group of objects which were copied.

New objects are given the same names as the originals except that a number is added to the end of the name to make the name unique. (If the object name already has a number at the end, then this number is changed).

Normally, when groups of objects are copied, the resulting objects are connected in parallel with the original object. SIMUL8 does however detect when it is appropriate to connect objects in series.
For example:

If you want to create a process simulation which looks like this:

Then the quickest way is to create:

Then copy the last two objects by selecting them:

and dragging them with CTRL pressed so the first of the new set of objects (the new storage bin) is close to the last of the old set (Work Center 1):

SIMUL8 automatically creates the new routing arrows so the new set of objects is in series with the old one when the old set has only one input route and no output routes AND you place the new first objects close to the old last object.

Now keep copying the last pair as often as you need.
(To make a large number of copies of simulation objects it may be easier to use the Duplication Wizard - see page 123).

**Aligning Objects on the display**

To align a number of objects select the objects to be aligned and click the right mouse button. Choose Align.

**UNDO**

Undo **ONLY** works if UNDO is enabled. (By default UNDO is enabled).

Click the Undo button (undo) to remove the changes which have most recently been made to your model. **SIMUL8** stores up to 20 previous sets of changes. To select any of these (rather than just the most recent) use EDIT / Undo..... from the main menu. The model reverts back to the point mentioned in the Undo menu command.

For example, if you have just edited an object, then EDIT / UNDO would say:

[Image of undo icon]

and clicking this menu item would return the model to the point immediately before you edited the object.
**Redo**

Redo lets you put back the changes which were removed by the most recent Undo operation. Access Redo from the Edit menu.

**Enable/Disable UNDO**

Enable or disable undo using the File/Preferences menu.

It is a good idea to keep Undo enabled in case you should make a mistake which is time-consuming to recover from. However, on slow computers, or with large models, Undo can delay your work because of the large amount of information which the undo facility needs to handle.

If you find your model is slow at opening dialog boxes to edit objects on the screen, try disabling undo.

The default setting for Undo cause it to always reset the simulation clock to the start of the simulation run after you “undo”. File/Preferences can also switch this off. However, to operate in this mode SIMUL8 has to store all “results” information when it saves information about how to undo. With a large model which has run for a long time this can use considerable amounts of disk space.

**Find**

OBJECTS / FIND in the main menu can be used to search for any text in the titles of any simulation objects. If an object is found its name is displayed in the list which can then be used to access the object’s details.

You can also access the FIND facility by pressing CTRL-F on the keyboard.
Preferences

The **FILE / PREFERENCES** menu item lets you select a number of options which customize the way **SIMUL8** works with your model.
The Simulation Time

**SIMUL8**, like all simulation packages, contains a clock against which the model runs. (Everything in the model is adjusted to match with what would happen in the real situation at the time which is currently shown on the clock). The clock runs much faster than it does in real life. This enables you to watch what happens in the situation you are modeling in fast-time - so you can gain insights as to how to improve it.

**Clock**

The format of the clock can be changed using the CLOCK / CLOCK FORMAT menu or by "double-clicking" the clock window.

The clock can be digital or analog and can show information about the day and week too.

**Results Collection Period**

The results collection time is how long the model should be run before automatically stopping while collecting results. The model will not start to collect results until the warm-up-time is finished.

Set the results collection period from CLOCK / RESULTS COLLECTION in the main menu.

For a detailed discussion of this aspect of simulation modeling see the section “Simulation - The Technique - A Guide”

**Warm Up Period**

The warm up time is the time that the model will run before starting to collect results. This allows the queues (and other aspects in the model) to get into conditions which are typical of normal running conditions in the system you are modeling.
Sometimes you do not need a warm up time. An example of such a situation would be when you are modeling a shop which starts and ends the day empty. A warm up time would be required when modeling a week in a factory, when the factory starts on Monday morning in the exact same state as it finished on Friday evening.

Set the warm up period from CLOCK / WARM UP in the main menu.

For a detailed discussion of this aspect of simulation modeling see the section “Simulation - The Technique - A Guide”

**High Speed Clock**

Changing and updating the clock on the display can take a significant proportion of run time. While the most aesthetically pleasing view of the clock is achieved by keeping the clock exactly in line with events on the screen, this is slower than using the "automatic" clock refresh mode.

In Automatic mode the clock is updated only once every 1/10th of a second.

Use Clock/Format and then the MORE button to change the clock update mode.

An accelerated version of the high speed clock is automatically used when the simulation window graphics are not being updated (when the speed control is set to full speed).
Routing

Routing lets you control the path taken by each individual work item through the simulation model.

You have an opportunity to control routing as work items enter and leave objects on the screen.

Click the Routing In or Routing Out buttons in the work center dialog box.

Routing can be controlled at work centers and conveyors but can also be controlled at other objects by using "invisible" work centers.

Drawing the Routing

Draw the route your work items take by using the main simulation window. (Please note that in the case of very simple models SIMUL8 will help you by drawing the route for you. However, as soon as it realizes its guesses about the route are not helping you, it will stop doing this).

There are two ways to add and delete routing arrows. Which way you use depends on your own preferences for use of the computer and the mouse.

The first involves using the SHIFT key on the keyboard:

Press the SHIFT key on the keyboard and drag with the mouse from the first (source) work center (or other simulation object) to the second (destination) work center. Let go of the left mouse button when the mouse arrow is over the destination work center.
The second (and many people find this easier) is to click the Route Drawing Mode button (\[\text{Route Drawing Mode} \]).

Click the Route Drawing Mode button so it is down, then click on the first (source) work center (or other simulation object) and then on the second (destination) work center to describe the route. You can click on a series of simulation objects to create a series of routing arrows, then click the Route Drawing Mode button again to release SIMUL8 from this mode.

You can also connect / disconnect routings inside the Routing-In and Routing-Out dialogs for Work Centers.

**Deleting a route arrow**

To remove a route simply redraw it (using either of the above methods).

Either:

Drag (with SHIFT on the keyboard pressed) between the source and the destination work centers.

Or:

Click the Route Drawing Mode button so it is down, then click on the first (source) work center and then on the second (destination) work center.

**Switch Route Lines on/off**

The route lines can be switched on and off at any time when the model is not running by clicking the \[\text{Switch Route Lines on/off} \] button.
Routing In to a Work Center

Click the Routing In button in the Work Center dialog box.

This lets you control where a work center gets the work items on which it works.

The following routing methods are available when selecting work items:

**Priority in determining the Work Item to select for a Work Center**

Selects a work item from the first place in the list of objects which feed this work center. If no work item is available from that place, then the 2nd object in the list is tried, etc. etc.

**Collect work items**

This method of "routing" work items into a work center lets you collect a number of work items from a number of places.

For example you might want 12 bottles of whisky and 1 box.

When they have all been collected the work center will start work. Only one work item will be worked on and by default this is the first work item in the list of work items.

By default a work center does not start collecting the items it needs until it has all the resources it needs.
Expired Only

Use this Routing-In discipline to tell the work center to ONLY accept work items which have been in a storage bin for more than the “shelf life” set for that storage bin.

Shelf Life

Use SHELF LIFE (or max waiting time) in storage bins to control how long a work item can usefully stay in a queue.

Setting the shelf life by itself is not sufficient to change the way your model works - you must also say what is to happen to the work item when its time expires.

Set up a route arrow to a work center which has its ROUTING IN set to "EXPIRED ONLY". This work center will then not normally take work items from the queue (storage bin). It will only accept work items whose time has expired.

Time expired work items will be sent to any other destination from the storage bin if they reach the front of the queue without being removed by a work center which is set to search for expired items (this might happen if your "expired items" route is busy). If you want to avoid this then your "expired items" work center should simply pass the work items onto wherever they need to go (with a zero processing time) to avoid the possibility of blocking expired items.

You can make shelf life a variable which is different for each work item by using a work item's attribute to contain the amount of time before the work item should expire. Tell a storage bin to use a particular attribute for this by double-clicking the box in which you would type a fixed shelf life.
Interruptible Work Centers

Normally work centers always finish the task they are working on before they consider starting work on a second work item. However, in some environments low priority tasks are performed which can be interrupted by other more important jobs.

Set a work center to be “interruptible” by checking the check box in the Routing In dialog (it is items of high priority coming into the work center which may interrupt existing work). Each interruptible work center must have a storage bin associated with it which will automatically receive interrupted work items. (Such items do not leave the work center using the normal “routing out” mechanism). You will be asked for the storage bin to use when you check the “interruptible” check box.

Only work items which have a value in their “priority attribute” which is greater than the value in the work item being processed will actually interrupt the work. For a work center to be interruptible it must be fed by an object which actively pushes out work (like a work center). Storage Bins do not actively feed work onward, they simply wait for it to be pulled. If you need to have work items in a storage bin interrupt work at a work center, simply feed the work items to an intermediate work center which takes zero time to process and feeds directly onto the interruptible work center.

Attribute batching

Attribute batching lets you batch together work items for a task to be performed at a work center. You select an attribute which the work items will all know about (for example "Volume").

As work arrives at the work center SIMUL8 adds up the values in these attributes until they reach at least the minimum batch size, but not more than the maximum batch size.
If the work center can take a maximum of 200 litres of "volume" and you are prepared to work with any batch of at least 50 litres then you would set the batching parameters to 50 and 200 and "use" attribute "Volume"

In the routing in dialog click the USE ATTRIBUTE BATCHING check box and select an attribute to be used.

**Batching by type**

This method of routing in to a work center tells the work center to only accept work items which have a certain value in one of their attributes. Select an attribute which will contain a value which in your model means the type of work which it is (e.g. product code, illness type).

When the work center cannot find any more of this type of work (in any of the objects which feed it) then it will automatically look for a new type. When searching for a work item of the same type the work center will look anywhere in the queues which feed it (not just at the head of the queue).

Negative and zero values are not allowed (they will cause any product to be accepted next).

**Routing Out of a Work Center**

Use the routing-out dialog to tell **SIMUL8** where work items should go next after being at a work center.

If one work center is connected to the next with one simple connection (arrow) then the routing dialog has no options.

If there is a choice of two or more locations to go to then the routing dialog box can be used to change the routing rules.
Available routing rules are:

- Circulate (the default)
- Uniform
- Percent
- Priority
- Attribute
- Shortest Queue

**Changing the work item during a task at a work center**

To have a work item of a particular type enter a work center and a different one leave, click the Routing Out dialog box in the work center and change the work item type. By default the work item leaving a work center is the same as the one which enters.

If two work item types have attributes with the same name, the values in the attributes are copied across from the work item which entered the work center to the work item which leaves the work center. Statistics on the amount of time a work item has been in the system are also transferred. This means that "widget unfinished" can change into "widget finished" and retain all data if required.

Data is NOT transferred if "collect" Routing In is used.

**List of Destinations**

These are normally set by drawing the routing on the simulation window. However, you can directly add or remove simulation objects from this list using the ADD and REMOVE buttons in the routing out dialog.

Depending on the routing out discipline, the sequence of the objects in the routing out list can be of some significance. For example, “priority routing” attempt to send work items to the first object in the list. To change the sequence, click on one of the name of an object in the list and use the UP and DOWN buttons.
To change the percentage of work items which are routed to each destination (when using the percentage discipline) click on the destination you wish to change and enter the new percentage in the percentage box.

**Circulate Routing Rule**

The first work item will go to the first destination in the list, the second work item to the second,....., the Nth work item to the Nth destination on the list. The Nth+1 work item with go to the first destination etc. etc. (Thus the decision about where to go next "circulates" through the list).

**Priority Routing (First Destination = First Choice)**

The priority routing rule sends all work items to the first destination in the list, unless this is blocked, when the 2nd is tried...and so on.

**Uniform Routing Rule**

The destination is decided randomly with each destination having an equal chance of being chosen.

**Percent Routing Rule**

The destination work center is decided randomly (like "uniform ") except that the exact percentage going to each destination can be specified.

To change the percentage of work items which are routed to each destination click on the destination you wish to change.

**Percentage Automatically = 100**

If the percentages do not add to 100 they will be pro-rata adjusted when you click OK to exit from the dialog box.
Alternatively you can click the "=100" button to see the adjusted.

**Shortest Queue**
This routing method sends Work Items to the queue which is shortest. If one or more of the places to which work items can be sent is a Work Center then work items will go to the work center if it can currently take more work. i.e. - A work center which can take work immediately counts the same as a queue with nothing in it.

"Preference Routing " does not apply to this type of routing rule.

**Preference Routing**
In addition it is possible to specify what should happen if the chosen route is blocked. "Preference Only” means that if the desired route is blocked then the same rule will be used again to find another route. Switching "Preference Only” off means that once a route has been chosen (using the rule), then for this particular work item leaving the work center, it must go down the chosen route and it will wait until that route becomes available.
Traveling Time

Work Items travel between objects on the screen automatically. Their travel is animated so that you can see the movement of work around the system you are modeling, however the travel does not necessarily use up "simulation time".

Some objects are always assumed to be immediately adjacent to the objects they feed. In this case you can still see work items move between the objects, however, simulation time is not used during this traveling (the clock freezes while the work item moves between the two objects).

Travel between objects takes up simulation time when items move from:

- WORK CENTER to STORAGE BIN
- WORK CENTER to WORK EXIT POINT

As you adjust the speed bar you will see less animation, this enables your computer to make the model run faster. However, the logic of your model is unchanged. Adjusting the speed of your model with the speed control has no effect on the results of the model.

You can tell SIMUL8 to automatically set up travel times based on their distance apart on the screen.

Use the GRAPHICS / DISTANCE menu to change the parameters used to define the time it takes work items to travel between some objects in the model.

Objects which feed work centers are assumed to be close to the work center and so no time delay is introduced between the work item leaving any object and reaching a work center. However, work centers may feed storage bins which are some distance away and it may take some time for work items to travel to these locations.
To change the distance between a work center and the object it feeds click TRAVEL in the ROUTING OUT dialog. (If the button is gray, then click on the name of the relevant simulation object in the routing out dialog).

A travel link is a route (gray arrow) along which it takes time for work items to travel.

If you set the travel time to zero then the two objects are directly linked (without any traveling time required between the objects) - this is the same as setting up the routing links with "automatically set up traveling times" NOT checked.

You can switch off the automatic set-up of travel times (so all links take zero time) using the File / Preferences / Distance dialog. You can also use this to change all existing non-zero travel times to zero. If you want to recalculate all existing non-zero travel times use the File / Preferences / Distance dialog.
Attributes

Attributes can be attached to any "work item" going through the simulation model. For example you might attach an attribute "Illness" to work item type: "Patients" and set this attribute to a code number as they enter your simulation model of a hospital. Each code number would represent a type of illness and could be used to route the patients through the different "work centers" in the simulation model.

Attributes can contain either text or numbers.

Tell work centers to change the values of attributes using the ATTRIBUTE ACTION button in the work center dialog.

Attributes can be used to tell work centers which distributions to use for sampling work times.

Adding attributes to Work Items

Attributes can be used to route work items, to collect data and for a number of other purposes.

To enable a work item to be controlled by an attribute's value or to be able to collect data in an attribute, you need to first add the attribute to the work item.

Use the main menu OBJECTS / WORK ITEMS to select the work item type. Click the DETAIL button. Now click the ADD button to be able to select an attribute to add to this work item type.

If you forget to add an attribute to a work item type, SIMUL8 will identify this (usually once the model is running), and will give you an opportunity to have the mistake corrected automatically.
Attributes Containing Numbers
Values in attributes can be fixed numbers, come from distributions or come from
the results of calculations.

Use numbers in attributes to, for example, affect the route work items take through
your model.

Attributes Containing Text
Text in attributes can be used, for example, to select which distribution to use at
work centers and how the work items are displayed.

Attributes List
Use the attributes list dialog box to select the attribute you want to use.

Click the NEW button to create a new attribute which you can then use.

To select an attribute to use, click the name of the attribute, then click OK.

If you want to find out more about a particular attribute, click the name of the
attribute, then click the detail button.

Attributes can only be deleted if they are not in use by any work item or work
center.

Attribute Actions
Attribute Actions allow work centers to change the values in the attributes of work
items.
Use the Attribute Action dialog box to tell a work center how to change a work item's attributes as it comes through the work center.

**List of Action Attributes**

Attributes in this list are controlled by this work center. If a work item is processed by this work center and it has one or more of these attributes then the attribute will be changed when the work center finishes the processing time for the work item. Other attributes attached to a work item are ignored.

If a work item type does not have one of these attributes, then that "action" has no affect on the work item.

The highlighted attribute is the one whose details appear in the "action" box to the right of the dialog box.

Use the ADD and REMOVE buttons to select which attributes are affected by this work center.

**Add to the list of Action Attributes**

Use the ADD button to add to the list of attributes affected by this work center. Clicking the add button brings up a list of all attributes which exist in the model. At this stage you can add to this list.

Remember that a work center cannot change an attribute if a work item does not have that attribute!
Remove Action

This button removes the highlighted attribute from the list of attributes which can be changed by the work center.

Attribute Value

The attribute's value will be set to the value supplied from this dialog box:

In some cases you will simply want to set the attribute to a fixed value, in which case set "Distribution" to "Fixed". In other cases you may want the value to be set to a value which is sampled from a distribution. The distribution you sample from can be any which have been created anywhere in your model.

Attribute Text - Setting

Use this dialog box to change the text which is in an attribute as it passes through a work center's "actions".

The text you enter in the field in this dialog box will be placed in the named attribute.
This text can later be tested in other work centers and can be used to determine which distribution should be used to sample times for the work item.

Attributes control Work Item Images
To use attributes to change the image used to display a work item as it completes work at a work center ensure one of the work items attributes is set in one of the ways described below.

The value of the attribute DOES NOT have to be changed by the work center's ACTIONS but the attribute does need to be listed in the ACTIONS (the particular attribute can be set to NO CHANGE if desired).

Text attributes:

If a work item has a text attribute with the name IMAGE and that attribute is listed in a work center's ACTIONS then the image for the work item will be changed to the image named in the attribute. (SIMUL8 looks for an image whose name matches the text contained in the attribute of the work item).

Number attributes:

If a work item has a number attribute with a name beginning with IMAGE: and that attribute is listed in a work center's ACTIONS then SIMUL8 will take the rest of the attribute's name and add to it the value in the attribute and then look for an image with that name.

For example an attribute called IMAGE:Product containing the number 4 will cause the work center to change the work items image to Product4 (if that image exists).

In all cases, the IMAGE button in the work center GRAPHICS dialog has priority over any changes based on attributes.
Special Attributes

Some attribute names have a special meaning in SIMUL8. We have already seen how attributes can contain the names of distributions to be used to sample processing time and how attributes whose names beginning with “IMAGE” can be used to change the look of a work item.

Priority attribute

Attributes are used to control priorities in a number of ways. In many cases any name of attribute can be used because the particular facility will ask the name of the attribute to use. The “Priority” attribute (which must be called “PRIORITY”) is used by SIMUL8 automatically to determine whether a work item can interrupt another work item.

Due attribute

The Due attribute (always called “DUE”) is set, if it exists, to the amount of time which was remaining when a work item was interrupted at a work center.

Quantity Attribute

The Quantity Attribute determines the size of batches (one batch = one work item) at High Volume work centers.

Routing using Attributes

To route a work item depending on attribute value: Select ATTRIBUTE in the routing dialog and click the DETAIL button to select which attribute to use. The work items to be routed must have this attribute attached to them. If an attribute is missing from a work item type which goes through the work center, then the default routine rules (circulate) will be used.
The value of the attribute should be between 1 and \(N\), where \(N\) is the number of alternative routing destinations.

If the value is \(> N\) the \(N\)th destination will be used. Similarly the 1st is used if the attribute is zero or less.

For an example of routing using attributes see the example model: ATACTION.S8
Animation

SIMUL8 contains many animation features which are designed to make your model look like the real world it represents.

This section discusses most of the aspects of SIMUL8 which relate to animation.

When you build a model SIMUL8 sets the model up so that you will be able to see what is happening as the model runs. In particular work items can be seen moving between objects in the model and the number of work items currently at each work center is displayed.

The speed at which work items move across the screen is dependent on the position of the speed control.

You can show more animation:

In particular you can animate queues (storage bins) and conveyors so that you can see each work item in them as the model runs.

You can animate work centers, so that they move about as they change state (for example a work center which is working can be shown differently if it is waiting for work, or broken down). To achieve this click the GRAPHICS button inside the work center’s dialog and select a different image to be displayed in each condition. See the example model: ANIMAT2.

You can also show work items which are at work centers (and if the work center's image is also displayed then the image for the work center and the work item are automatically combined). Again, check the appropriate settings in the work center’s GRAPHICS dialog.
Work centers can change the way a work item looks. To achieve this click the work item IMAGE button in the graphics dialog or use the value of attributes to change the image (see the section above on attributes).

All objects and work items are displayed on the screen using images. An image is a special type of object within SIMUL8. It has a name and information about the way it looks on the screen. Manage the images in your model using GRAPHICS / IMAGES from the main menu.

When work items move between works centers there are two different ways in which this can happen (and this affects the way in which they are animated). Travel between some types of object takes (or can take) time. In this case work items are seen moving between the objects as the clock turns and as other aspects of the model are also updated on the screen. In other cases (where work items move between objects without using simulation time) it is still possible (sometimes desirable) to see the work items "move" between these object. This can help understanding of the dynamics of the model even though such movement is actually "instant" in real life (and "instant" in the logic of the model). This movement is shown on the screen by briefly stopping the clock and moving the work item visually on the screen between the two object (although with the speed control set at around half way or more you will hardly notice that the clock has been stopped).

Objects do not have to be displayed. Often models contain objects which are included in the model purely to control the logic rather than actually physically existing in the real world. (Examples are rules which operate in the real world you are modeling - these don't "physically" exist). In this case you can set the object to "invisible".
Priority

A number of methods exist for giving aspects of your simulation model priority over other aspects.

Prioritizing Work

Normally all work items have equal priority in queues (storage areas).

The storage area dialog box allows a prioritization option. If checked you are given the opportunity to select an attribute, the value of which is used to prioritize the work items. The higher the value of a work item's attribute, the higher its priority (the nearer the front of the queue it will be placed).

Zero is the lowest possible priority (negative values have the same priority as zero). Work items without the selected attribute are treated as zero priority.

Prioritizing Work Centers

If two or more work centers are ready to start work (work items are available to be worked on) but they cannot start because a floating (shared) resource such as labor is unavailable then they have to compete to get this resource.

When a unit of resource becomes available the work center which gets it is determined by the work center's priority.

By default all work centers have the same priority (50). The higher the priority value, the more important the work center is in getting resources.

Priority values can be any number between 0 (lowest) and 100 (highest).

Set a work center’s priority by clicking PRIORITY in the work center’s dialog box. Normally priority is a fixed value but it can be changed dynamically.
The Priority of a work center can be adjusted dynamically in two ways.

1) Set the work center's priority method to "Attribute" and choose an attribute name to use. The value of this attribute for all work items at the heads of queues feeding the work center will be checked. The highest of these values will be used for prioritizing the work center's request for resources. These attribute values should be in the range 0-100. If a work item does not have the appropriate attribute its value will be treated as zero.

2) If the work center's priority method is set to "Fixed" then this "fixed" value can be changed from EXCEL using the SetPriority procedure (see page 202).

For other prioritization in your SIMUL8 model see pages: 82, 87, 96.
Efficiency and Breakdowns

Both Work Centers and Conveyors can simulate situations where equipment or people work with less than 100% efficiency.

Click the Efficiency button in either the Conveyor or Work Center dialog to specify the efficiency rate.

Simply set the efficiency value (in % terms) and the typical time it takes to repair the stoppage.

Distributions used for Efficiency

With efficiency set to "AUTO" SIMUL8 calculates automatically mean-time-between-failure figures and uses negative exponential distributions for mean-time-between-failure and Erlang 2 distributions for repair time.

Detail / Auto Efficiency

Use AUTO in the breakdowns / Efficiency dialog to be able to quickly specify efficiency information. SIMUL8 does calculations for you and uses typical distributions to generate breakdowns / stoppages. If you do not know typical repair times place a ? in the repair time field. SIMUL8 will calculate a repair time consistent with the efficiency rating you give so that typical repair times are relatively short compared with task durations.

To be able to select distributions yourself click the DETAIL marker.

Stoppages - special cases

Sometimes a stoppage (or breakdown) occurs in such a way that the work center concerned will finish work on the items which are currently waiting at the work center.
At checkouts in a food store customers select the queue to enter, but if the checkout operator's lunch break time comes up while customers are in the queue, these people are still served before the operator leaves the till.

To simulate this situation switch off the "stop work immediately" check box in the breakdowns / efficiency dialog and switch on the "close feeding queues" check box. This stops customers entering the queue until the "breakdown" is over, but lets the work center continue working on remaining items.

You cannot have both check boxes off at the same time (because then the breakdown would have no effect!).

A queue will not be closed if it feeds more than one object in your simulation model.

In cases where the length of the breakdown is typically short compared to the time it takes to finish processing the queue, you may want to have the breakdown timed from the moment the last work item leaves the work center (rather than the time the queue was closed). To achieve this select "Extend Breakdown time by time to process queue".
Random Numbers

Random numbers enable a simulation model to include the variability which occurs in real life.

Each place where random numbers are used within a model uses a separate stream of random numbers. This enables a change to be made to one aspect of a model, without affecting the random occurrences which will happen at other areas.

By default, when the clock is set back to the start of the simulation time, all the random streams are reset to their starting seeds. This means that a second run of the same model will produce the same results.

To change the random numbers which are used, TRIALS / RANDOM SAMPLING from the main menu and change the set of random numbers.

Random number sets are numbered. There are 30000 sets of random number streams available in SIMUL8.

Where random numbers are used

Random numbers are used in:
- Distributions (unless these are "external" or "fixed")
- Distributions themselves are used just about anywhere in SIMUL8 where you can specify a number for any purpose.
- Routing Out (The uniform and percent options)

Auto Change

Set "Auto change" if you want a different set of random number streams used every time you re-run the model.
The RE-RUN button can also be used (instead of the RUN button) to achieve the same effect.

**Antithetic**

Select this to reverse all the random numbers. (At a low level, before random numbers are used in distributions, they are all in the range 0.00000 to 1.00000. Antithetic random numbers are exactly the same streams of numbers except they range from 1.00000 to 0.00000). Changing to the antithetic of a set of random numbers is a quick way to see if the random numbers are having a large impact on the result of a model. (But see also: Accuracy of results)

**Saving Random Number Streams**

Random number streams are normally saved inside your model file. This ensures that you can always return to a random number stream set and re-run the model and get exactly the same behavior from variable parts of your model. However, if you use large numbers of random number streams these can consume a large amount of disk space and memory (this tends to happen if you are using “auto change random numbers” - see page 104). You can tell SIMUL8 not to save the random stream information (and so reduce memory/disk requirements) in the File/Preferences dialog.
Timing and Distributions

All timing and number information in **SIMUL8** is given in terms of "distributions".

A distribution panel in a dialog box looks like this:

![Timing Panel](image)

Typically this enables you to say how long a process takes, but sometimes it is used to set the value of an attribute, or other numbers in **SIMUL8**.

**Distribution Parameters**

In distribution/timing panels the parameters of the distribution vary with the type of distribution.

The first box is almost always the MEAN (or expected value) of the time being specified. So you can expect the average amount of time taken by a process to be the value in the first box. (But beware of the labels on the boxes - some distributions do not have MEAN as a parameter - for example: Uniform Distributions).
Distribution Type

Use the box to select the distribution you want to use to generate the time for the particular activity to which the timing panel relates.

Click the button to see a list of available distributions (click the one you want) or click the NEW button to create a new distribution of your own.

A number of standard distributions are available in SIMUL8.

New Distribution

Click the NEW button to create a distribution of your own.

There are of a number of types:

- Named Distributions
- Probability Profile Distributions
- Time Dependent Distributions
- Time Absolute Distributions
- External Distributions
- Combination Distributions
- Attribute Based Distributions

Distribution Detail

The DETAIL button in timing/distribution panels enables you to check and change the parameters of distribution which you have specially created for your model (including named distributions).

The DETAIL button is only enabled if the distribution type selection box is set to a distribution which you have defined.
Standard Distributions

Statistical distributions provide a method of simulating the variations that occur in timing (and other numbers) in any process involving people or machines or anything in nature.

SIMUL8 allows you to create your own distributions, however certain classical statistical distributions are provided:

- Fixed
- Exponential
- Log Normal
- Negative Exponential
- Normal
- Uniform
- Weibull
- Beta
- Gamma
- Triangular
- Pearson V & VI
Fixed Distribution

A "Fixed" distribution is not a distribution! - it is a fixed value.

A sample from this distribution will always return the value in the "Fixed Value" field.

This facility enables randomness to be removed from an element of the model but also enables "Named" distributions to be used to return fixed values.

Uniform Distributions

Samples from a uniform distribution are equally spread between the lower bound and the upper bound.

The mean of the value sampled is half way between the two values specified.

No sample values occur outside this range.

Named Distributions

In SIMUL8 you can name distributions and then use those in many places. For example if 50 different machines are work at a given rate, but you want to be able
to change that rate easily, then you can tell each machine that it uses a distribution call MACHDIST, and then just change the parameters of MACHDIST.

To create a named distribution use the NEW button in the distribution panel of any of the dialogs which select distributions.

**Ask VBA Check Box**

In the dialog box which sets up a named distribution (where you set the type of distribution etc.) you can also instruct SIMUL8 to "Ask EXCEL’s VBA" about the distribution before each time the distribution is sampled. This enables you to change the parameters of the distribution as the model runs, if you should require this.

**Time Dependent Distributions**

In many situations the speed of a part of a system is related to the time of day. Time dependent distributions enable this to be simulated.

Time dependent distributions are made up of a sequence of "Named " distributions. Create a named distribution for each different part of the day which can be identified as requiring a different underlying distribution.

In any of the random distribution panels in dialog boxes, click NEW and then TIME DEPENDENT DISTRIBUTION, and then NEXT>>.

Click ADD to select a named distribution and to set the time of day at which it will start.

For an example, see the example file TIMEDD.S8.
Re-Sampling Overrun's between time slots

Click this check box in the Time-Dependent Distributions Time-Slot dialog if the value being sampled represents a time and if you want times sampled in the previous cell not to overlap into the current cell.

The default is OFF (not checked). Only use this facility if you are sure you need it because it can distort the results collected from your simulation if used when not required.

Example:

In this example of arrivals to a hospital's "accident and emergency" unit the time interval between arrivals is very different at different times of day. Rush hour is a very busy time.

<table>
<thead>
<tr>
<th>Time Slot Start</th>
<th>Average Interval Between Arrivals (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:00</td>
<td>60</td>
</tr>
<tr>
<td>17:30</td>
<td>5 (Should Re-sample Overruns)</td>
</tr>
<tr>
<td>18:30</td>
<td>30</td>
</tr>
</tbody>
</table>

Without Re-Sampling Overruns set on the 17:30 - 18:30 period a sample taken at time 17:29 could easily result in the arrivals jumping right over the 17:30 - 18:30 period. This is because if an arrival occurred at 17:29 (in the first time slot in the above table) the time to next arrival might be sampled as 69 minutes from 17:29 - and no arrivals would occur in the busy time.

Don't use this facility if there is only a small difference between sample values in adjacent slots because to do so tends to reduce the average sample value (large numbers tend to get rejected unnecessarily).
Don't use this facility if the value sampled is not used for a time or if a sampled time will not affect when the next sample is taken. It usually only applies to distributions used for inter-arrival times (in Work Entry Objects).

**External Distributions**

External distributions allow SIMUL8 to collect timing information from other WINDOWS packages, rather than simply using built-in statistical distributions.

External data may come from either Excel (typically a column of data in EXCEL) or from Visual Basic. For a discussion on how to use external distributions to collect data out of a Visual Basic program see page 221).

This next section deals with using an external distribution to obtain data from Excel.

**External Distributions to collect data from Excel**

A typical use of this facility is to use real data collected from the situation you are modeling, rather than using an equivalent statistical distribution. This is useful if the data does not match any distribution (but see page 114) or you do not want to include any random elements in your model.

Proceed as follows:

1. Run Excel

2. Type, paste, open (read), or otherwise enter your data into a column in Excel.

3. In SIMUL8 go to the dialog box where you want this data used (for example it might be a work center dialog box if the data in Excel represents the times a process has taken to be completed.
Click NEW in the timing/distribution panel. This asks SIMUL8 to create a new type of distribution (one which is not in the drop-down list).

Choose EXTERNAL and click the NEXT button.

Now choose EXCEL and enter the name of the spreadsheet (e.g. Sheet1) and the location where the data starts (this defaults to R1C1 which is the top left corner of the spreadsheet).

Click OK.

Now when you run the model the work center etc. will make use of the data in Excel.

SIMUL8 always starts at the top of the list when you reset the clock back to the start. If SIMUL8 detects it has reached the end of your data it goes back to the beginning of the data and re-uses it. The end of the data is signified by a blank cell (or a cell which contains something other than a number - e.g. text). Formula can appear in cells because SIMUL8 uses the results of the formula.

If the data is in a row instead of a column click the ROW check in the dialog where you entered the start location of the data. If the data is just a single value (perhaps because you simply want SIMUL8 to use a fixed value which you would prefer to come from a spreadsheet rather than be coded directly into your model) then click the SINGLE CELL check box.

If you are using a version of EXCEL earlier than 5 then note that the naming of the spreadsheet is handled differently. With EXCEL 5 and newer, the name you place in the external distribution dialog is the name of the sheet (e.g. Sheet1). This is also the default name for the sheet in previous versions and will work with a new file. When you save your spreadsheet, exit and re-load Excel and your spreadsheet name changes to the name of the file (e.g. C:\DOCS\MYSPREAD.XLS). (It keeps the name Sheet1 in the internal workings of Excel until you exit and reload - this causes
some confusion!). When you reload your spreadsheet make sure you change the entry in the SIMUL8 external distribution dialog to read the complete path name of the file. This only applies to Excel 4 and earlier.

**Probability Profile Distributions**

These distributions can be configured entirely under your control to generate any shape of distribution desired.

Drag the blue bars up or down to change the probability (percentage chance) of each sample value occurring. (Or type in the required percentage)

The figures above each bar are the **percentage chance** of that particular value being sampled from the distribution. (These figures are rounded to the nearest whole number in this picture).

The example model PPDIST illustrates the use of this type of distribution to set attribute values and then route the work items on the attribute values.

Click on one of the blue bars to select it and obtain its percentage and sample value figures in the edit boxes at the bottom of the window.
Drag a bar up or down to change its percentage quickly.

When you do this all other bars are adjusted so the total still adds to 100% (Unless a bar is locked).

**Discrete / Continuous**

Probability Profile distributions are either Discrete or Continuous.

Discrete: The value actually sampled will always be EXACTLY one of the values that appear under the blue bars.

Continuous: The value sampled will be interpolated between the sample values under the blue bars to supply any value between the lowest sample value given and the highest sample value given. This enables you to specify a shape of a smooth distribution without having to enter a percentage change and sample value for an infinite number of bars!!

A red line on the picture indicates the nature of the continuous distribution (if selected).

Click RIGHT on the Probability Profile window in **SIMUL8** to be able to:

- Add new bars
- Delete bars
- Lock/unlock the percentage of a bar.
- Equalize all the unlocked percentages.

Locking/unlocking the percentage of a bar:

This is useful because, having set one percentage you may want it to stay fixed while you change others (and **SIMUL8** normally adjusts all percentage figures so that they add up to 100%).
Locking a bar still lets them all add to 100% - but the locked bars are not included in the adjustment.

Locked bars are indicated by red percentage figures

**Attribute Based Distributions**

An Attribute Based Distribution lets you place information about which distribution should be used for sampling in Work Items' attributes.

For example, if patients in a hospital all have to see a doctor, but the average time they spend with the doctor is dependent on their illness, then you can set up an attribute (called, say, "ILLNESS") which contains the type of illness they have (for example, it might contain phrases like "BROKEN BONE", "YELLOW FEVER" etc.). Then, at the Work Center for the doctor you can instruct **SIMUL8** to use a distribution which will look in an attribute of the relevant work item to find the name of an actual distribution to sample for the time with the doctor.

**Combination Distributions**

A combination distribution adds together samples from a number of individual distributions.

This can be used to create special shapes of distribution or to enable unusual circumstances to be added to normal performance. For example a machine may usually take 5 mins (with a standard deviation of 1 minute) to perform its work but on 10% of occasions this may take an additional hour. (Use a named normal distribution in combination with a probability profile distribution which has 60 minutes for 10% and 0 minutes for 90% of its results).
Time Absolute Distributions

A time absolute distribute lets you specify when you want something to happen (rather than a more usual approach of specifying how long it will be from now until the next thing happens).

This is particularly useful in situations where something happens at the same time (or about the same time) each day. For example, a mail drop may happen at around 8am and 2pm each day.

To use time absolute distributions just select NEW in any timing/distribution panel and then TIME ABSOLUTE DISTRIBUTION, then click the NEXT button.

The Time Absolute Distribution dialog lets you add a number of times the thing should happen each day.

For each time you also give another distribution name - this lets you say how much variability occurs around the time of day. For example if the morning mail drop is at 8am but the mail guy is up to 15 mins late specify 8am and a distribution which gives values from 0 to 15 (perhaps uniform). If the afternoon mail drop is normally at 2pm but can be early or late then you might want to specify 2pm plus a distribution which is "normal" with a mean of 0 and a standard deviation of 5 mins.

To add a new time of day click the ADD button and then type the time of day and select the required distribution.
Shift Patterns

A SIMUL8 shift pattern lets you say at what times of day parts of your model work. (If all of your model starts and stops at the same times then you do not need to use shift patterns - instead just set the clock to control the start time and length of each day).

To make use of shift patterns first set up a series of shifts (use Clock / Shifts).

Click NEW to add each shift (for example you might want two shifts called DAY SHIFT and NIGHT SHIFT).

Click DETAIL if you want to change the time each shift starts and ends each day. You can also say here which days of the week each shift applies to.

The shift details dialog box lets you change the name of the shift (which appears in all other dialog boxes of simulation objects which make use of shift patterns).

It also lets you change the times of day when this shift is "on". Enter the start time and then the duration. Use the 24 hour clock (e.g. 5pm is 17:00).

The behavior tab lets you say what should happen to work centers which are using resources when those resources are removed at the end of a shift.

At the end of a shift there may be a reduction in the number of resources available. If some or all of the resources which are removed are currently engaged in work at work centers then the behavior tab describes what happens.

Complete Tasks - behave as if the resource was still at the work center (or stays on late after the end of the shift) and complete the task in hand - then don't start another.
Suspend but restart ASAP - stop the work and hold the work center in suspension until another resource can be found (perhaps because one finishes work at another work center).

Suspend until next shift change - as above but don’t let any remaining resources be used on suspended work centers. Suspended work centers should not be considered for restart until the next shift change.

The behavior tab also lets you control what happens when shifts overlap. The default is that resources which are available in two shifts which overlap are available to the extent of the largest availability of the two shifts. You can change this assumption to summing together the two availabilities.
High Volume (BPR and FMCG) Applications

A High Volume Work Center (check the high volume box in the work center dialog) is particularly useful for Business Process Re-engineering (BPR) and Fast Moving Consumer Goods (FMCG) type applications. In these cases very high volumes of work may be processed at work centers and it may not be necessary to individually distinguish between every piece of work a work center does. For example, documents being processed by a work center may each be different but a “data entry” work center does not need to simulate them differently. This contrasts with lower volume applications where every piece of work may be very different (e.g. patients in a hospital where every patient has different attributes and is treated differently depending on these attributes. In high volume work centers, work also tends to be done in batches.

High volume work centers make use of the Quantity Attribute. Every work item type which goes through a high volume work center must have the quantity attribute attached to it.

High volume mode lets SIMUL8 process large numbers of transactions very quickly in batches while still retaining the ability to consider work individually (in terms of timing etc.). The timing information you place in a work center dialog box for a high volume work center is the time it will take to work on one unit of work in the batch. For example if the timing panel in the work center dialog says FIXED 10 and the Quantity attribute for a work item is set to 20 then it will take 200 minutes to process the work item. If the timing panel specifies a distribution then SIMUL8 will automatically adjust that distribution to one which is appropriate for the batch of work.

If you use batching out (from the routing out dialog) a high volume work center will split the quantity attribute across the newly generated work items. You can ask batching out to use the batch size as the new value for the quantity attribute. In which case it will automatically generate sufficient work items to use up all of the quantity in the work item which was in the work center. (More)
If you use the percent routing out rule at a high volume work center then the work item (batch) is automatically split into as many work items as there are routes leaving the work center and the quantity attribute is changed to be as close as possible to the percentages specified.

Remember you do not need to use high volume mode at every work center which processes work items which are actually batches of work.
Groups

A "group" is a collection of simulation objects.

You can use a group to limit the number of work items which can be in one area of your operation or to quickly create many copies of one of your objects.

Example of the use of a group to limit the work items in an area:

If you have a machine in your operation which performs two tasks in sequence on each work item (e.g. Setup and Weld on a welding machine) it may be best modeled as two separate work centers directly connected together with no buffer between them. However, without giving SIMUL8 an instruction, these two work centers could both contain a work item at the same time (the same machine being SETUP and WELDING at the same time!).

Simple solution: Create the two work centers and join the SETUP one to the WELD one with a routine arrow. Then select both work centers (drag a box around them) and click the right mouse button. Select "Create Group". Now go to the Group's dialog box (Objects / Groups in the main menu) and enter 1 in the "Limit Work Items in Group" box.

(Remember to also click on each work center in turn and enter the appropriate timing information for how long it takes to SETUP and WELD).

Example of use of groups to let you easily maintain copies of work centers.

If you have a model which contains five identical machines which work in parallel, each of which can perform a given task, then create just one machine and set up all the parameters as you require.
Then create a group with just this work center in it (drag a box around the work center and click the right mouse button). You can now use the Group Dialog box to make as many copies of this work center as you like.

You can also use the Group Dialog Box to delete all these extra work centers should you decide you need to change some parameters of the original work center and want to be able to recopy it.

Creating Groups

Groups are created by selecting the simulation objects and clicking the right mouse button. Groups are also created automatically when you use the Duplication Wizard.

The duplication wizard lets you do the equivalent of on-screen copy many times to create many copies of one or more objects (including their appropriate routing structure).

All the objects created are placed in a new group.

Access the duplication wizard by selecting a number of objects and the clicking the right mouse button.

A simulation object can only be in one group at any one time.

Add objects to an existing group by selecting at least one of the objects in a group and the objects you want to add to the group (then click the right mouse button).

To change details of a group use Objects / Groups in the main menu.
Results

A number of facilities enable results to be obtained from SIMUL8 models.

Most results from a simulation run are accessed via the dialog boxes for individual Work Centers (or other screen-based objects).

For example: click on the RESULTS button in the dialog box for a storage area to view the average waiting time of items which have been through that storage area.

Results Summary Window

This is one of the most important windows in your simulation model.

The results summary shows a collection of results information which you want to be able to conveniently see together (perhaps at the end of each simulation run you perform).

Access this via RESULTS / SUMMARY in the main menu.

The information in the results summary window is updated at the end of each simulation run (and also at the end of each simulation "trial").

Two different sets of figures are displayed. The most common are the results at the end of a run. Alternatively the results summary can contain the results of a trial.
The results at the end of a run

The results information displayed in the results summary window shows what happened during the run (either as an average, or as a limit value - for example when the result is something like a "maximum" size of a queue).

The result is the relevant value which was found over the length of the simulation run. For example, if the model runs for one week, and you look at a result like "average time in queue" then the result figure is the average of all the individual times which work items spent in the queue.

The results of a trial

The results at the end of a trial (see page 129) are displayed differently from the results at the end of a run.

The purpose of a trial is to check the reliability of results. At the end of just one run you have simulated one week in your organization. You have a result which is from one week. Just like real life, that could have been an average week, or a bad week, or a good week. We have no way of knowing what sort of week it was unless we run a few more of them.

A trial is a run of several weeks and the trial results summarize the results of these several weeks.

The central column of figures gives the result averaged across the weeks in the trial. This gives a guide as to what we expect the long term average to be (if we had time to do 1000's of runs).

The left and right columns give you some indication of how reliable the central (average) figure is. (In some cases the average figure may be very reliable because, perhaps, there is little variation from week to week, on the other hand, it may be less reliable, and you need to know if this is the case....for example an average of 50
when alternate weeks swing from 0 to 100 is very different from an average of 50 when they vary only between 49 and 51!!).

The left and right columns give a prediction of the range in which the average figure will be, if we had time to do 1000's of runs. Because simulation is a predictive method, which includes the variability which is in real life, it is never possible to be 100% accurate. These left and right figures give the range in which the long term average will be on 95% of times the prediction is made. (There is an option to change this to 99% in the Trials dialog).

The more runs you do in a trial, the more accurate the predictions will become.

**Adding results to the results summary**

To add a new entry in the results summary go to the individual results dialog which displays the information you want (for example a storage bin's results dialog) and click (with the right mouse button) on the piece of information you want to be displayed in the results summary. This will be added to the results summary immediately and the value on the results summary will be automatically updated from the dialog box whenever a simulation run (or trial) ends.

The mouse arrow contains a letter "R" when it is on top of a piece of information which can be added to the results window.

You can access the individual results dialogs by clicking on the objects on the simulation window, and then on the RESULTS buttons, or, if you have the results summary on the screen, click the DETAIL button to see a list of all objects for which results are available.

To remove a result entry from the results summary window either:

- Click right on it again (in the same way as you did to add it to the results summary).
or
Click on its subtitle in the results window (the second column) and click the delete option in the menu which appears.

Pie Charts

Pie Charts are available to show the amount of time work centers have been working, waiting, blocked or broken down. Click the button in the results dialog for the appropriate work center.

Histograms

Histograms are available showing the time work items have waited in queues and the time they have spent in the entire model (Click the button in the results dialog for the appropriate queue or work complete object).

Time graphs

Time graphs give a visual insight into the performance of part (or combinations of parts) of a SIMUL8 model.

A time graph of the number of items in a storage area can, for example, show you patterns in a queue size, or indicate when the model will never reach a steady state (if the queue size is increasing more-or-less continuously).

To see a time-graph, click on an object on the screen (such as a storage bin) and then click its RESULTS button, followed by its button.
TIME GRAPH button on the tool bar

The time graph button is used to create time graphs of selected objects (when the model is not running) and to update preexisting time graphs, histograms and pie charts while the model is running.

Combining Time Graphs

Time Graphs from a number of different simulation objects can be combined to enable you to see the total amount of work going in the whole (or some subset) of your simulated system.

Select a number of objects on the screen.

Then click the TIME GRAPH button. A time graph is created which shows a graph against time of the total work items in the selected objects.

Exporting Time Graph Results Data to Excel

Data in Time Graphs can be exported to Excel (or other WINDOWS packages).

In a Time Graph Window, click the system menu icon (in the top left of the window) once and choose "Copy Data". This copies all the data which makes up the time graph to the WINDOWS clip board.

In Excel, click on a cell and choose Edit / Paste from the menu. The data from the time graph will be pasted into your spreadsheet where you can perform detailed analysis.
Trials

A trial (or experiment) is a series of runs of the simulation model which are performed with the same settings of all the parameters in the models except for the "random numbers". As a simulation model is just like real life (it contains variability!) it is important to run a model more than once. (Was the run we did just a lucky week?). A trial gives you information about how variable the weeks are and how accurate you can expect your performance measures (results) to be.

For more information about the purpose of this see: Results of Trials

In the trials dialog you can adjust the number of runs in each trial.

The base random number stream can be changed if you want to make two trials, but use different random numbers for the two trials. (This is not normally necessary). The actual random number stream sets used for the first run in the trial is the base set, this is incremented for each run in the trial.

The name of the trial will be displayed in the results summary window.
Additional Graphics

Add additional static graphics to your model (to make it look more like the real situation you are modeling, by using the graphics palette. Click the button to switch on the graphics palette (or switch it off).

Use the graphics palette to add lines, boxes, circles, bitmaps (which can be created using any graphics editor such as the Windows Paint program), text etc.

To select a graphics object (to make changes to it after you first created it) just click on it (handles will appear).

To delete graphics object, select it, then press the delete key on the keyboard.

To change the shape of a graphics object select it, then drag the handles.
Diagnostic Facilities

These facilities are design to help you see what is happening inside your model as it runs. This helps you validate your model (see page 36).

Contents Button
The contents dialog lets you look inside a work center or storage bin at the actual work items currently inside it. Click "Contents" in the dialog box for the particular object.

The displayed dialog box lists the work items (the names shown in the left hand list are the names of the work item type) and their attributes. The work items in a queue are shown in sequence in the queue. The top of the list is the work item at the head of the queue (the next to be taken from the queue).

Clicking on a work item and an attribute name displays the values of the attribute.

You can also reach the contents dialog to see an individual work item. If you can see the work item on the screen (on a conveyor, in an animated queue, or just traveling between objects) just click on it while holding down CTRL and SHIFT on the keyboard.....this will take you straight to the contents dialog for that work item.

Simulation Monitor
The simulation monitor lets you see the detailed activity taking place within SIMUL8's internal simulation engine.

Access the monitor from the main menu (Clock / Simulation monitor).
The Next Step tab shows a description of the next thing which is going to happen or the next thing which SIMUL8 will test to see if it can happen. (For example "End of job on work center" or "Try to move a work item to a work center").

The History tab shows a list of the previous 100 "Steps" (as on the Next Step tab) except that the history only shows successful steps. (For example if "Try to move a work item to a work center" was not actually able to move the work item because the work center was already busy then this is not shown in the history as it did not actually happen). The history tab also shows the times at which each step took place.

Click RUN to run the simulation while continuously monitoring the steps. Click STEP to move forward just one step. Note that Step here is more detailed than Step on the main button bar because the main button bar step only shows steps which have an effect on the simulation model (the successful steps in the simulation monitor).
SIMUL8 Assistant

The SIMUL8 Assistant uses Expert System technology to help you get your model right. It watches the simulation model (and the way you are using it) to provide advice when it might be appropriate to do so.

How do you know the assistant has some advice. The SIMUL8 Assistant icon on the tool bar will light up to indicate if there is some advice for you.

If you do not want the assistant to watch your model you can switch it off (choose Wizards / SIMUL8 Assistant in the main menu and check the Hide All Advice check box).

Once you have read some advice you might decide it is not relevant to the way you use the model. In this case you can ask the assistant not to give you that advice again. You can switch off a particular category of advice completely, or you can switch it off just for the particular combination of simulation objects.

The RESET ALL button in the assistant dialog can be used to switch back on all categories of advice.
SIMUL8 Information Store

The Information Store holds data which you can use in your model. Access the store interactively using OBJECTS / INFORMATION STORE.

All information store items are “global” (anywhere in the model that they are referenced will find the same value in them). This differs from attributes (page 91) which have different values depending on the Work Item you are referencing. Hence they are all called “Global Data Items”.

**SIMUL8** provides some standard global data items in the information store:

Simulation Time - A number which is zero when the model has been reset and which (as the model runs) increases as the clock progresses. If you are working in minutes (Clock / Format) then, when the clock has run for 1 hour this global data item will contain 60.0. (If you where working in hours then it would contain 1.0)

Current Work Item - A unique identifier for the work item which is relevant inside a particular piece of Visual Logic (see page 145).

Results Collection Period - the amount of time for which result will be collected.

Warm Up Period - the amount of time for which the model will run before it starts to collect results.

Create new global data items by typing a name into the Name box (Objects / Information store menu) and clicking the Add button. Rename a global data item by clicking on its name in the list of names and changing the entry in the Name box.

You can change the value of a global data item at any stage interactively (just change the value in the Value box). You can also specify the value to be placed in the global data item whenever the model is reset.
Global data items can be single variables or “spreadsheets” of data. SIMUL8 spreadsheets are not designed to replace spreadsheet packages, they are storage places for data, or places you can create reports.

Create a spreadsheet by creating a global data item and changing its type to “spreadsheet”.

Click view to review / change the data in the spreadsheet. The COPY button will copy the entire contents of the spreadsheet to the clipboard in an EXCEL compatible format. The PASTE button will replace the contents of the spreadsheet with data on the clipboard (usually pasted there from a spreadsheet package).

Global data items are normally used from Visual Logic but you can also reference them directly from SIMUL8 timing boxes. For example, instead of setting the average time of a work center to 10 you could set it to MACHINE SPEED where machine speed is a global data item.
Visual Logic Extension User Guide

Visual Logic Introduction

Visual Logic lets you add detailed rules to your simulation model. It lets you tell the simulation exactly how to behave like the real operation you manage.

The difference between Visual Logic and “programming” is that in Visual Logic you are guided, step-by-step as you choose what logic to use - so you cannot get the syntax wrong.

This guide assumes you understand the most SIMUL8 concepts (but not the external extensions like the VB and EXCEL extensions).

The section Visual Logic Basics (page 144) explains how SIMUL8 obeys the Visual Logic you give it. (What the commands do to the work items and simulation objects etc.)

The best way to learn Visual Logic is by example. The section Building Visual Logic (page 148) will show you examples of Visual Logic in action (and refers you to models in the SIMUL8 Examples Directory). It also guides you through the Visual Logic editor dialogs.

The section “Visual Logic Commands by Category” gives a list by category of the facilities available in Visual Logic.

The section “Visual Logic Command Details” gives an alphabetical detailed list of all the Visual Logic facilities along with specifications of how each facility operates.
First we take a look at how Visual Logic fits into your simulation model and also give an overview of what it looks like.
Visual Logic Structure

This section deals with understanding where Visual Logic (VL) fits into your model. We also take a look at VL. We show you what it looks like. We show you how it is VL’s structure which makes it easy to use, easy to read and easy to write.

The structure of Visual Logic in your simulation model.

VL does not appear as one large stream of code in your simulation model. It is structured and divided into appropriate places that you simply access where you need to use them.
Here is an example of where you can place VL:

If you click this button you will go directly to a page where you can choose logic ("Visual Logic") to be obeyed by your simulation when this distribution is about to be sampled.

Once you have entered some VL here, clicking the button goes straight back to the same logic so you can easily see exactly what logic is used at a particular place.

Here are some more examples of where you can access VL:
When a work item completes work in a work center, the “On Work Complete” logic can be obeyed.

There are many places where you can access VL directly from the relevant place in your model:

- On Work Complete
- On Action (in Work Centers)
- On Action (in Work Entry Points)
- On Route In (before selecting a work item)
- On Route In (after selecting a work item)
- On Action (in Storage Bin initialization)
- On Reset
- On EndRun
- On Start Run
- On Stop Run (when the stop button clicked but not at the end of a run)
- On End Trial (at the end of multiple runs)
- On Time Check (at any time you decide)
You can also access any VL code directly from the Visual Logic Dialog (Choose Extensions / Visual Logic):

![Visual Logic Dialog](image)

The structure of Visual Logic Itself.

VL is different from most code you will have seen before.

Most VL is not typed. It is chosen from menus - but menus which are structured to make the choosing much faster than typing - and much easier than remembering what to type.

It is also hierarchical - so even large amounts of VL can be reviewed and understood quickly.
Here is a very small example. (This example is based on VLDEMO2.S8 in the examples directory):

The “+” next to “Fred Action Logic” means there is some VL inside the structured block called “Fred Action Logic”. This is the logic to be obeyed when Fred finishes work on an work item.

Click on the “+”

Here are two additional blocks. The contents of the first to be obeyed if the queue for Jack contains more than 5 work items.

You can open and close any block or sub block to make reading of VL convenient and organized.
In this VL the attribute MYROUTE is set to 1 or 2 depending on the size of the queue for Jack.

You may have as many levels in the hierarchy as you need and as many lines within any block as you need. There are no limitations.
Visual Logic Basics

What happens when SIMUL8 obeys some VL? The answer is similar to other scripting languages (like Visual Basic, for example) but with a few keys points which help you in the context of simulations.

VL Blocks

VL code is blocked in a hierarchy. Each line has a parent line and may have a number of siblings. A VL block is a group of siblings which all belong to one parent. You have already seen that you can hide all the children of a parent by clicking the “-” next to the parent line. The visibility of the block in the VL window has no affect of whether a block is obeyed. A block is obeyed whenever its parent is obeyed and the parent line tells the child block to be obeyed.

This sounds complicated but it is not. Example:

```
1  WORK CENTER 1 ROUTE-IN BEFORE LOGIC

2  SET Qpos = 0

3  WHILE Qpos < Wait Queue Count Contents

4  SET Qpos = Qpos+1

5  Select Current Work Item Wait Queue Qpos

6  IF Type Code = 4

7  Move Work Item To Queue for Work Center 1 -1

8  IF Queue for Work Center 1 Count Contents = 0

9  IF Wait Queue Count Contents => 0

10 Move Work Item To Queue for Work Center 1 -1
```

(in this example we have numbered the lines to make it easier to explain).
Lines 2, 3 and 8 are all siblings. They are one block. They have one parent (Line 1) and will all be obeyed if line 1 is obeyed.

Line 3 and 8 also have children of their own.

When Line 3 is obeyed it tells its children to be obeyed if “Qpos” is less than the number of work items in the queue called Wait Queue.

Because Line 3 is a “WHILE” it will repeatedly tell its children to be obeyed unless the condition fails. (Notice that one of the child lines (Line 4) increments Qpos - so eventually Qpos will exceed the limit tested in Line 3 and the children of line 3 will not long be obeyed. SIMUL8 then jumps to line 8.

Similarly Line 7 will only be obeyed if the condition in Line 6 is OK and so on.

**Current Work Item**

The “Current Work Item” is a very useful concept which lets you quickly do many things with one individual work item.

For example, during the VL code inside the “On Work Complete” button the “Current Work Item” is the work item which has just finished work. So, if you set an attribute you will be setting an attribute on the current work item.

Here’s an example:
The attribute called “Job Stage in Process” is incremented so that the work item knows how many times it has been through this work center. Every work item will have this attribute, but each of them may contain different values.

The “Current Work Item” is almost always the work item you are dealing with in your VL code. If a operation does not mention which work item to use then it is the “current work item” which will be used.

“Delete Work Item”, will, for example, remove the “Current Work Item” from the simulation object which currently holds it and completely remove it from the simulation.

In some sections of VL there is no current work item. For example “On Route In Before”. VL code inside this button is obeyed when an empty work center is about to select a work item to draw into the work center. This means that there is no work item in the work center and many potential work items might be selected. Hence there is no current work item at this stage.

You can select a current work item. Do this in some code which does not normally have a current work item (as just discussed), or when you want to deal with a different work item (different from the one which SIMUL8 made the current work item at the start of the section of VL code).

Example: In this VL we are looking for work which has an attribute “Type Code” set to 4. (When we find it we move it to a particular queue).
Above: The “WHILE” block checks every work item in the “Wait Queue” for a 4 in its Type Code attribute. Note that the “IF Type Code….” Line does not mention a particular work item. Instead the work item is the “Current Work Item” and this has been “selected” in the previous line.

Some facilities in VL automatically select a new current work item. For example “Add Work Item to Queue” creates a new work item and adds it to a queue. The created work item is then the current work item.

In those sections of VL which automatically provide a current work item, the Current Work Item supplied by SIMUL8 is only accessible until you change it (using one of the facilities which selects a different work item as current).
Building Visual Logic

This section explains how to build VL code. It gives detailed instructions about how to input and edit VL.

We are going to build a simulation in which a standard facility is supplemented by a back up facility at busy times.

Create a simple model like this:

![Diagram of a simple model](image)

and change the processing time at Work Center 2 to 11 minutes.

Work Center 3 is very expensive so we want to make sure it is highly utilized so we are going to supplement work center 2 with a additional help but we will only use this when it would be useful to do so.

Copy work center 2 and its storage bin (drag a box around the bin and the work center, then drag the box while holding down the control key on the keyboard).
We are going to send work to work center 4 from work center 1 when there is not much work waiting at the important work center (3) but only if work center 2 is getting a behind with its work (there is a queue of work at it).

Specifically, if there are less than 3 items waiting for work center 3, and there is a queue at work center 2, then we will send work to work center 4 instead of 2.

Go to Objects / Work Item Types and enter the detail dialog for work item type 1.

Add an attribute to these work items (and call it “Route to take”) - we are going to use “Attribute Routing” out of work center 1.

In the above dialog click ADD:

Then click NEW to create a new attribute and type in its name:
Click OK several times to get back to the main screen.

Click on work center 1 and change its routing out to “Attribute” and set the required attribute to “Route to take” in the DETAIL button.

Now we have to add the VL code to set the attribute to either: 1 (go to the first destination in the above list) or 2 (go to the second destination in the above list).

Click the On Work Complete button.
Now we have a blank window into which we can enter VL. This VL will be obeyed whenever a work item finishes its work at work center 1.

As the window says, click the RIGHT mouse button (in the window) to start adding logic.

The first thing to do is set the attribute to the default value (1 = go to Queue for Work Center 2).

Choose SET… from the menu when you click RIGHT.
The SET VALUE dialog appears. You can set attributes, global data (variables) and other changeable data here.

Click the left hand side box.

The formulae editor appears:

(make sure the OBJECT check box is checked)
We want to set ROUTE TO TAKE - so double click it and then click OK.
Now click the right hand side of the SET dialog.

We are now back in the formula editor but this time we want a value:

Just type a 1 into the box and click OK.

Click OK in the SET DIALOG too.
If you run the model with just this VL code it will behave as it did without Work center 4 because the attribute will always be 1 (= go to destination 1).

Click the right mouse button again (in the VL code window) and add an IF line. The Conditional Block Editor will appear.

(The Conditional Block Editor is also used to While and UNTIL blocks).

When you click either of the formula boxes you will see the formula editor again.

You want a line which looks like this:

\[
\text{IF Queue for Work Center 3.Count Contents} < 3
\]

so, click the top box and use the formula editor to select OBJECT. Then double click QUEUE FOR WORK CENTER 3 and then COUNT CONTENTS. You will see that the formula editor knows to insert a dot for you between the name of the object and the name of its property.

Use the formula editor to put a 3 in the bottom box and choose IS LESS THAN for the type of condition.
All VL lines which can have child lines automatically insert a dummy line when they have no children.

If you click RIGHT when a line is highlighted new lines are inserted at the same level in the hierarchy.

If you wanted to add a line at the same level as the SET and the IF lines (to be obeyed after the IF block) you would highlight (click on) the IF line first, then RIGHT click.

Add another IF inside the block for the first IF and then add another SET so you have some logic like this:

Running the model like this will increase the utilization of Work Center 3.
So far we have only looked at SET and IF lines.

Loop lines are important because they let you perform operations a specific number of multiple times.

Loops require a “Global Data Item” to be used to count to number of times the loop is performed. (Create a Global Data Item in Objects / Information Store - make sure the item you create can contain NUMBERS).

This loop is used to add 5 work items to the queue for Work Center 5 when the model is reset.

What type of line is “Add Work Item to Queue”?

This is a “Command”. Commands are not part of the structure of VL. They are specific to the requirements of SIMUL8. There are many commands - explore the menus and see the section of this manual which lists them (page 161).
You can also explore the commands available from within the command editor. The command editor lets you easily enter parameters for each command.

As you change the command or click on parameters the text automatically changes to prompt you for the type of information required.
Visual Logic Commands By Category

The next section gives all the VL commands and the details of their operation but lists them alphabetically. The current section categorizes the commands by type so it is easier to find the command you need for a particular purpose.

Model related commands

- **Reset**: Resets the model to time zero
- **Run**: Runs the model
- **Stop**: Stops the model
- **Speed**: Changes the speed of the model
- **Save**: Saves the model to disk
- **VBA Signal**: Sends a signal to Visual Basic (or EXCEL)
- **Beep**: Sounds the PC bell
- **Delay**: Stops for x seconds
- **Display Message**: Stops and displays screen message
- **Break**: Jumps out of an UNTIL, WHILE or LOOP

Object Related Commands

*Work Items*

- **Select**: Selects a work item as the “current work item”
- **Select on Conveyor**: Selects a work item by distance from front of conveyor.
- **Image**: Changes the image used for the current work item
- **Add to Queue**: Creates a new work item and adds it to a queue
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Deletes the current work item from the model</td>
</tr>
<tr>
<td>Move</td>
<td>Moves the current work item to a new queue</td>
</tr>
</tbody>
</table>

**Simulation Object**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break Down</td>
<td>Stops a work center</td>
</tr>
<tr>
<td>Restart</td>
<td>Restarts a work center after Break Down</td>
</tr>
<tr>
<td>Empty</td>
<td>Deletes all work items in an object</td>
</tr>
<tr>
<td>Image</td>
<td>Changes the image for an object</td>
</tr>
<tr>
<td>Resources Required</td>
<td>Changes the number of a resource required at a work center</td>
</tr>
<tr>
<td>Set Collect</td>
<td>Changes the number of work items collected from a source during collect routing in</td>
</tr>
<tr>
<td>Set Route Out Discipline</td>
<td>Changes the method of routing out</td>
</tr>
<tr>
<td>Set Route Out Percent</td>
<td>For percentage routing out changes the percentage going to a particular destination</td>
</tr>
<tr>
<td>Percent Route Out Adjust to 100</td>
<td>Automatically pro-rata adjusts routing out %s so they add to 100</td>
</tr>
</tbody>
</table>

**Distribution**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability Profile Column</td>
<td>Changes individual cells in user defined distributions</td>
</tr>
<tr>
<td>Named Distribution Parameters</td>
<td>Changes type of distribution (and shape)</td>
</tr>
</tbody>
</table>
Visual Logic Command Details

This section lists alphabetically all the VL commands and explains their operation.

This information is also in the SIMUL8 online help.
Add Work Item to Queue command

Definition:

Add Work To Queue
Work Item Type
Storage Bin

Description:
Creates a work item and adds it to the storage bin. The added work item becomes the Current Work Item.

Beep

Definition:
Beep

Description:
Sounds the PC bell

Comment:
Useful for checking your logic - if the bell sounds then you know that piece of Visual Logic is being obeyed.

Break

Definition:
Break

Description:
Jumps out of a LOOP, UNTIL or WHILE and begins execution at the line after the loop.

Comment:
Useful when a loop is used to find a work item in a queue - break when you have found the item required.
**BreakDown**

**Definition:**
Break Down
Simulation Object
(Conveyor, Work Center, Work Entry Point)
Duration

**Description:**
Stops a work center (etc) for Duration time units. If Duration = 0 then Work Center will not automatically re-start.

**Comment:**
Use with Work Entry Points to stop arrivals

**See also:**
Break Restart

**BreakRestart**

**Definition:**
Break Restart
Simulation Object
(Conveyor, Work Center, Work Entry Point)

**Description:**
Restarts a stopped (broken down) a work center (etc).

**See also:**
Break Down

**Delay for Seconds**
Definition:
  Delay for Seconds
       Number of Seconds

Description:
  Stops the model for specified number of seconds. Model continues after the
time has elapsed. The main tool bar shows “Pausing” during the time.
Clicking the word “Pausing” will continue the model immediately.

Comment:
  Useful when model will be used to demonstrate a particular issue which
requires the simulation to slow down or stop for a time at key times.

Delete Work Item
Definition:
  Delete Work Item

Description:
  The Current Work Item is removed from its storage bin and deleted from the
model

See also:
  Move Work Item To
  Empty Simulation Object

Display Message
Definition:
  Display Message
       Message Text

Description:
Displays message text on screen with OK button. Model stops until OK clicked

Comment:
Useful for checking your logic or displaying special information to user.

**Empty Simulation Object**

**Definition:**

Empty Simulation Object  
Simulation Object  
(Storage Bin, Work Center, Conveyor)

**Description:**
Removes all work items from the object and deletes them from the model

**See also:**
  
  Move Work Item To  
  Delete Work Item

**Get from EXCEL**

**Definition:**

Get from EXCEL  
Global Data Item (Number, Text, or Spreadsheet location)  
EXCEL Sheet name in format "[?????.XLS]Sheet1"  
Column in Excel  
Row in Excel  
Width in Excel (1 for single cell)  
Height in Excel (1 for single cell)
Description:
Copies information from EXCEL sheet into a SIMUL8 Global data item. If the Global data item is a spreadsheet location then Width and Height can be used to copy an area of a spreadsheet from EXCEL to SIMUL8.

Comment:
Use at the start of a run (during RESTART logic) to copy in relevant information for this run.
If the EXCEL Sheet name braces contain no file name (eg []Sheet1) then the file name from the Excel Extension dialog is automatically used.

See also:
Set in Excel

Get Result
Definition:
Get Result
Global Data Item (Number)
Result Type (Current Run, Average, Confidence Limit etc)
Result (Choose from any on the results summary)

Description:
Copies a result summary value to a Global Data Item.

Comment:
Really useful for compiling results reports or basing your next run on the result of a previous run.

See also:
Set in Excel
If...ElseIF....Else
Definition:
    IF
    Condition
    Visual Logic Sub-Block

Description:
Tests the condition and if this is successful obeys the lines of Visual Logic in the sub-block.

Definition:
    ELSEIF
    Condition
    Visual Logic Sub-Block

Description:
If and only if none of the preceding IF / ELSEIF lines in a continuous sequence at this indent level have been successful ELSEIF tests the condition and if this is successful obeys the lines of Visual Logic in the sub-block.

Definition:
    ELSE
    Visual Logic Sub-Block

Description:
If and only if none of the preceding IF / ELSEIF lines in a continuous sequence at this indent level have been successful obeys the lines of Visual Logic in the sub-block without testing any further conditions.
Loop
Definition:
   LOOP
       Starting value of global data item
       Name of global data item used to count the loop
       Ending value of global data item
       Visual Logic Sub-Block

Description:
   (1) The global data item is set to the starting value.
   (2) If the global data item if <= to the Ending value then Visual Logic Sub Block is obeyed otherwise stop the loop.
   (3) The global data item is incremented by one
   (4) Go back to (2) and repeat

Move Work Item To
Definition:
   Move Work Item To
       Storage Bin

Description:
   The Current Work Item is removed from its storage bin moved to a new storage bin.

See also:
   Delete Work Item
   Empty Simulation Object
**Percent Route Out Adjust to 100**

**Definition:**
Percent Route Out Adjust to 100
Leaving Work Center

**Description:**
Pro-rata adjusts all percent route out from the work center so that they add to 100%.

**Comment:**
This facility is rarely required. You should normally use Set Route Out Percent to correctly see all percentages.

**See also:**
Set Route Out Percent

**Reset Clock**

**Definition:**
Reset Clock
Random Number Stream Set

**Description:**
The clock is set to zero, the random number generators reset using the new stream set number and all simulation objects initialized. If the Random Number Stream Set is zero then the stream set is not changed.

**RunModel**

**Definition:**
Run Model
   End Run Time

Description:
   Runs the model until End Run Time. This sets Results Collection Period =
   End Run Time minus Warm Up Period

Comment:
   Useful for checking your logic - if the bell sounds then you know that piece
   of Visual Logic is being obeyed.

Save Model
Definition:
   Save Model
      Filename Text string
         (Valid file name or blank)

Description:
   Saves the model to disk as File Name. If File Name is blank then the current
   filename of the model is used.

Select Current Work Item
Definition:
   Select Current Work Item
      Simulation Object
         (Storage Bin, Work Center, Conveyor)
      Position in Object
         (0=tail, 1=head, 2=second in queue etc)

Description:
Selects a work items as the current work item. If there is not a work item at
the selected location the current work item becomes zero.

See also:
Move Work Item To
Delete Work Item
SET (an attribute)

Select Work Item on Conveyor

Definition:
Select Work Item on Conveyor
    Conveyor
    Start of Search Region
        (metres from front of conveyor)
    Length of Search Region
        (metres from front of conveyor)

Description:
Selects a work items as the current work item if it is the first work item
found when searching a part of a conveyor starting at "start" distance from
the front of the conveyor and ending at start+length. If there is not a work
item at the region the current work item becomes zero.

Comment:
The "front" of a conveyor is the point where the work items travel to while
on the conveyor. To select a work item which is ready to be removed from
the conveyor because it has traveled the full length of the conveyor set the
start and length parameters to zero.

See also:
Move Work Item To
Delete Work Item
SET (an attribute)
SET command
Definition:
SET
   A single data item
       (Attribute, Global Data Item, Simulation Object parameter
        etc)
   =
   A value
       (An expression)

Description:
Sets the single data item such as a global data item, or a particular attribute
on the current work item or a property of a simulation objects (like the max
size of a storage bin) to the value on the right hand side of the ".=".

Set Collect Number
Definition:
Set Collect Number
   Input Simulation Object
       (Where the work items come from)
   Collect Number
       (Number of items to be collected)
   Work Center
       (Where the work items are collected into)

Description:
For work centers which use COLLECT routing in, sets the number of work
items to be collected from a particular route in.

Comment:
The "Input Simulation Object" must already be in the list of objects "routing in" to the work center.

**Set in EXCEL**

**Definition:**
- Set in EXCEL
  - Global Data Item (Number, Text, or Spreadsheet location)
  - EXCEL Sheet name in format "[?????.XLS]Sheet1"
  - Column in Excel
  - Row in Excel
  - Width in Excel (1 for single cell)
  - Height in Excel (1 for single cell)

**Description:**
Covers information to EXCEL sheet from a SIMUL8 Global data item. If the Global data item is a spreadsheet location then Width and Height can be used to copy an area of a spreadsheet from SIMUL8 to EXCEL.

**Comment:**
Use at the end of a run (during ENDRUN logic) to copy relevant results information to EXCEL.
If the EXCEL Sheet name braces contain no file name (eg []Sheet1) then the file name from the Excel Extension dialog is automatically used.

**See also:**
- Get from Excel

**Set Route Out Percent**

**Definition:**
- Set Route Out Percent
  - Leaving Work Center
  - Percent
  - Destination Object
Description:
When Routing Out Discipline Percent is used this changes the percentage of work items which are sent to a particular destination.

Comment:
Normally you will set several Route-Out-Percents in sequence because the complete list of route out percents should add to 100%. If you do not change several so that the total in 100% SIMUL8 will not correct this error for you (until you interactive use the Routing Out Dialog). If a work center leaves a work center which has a set of percents which do not add to 100% any surplus or deficit affects the percentage of work items going to the final objects in the destination list.

If you want SIMUL8 to automatically correct the total to 100% by pro-rata adjustment of all percentages then use Percent Route Out Adjust to 100.

See also:
Percent Route Out Adjust to 100

Set Route Out Discipline
Definition:
Set Route Out Discipline
Leaving Work Center
(Work center where routing out applies)
Routing out method
(eg "Circulate" - choose from list)

Description:
Changes the routing out method.
Set Distribution Parameters

Definition:

Set Distribution Parameters

- Distribution Name
  (Named Distribution)

- Distribution Type
  (Eg "Normal", "Fixed" etc - choose from list)

- Parameter 1
  (Purpose depends on Distribution Type)

- Parameter 2
  (Purpose depends on Distribution Type - not used in some cases)

- Parameter 3
  (Purpose depends on Distribution Type - not used in some cases)

- Parameter 4
  (Purpose depends on Distribution Type - not used in some cases)

Description:

Changes a named distribution.

Set Resources Required

Definition:

Set Resources Required

- Work Center
- Resource
- Minimum Number Required
- Maximum Useable Number
Description:
Changes the quantity of resources required at a work center.

Comment:
In most cases the Maximum and Minimum parameters will be the same (For exceptions see Resource Details Dialog). The Resource must already be connected to the work center. This command does not make new connections. 0 is a valid numbers (so that none of a particular resource are required sometime).

See also:
Clear All Resources Required

**Set Object Image**

Definition:
Set Object Image
Simulation Object
Image Name

Description:
Changes the image used for the object

**Set Prob-Profile Distrib Column**

Definition:
Set Prob-Profile Distrib Column
Distribution Name
(Probability Profile Distribution)
Column Number
(1= left most column in dialog)
Sample Value
Percentage Chance
Description:
Sets the sample value and percentage values for a single column (cell) in a
probability profile distribution. Percentage chance values for columns to the
right of the "Column Number" column are adjusted so that the total
percentage = 100.

Comment:
As columns to the right are adjusted to ensure total percentage = 100 you
are recommended to only set columns in numerical (left to right) sequence.

**Set Work Item Image**

Definition:
Set Work Item Image
   Image Name

Description:
Changes the image used for the Current Work Item

**Signal VBA**

Definition:
Signal VBA
   Text String

Description:
Sends a text message to EXCEL/VB.

Comment:
Although this is mostly useful for back compatibility with past SIMUL8
models which used EXCEL/VB to control logic, it also lets you create more
signals to EXCEL/VB to better control you model from EXCEL/VB should you choose this as your preferred extension language.

**Simulation Speed**

**Definition:**

Simulation Speed  
Speed %  
(0-100)

**Description:**  
Changes the current running speed (equivalent to moving the speed control slider on the main window). 0 stops the model running. 100 switches off all graphics updates to maximize speed.

**StopModel**

**Definition:**

StopModel

**Description:**  
The model stops running.

**Until**

**Definition:**

Until  
Condition  
Visual Logic Sub-Block

**Description:**  
Obeys the lines of Visual Logic in the sub-block and then tests the condition and if this is UNsuccessful then repeats. Continues like this until the test is successful.
Comment:
Significantly different from WHILE in that the sub-block is always obeyed at least once.

**While**

Definition:
While

Condition
Visual Logic Sub-Block

Description:
Tests the condition and if this is successful obeys the lines of Visual Logic in the sub-block. Then repeats. Continues like this until the test is unsuccessful.

Comment:
Significantly different from UNTIL in that if the condition is unsuccessful on the first attempt the sub-block is not obeyed even once.
SIMUL8 Extensions

The purpose of the SIMUL8 Extension facility is to allow you to extend SIMUL8 in any way you need and interface it to other standard software packages.

SIMUL8 Visual Logic lets you add any control rules you need. However, you may prefer to work in an industry standard language or you may wish to use Windows API facilities which can be accessed directly from extension packages.

The SIMUL8 extensions work by letting you write logic in EXCEL’s macro language (Visual Basic Applications edition).

As a by product the extensions facility also lets you do a few other useful things:

- Store many of your model variables in Excel (or VB)
- Present results or variables in a very recognizable spreadsheet form
- Build completely tailored front ends or back ends to your model (dialog boxes and whole screens designed to present information in whatever way you like).
- Even completely hide SIMUL8 so that a system you build can contain a simulation without an end user needing to realize SIMUL8 is there!

Note that, although this part of the manual is written mainly discussing EXCEL’s VBA (Visual Basic Applications edition) what appears here also works with the full Visual Basic (either Standard or Professional edition). Some examples are provided. The use of Visual Basic (Standard or Professional) is necessary to build complete systems which hide within them SIMUL8. The interface to the extensions can also work with other windows languages which can use standard windows application interfacing techniques. Contact your dealer if you want information on how to do this.
The SIMUL8 / Excel Extension

This section describes how the SIMUL8 Excel Extension works, explains how to get started using it for the first time and then details all the facilities available.

The theory in the past has always been that any packages which is easy to use will also be restricted in what it can do for you. The idea of the SIMUL8 Excel extension is to overcome this because Excel’s macro language - Visual Basic Applications Edition is a full windows programming language - and so presents no real restrictions on what can be done. VBA is, of course, more difficult to use than simply drawing your SIMUL8 model on the screen. This is why you would not use VBA until you really need to.

Remember all that is written here also applies to Visual Basic Standard and Professional Editions.

From here on we will talk about VB. This means Excel and VBA, VB Standard or VB Professional.

There is no clear boundary where VB must be used. There can, for example, be things you might want to simulate which are not straightforward in SIMUL8, but which you can, nevertheless, describe to SIMUL8 (perhaps making use of “synthetic work centers” - a work center which does some routing but which takes no time and is set to “invisible” on the screen). These situations might be occasions where some people might prefer to use VB.

How It Works

The Excel extension works in a number of ways depending on what you want to do.
“SIGNALS” are sent to your VB code when SIMUL8 tries to do certain things to your model (you tell it to do this). For example, when SIMUL8 samples a distribution, it can be told to signal your VB code that it is about to do this.

When a “Signal” is received by your VB code a particular procedure in your code is told what the name is of the signal. (They all have names, and you can create your own named signals). At this point your VB code can inquire of SIMUL8 about the state of the model and can change the model so that SIMUL8 then knows what to do next with the model.

How to get started with the Excel Extension

A number of example models are provided, but one easy way to understand the Excel extension is to build a small extension to one of your models.

If you want to look at the examples work through the VBADEMO1, 2 and 3 SIMUL8 models to be found in your SIMUL8 examples directory. If you are using VB Standard or Professional look at VBDEMO1, 2 and 3 instead.

To use the SIMUL8 extensions you need to have Excel 5 or newer or Visual Basic Standard (or Professional) edition (version 3 upwards) on your computer.

- If you are using Excel choose EXTENSIONS / EXCEL in the main menu and click AUTO-CREATE.
- If you are using Visual Basic Standard (or Professional) choose EXTENSIONS / VISUAL BASIC in the main menu, type a project name into the project name box (for example, MYVBEXT) and click AUTO-CREATE.

This will load the appropriate software and create a shell set of macros ready to run with SIMUL8. It contains sample code so you can see how it works.
It also loads help screens which will take you step-by-step through using SIMUL8 with the software you have chosen. The first time you use this we suggest you follow these help screens carefully.

In normal use you will delete the sample code but the first time you see it you should spend a short while seeing how it works.

Basically, the code says “if you are sampling the distribution called Distrib1 then change its mean value to the value which appears on spreadsheet 1 in cell B1 (Row 1 Column 2)”. (The Visual Basic equivalent is “the value which appears in text box 1”)

To use this piece of code change one of the work centers in your model to use a new distribution called Distrib1 and check the “ASK VBA” box in that distribution’s dialog box.

- If you are using EXCEL you can simply click a tab which displays a spreadsheet (you can’t look at macros while they are running),
- if you are using Visual Basic you will need to run the program (press F5 in Visual Basic),

Now go back to SIMUL8 (use ALT-TAB), restart the clock and set the model running.

The work center will take (on average) the amount of time in the box on the screen to process each work item.

The following two sections detail the SIGNALS provided by SIMUL8 and the “functions” and “subs” (VB terms) which can be used to inquire about and change your model.
Functions which interrogate and control your model

EXCEL macros can ask **SIMUL8** what is going on at the current time in the model and can adjust parameters and direct control in the model.

The following SUBs and FUNCTIONs are supplied in the SIMUL8 Library tab in Excel extensions to **SIMUL8**.

(Note: If you are using Visual Basic (not VBA) this is called S8DEC.BAS on your disk and is automatically added to your Visual Basic project by AUTO-CREATE. If you do not use Auto-Create and want to make use of these subs and function then use FILE / ADD FILE in Visual Basic and select S8DEC.BAS in your **SIMUL8** directory to make the following SUBs and FUNCTIONs available to your program).

Overview: (for details see following pages):

- **AddWItoQueue**: Creates a new Work Item and adds it to the queue.
- **AdjustRoutePercent**: See **SetRoutePercent**.
- **AttribValue**: What are the values of the attributes on the "Current" Work Item?
- **AveSize**: Returns average contents of an object.
- **AveTimeIn**: Returns the average time work items have spent in a storage bin.
- **AveTimeInByCategory**: As AveTimeIn but can retrieve information by segregation category.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BreakDown</td>
<td>Stop a Work Center and suspend its activities (either for a specific time, or until you call BreakRestart).</td>
</tr>
<tr>
<td>BreakRestart</td>
<td>See above.</td>
</tr>
<tr>
<td>ClearAllResourcesRequired</td>
<td>See SetResourcesRequired.</td>
</tr>
<tr>
<td>CollectComplete</td>
<td>Tell a work center (which collects a number of items before starting to process them) that it should stop waiting for more items and process the current set of work.</td>
</tr>
<tr>
<td>ConnectSIMUL8</td>
<td>Connects your Excel macros to SIMUL8.</td>
</tr>
<tr>
<td>EmptyQueue</td>
<td>Empty a queue.</td>
</tr>
<tr>
<td>ExitSIMUL8</td>
<td>Close SIMUL8.</td>
</tr>
<tr>
<td>Get_WI_Held_Resource_Count</td>
<td>Find out how many of a resource a work item is using.</td>
</tr>
<tr>
<td>MinAveMax</td>
<td>Get results information from a storage bin (queue).</td>
</tr>
<tr>
<td>ModelSpeed</td>
<td>Adjust the position of the speed scroller.</td>
</tr>
<tr>
<td>QueueSize</td>
<td>What is the current size of a queue (storage bin)? Also works for Work Centers and Work Exit Locations.</td>
</tr>
<tr>
<td>ReSetModel</td>
<td>Set the clock back to the start time.</td>
</tr>
<tr>
<td>RunModel</td>
<td>Set the model running until a particular time.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SaveModel</strong></td>
<td>Save the model.</td>
</tr>
<tr>
<td><strong>SelectWI_in_Object</strong></td>
<td>Change the current work item to be the work item at a position in an object.</td>
</tr>
<tr>
<td><strong>SetAttribValue</strong></td>
<td>Change the values of the attributes on the &quot;Current&quot; Work Item.</td>
</tr>
<tr>
<td><strong>SetAttributeBatchSize</strong></td>
<td>Change a work center's routing-in attribute batch size parameters.</td>
</tr>
<tr>
<td><strong>SetCollect</strong></td>
<td>Change the number of items required by a work center using &quot;collect&quot; Routing-In.</td>
</tr>
<tr>
<td><strong>SetConveyor</strong></td>
<td>Change parameters of a conveyor.</td>
</tr>
<tr>
<td><strong>SetDistribution</strong></td>
<td>Change the parameters of one of the model's named distributions.</td>
</tr>
<tr>
<td><strong>SetMaxLevel</strong></td>
<td>Set the maximum size of a queue (or similar - e.g. max number of a resource available).</td>
</tr>
<tr>
<td><strong>SetPriority</strong></td>
<td>Change a work center's priority in looking for resources.</td>
</tr>
<tr>
<td><strong>SetResourcesRequired</strong></td>
<td>Change the number of a particular resource which is required per task at a work center.</td>
</tr>
<tr>
<td><strong>SetRouteMode</strong></td>
<td>Set routing out discipline for a work center.</td>
</tr>
<tr>
<td><strong>SetRoutePercent</strong></td>
<td>Change Routing Out Percentages.</td>
</tr>
<tr>
<td><strong>SetWarmUpPeriod</strong></td>
<td>Set the warm up period.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetWITypeLength</td>
<td>Change the size of a work item type.</td>
</tr>
<tr>
<td>SimTime</td>
<td>What is the current simulation time?</td>
</tr>
<tr>
<td>StepModel</td>
<td>Run the model for only one internal simulation event.</td>
</tr>
<tr>
<td>StopModel</td>
<td>Stop the model (stop the clock and release the RUN button).</td>
</tr>
<tr>
<td>StopTask</td>
<td>Cause a currently active task at a work center to end immediately and the associated Work Item to proceed to the next object in the model.</td>
</tr>
<tr>
<td>Utilization</td>
<td>Return the percentage utilization of the named object.</td>
</tr>
</tbody>
</table>
Detailed Description and Examples of Excel Interface Library

The following sections list in alphabetical sequence the routines in the SIMUL8 library. Each definition is followed by an example of its use. The examples are also in the help file and can be used by copying and pasting them from the help file.

AddWltoQueue

VBA Declaration:

Sub AddWltoQueue (witypename As String, qname As String)

Creates a new work item (of type witypename) and adds it to the back of queue qname. The Work Item created becomes the "Current Work Item"

AddWltoQueue Example

If Signal = "RESET:" Then
  For n = 1 To 20
    AddWltoQueue "Work Item Type 1", "Storage Area 1"
    SetAttribValue "Attribute 1", n
  Next n
End If

AdjustRoutePercent

VBA Declaration:

Sub AdjustRoutePercent (ByVal fromname$)

Pro-rata adjust all route-out percentages after a series of call to SetRoutePercent
AdjustRoutePercent Example

If Signal = "TIMECHECK" Then
    Day = SimTime() Mod (8*60)
    If Day = 1 Then
        SetRoutePercent "Work Center 1", 10, "Storage Area 1"
        SetRoutePercent "Work Center 1", 10, "Storage Area 2"
        SetRoutePercent "Work Center 1", 80, "Storage Area 3"
        AdjustRoutePercent "Work Center 1"
    Else
        SetRoutePercent "Work Center 1", 80, "Storage Area 1"
        SetRoutePercent "Work Center 1", 20, "Storage Area 2"
        SetRoutePercent "Work Center 1", 0, "Storage Area 3"
        AdjustRoutePercent "Work Center 1"
    End If
End If

AttribValue

VBA Declaration:

Function AttribValue (attname As String) As Single

Returns the value of the attribute named in attname on the current work item.

AttribValue Example

If Signal = "SET: Attribute   1" Then
    SetAttribValue "Attribute   1", AttribValue("Attribute   1")*10
End If

AveSize

VBA Declaration:

Function AveSize(objectname As String) As Single

Returns the average number of work items which have been in the storage bin over time.
**AveSize Example**  
At the end of the run place the average size of a queue in cell E1.

If Signal = "ENDRUN:" Then  
    Worksheets("Sheet1").Cells(5, 1).value = AveSize("Storage Bin 1")  
End If

**AveTimeIn**

VBA Declaration:

```vba
Function AveTimeIn(objectname As String) As Single

For a storage bin returns the average time work items have spent in the storage bin (for all work items which have left the bin/queue). If applied to a Work Exit Point it returns the average time the work items which have reached this work exit point have spent in the entire simulation model (i.e. from the point they were created. This is usually at a Work Entry Point, but work items can also be created at work centers, for example because batching out rules create multiple work items).

AveTimeIn Example
If Signal = "ENDRUN:" Then  
    Worksheets("Sheet1").Cells(5, 1).value = AveTimeIn("Storage Bin 1")  
End If

**AveTimeInByCategory**

VBA Declaration:

```vba
Function AveTimeInByCategory(objectname As String, Cat As Long) As Single

As AveTimeIn but can retrieve information by segregation category. Set Cat to the desired value of the segregation category. Cat = Zero returns value for all categories (exactly the same value as AveTimeIn).```
AveTimeInByCategory Example

If Signal = "ENDRUN:" Then
    Worksheets("Sheet1").Cells(5, 3).value = AveTimeInByCategory("Storage Bin 1", 3)
End If

BreakDown

VBA Declaration:

Sub BreakDown (Wname As String, tt As Single)

    Stops a Work Center for tt time units. If tt <= 0 then the Work Center will not restart until you call BreakRestart.
    
    For this utility to work the work center will have to have SIMUL8's own breakdown facility enabled. Enable this by creating any breakdown (with a repair time of zero if you do not really want any automatic breakdowns).
    
    If you call breakdown while the work center is already broken down (either by VB or by SIMUL8's own break down / efficiency options) then that breakdown will be canceled and the new one substituted.

BreakDown Example

If Signal = "TIMECHECK" Then
    If (SimTime() Mod (8*60)) = (2*60) Then 'Note: It is 10am
        'BreakDown "Storage Bin 1", 3
    End If
End If

BreakRestart

VBA Declaration:

Sub BreakRestart (Wname As String)
End a breakdown on Work Center Wname

**BreakRestart Example**

If Signal = "TIMECHECK" Then
   If QueueSize("Storage Area 1") < 5 Then
      BreakDown "Work Center 1", 0
   ElseIf QueueSize("Storage Area 1") > 10 Then
      BreakRestart "Work Center 1"
   End If
   End If

**ClearAllResourcesRequired**

VBA Declaration:

Sub ClearAllResourcesRequired (ByVal wcname$)

Sets all resource requirements at a work center to zero (i.e. no resources required). Normally used prior to a series of calls to SetResourcesRequired

**ClearAllResourcesRequired / SetResourcesRequired Example**

If Signal = "ROUTE_IN BEFORE: Work Center 1" Then
   SelectWI_in_Object "Queue for Work Center 1", 1
   If AttribValue("Attribute 1") = 1 Then
      ClearAllResourcesRequired "Work Center 1"
      SetResourcesRequired "Resource 1", 2, "Work Center 1"
      SetResourcesRequired "Resource 3", 1, "Work Center 1"
   Else
      ClearAllResourcesRequired "Work Center 1"
      SetResourcesRequired "Resource 1", 1, "Work Center 1"
      SetResourcesRequired "Resource 2", 1, "Work Center 1"
   End If
   End If
CollectComplete
VBA Declaration:
Sub CollectComplete (Wname As String)
The named work center will not wait for any more products. (This only applies to Work Centers using "Collect" routing-in.)

ConnectSIMUL8
VBA Declaration:
Sub ConnectSIMUL8()
Connects your Excel macros or Visual Basic code to SIMUL8.
Always use this when your model is loaded to ensure that Excel knows which SIMUL8 model it is operating on. (If you get a message like "Mistake in use of SIMUL8 Library - Work Center 1 does not exist" - when Work Center 1 does exist, this is usually because you have forgotten to call ConnectSIMUL8).

ConnectSIMUL8 Example
If left$(Signal,7) = "LOADED:" then 'Check first 7 characters only
'OK the model is now loaded
    ConnectSIMUL8 'Ensure SIMUL8 is ready for parameter changes
End If

EmptyQueue
VBA Declaration:
Sub EmptyQueue (qname As String)
Removes (and throws away) all Work Items in the named queue.
EmptyQueue Example

If Signal = "TIMECHECK" Then
If (SimTime() Mod (8*60)) = 0 Then 'End of Shift'
    EmptyQueue "Queue for Work Center 1"
    CollectComplete "Work Center 1"
End If
End If

ExitSIMUL8

VBA Declaration:

Sub ExitSIMUL8 ()
Closes SIMUL8

Get_WI_Held_Resource_Count

VBA Declaration:

Function Get_WI_Held_Resource_Count(ByVal ResName As String)

Find out how many of a resource a work item is using. When at a work center a work item may use more than the minimum requested number of resources. This function lets you find out how many have actually been obtained by the work item.

Works on the "current work item"

Get_WI_Held_Resource_Count Example

If Signal = "DISTRIB: Distrib1" Then
    N = Get_WI_Held_Resource_Count("Resource 1")
    SetDistribution "Distrib1" , 10 + (15/N) , 0 , 0 , 0 , "FIXED"
End If
MinAveMax

VBA Declaration:

Sub MinAveMax (objectname As String, srmin As Single, srave As Single, srmax As Single)

Retrieves results information from a Storage Bin

MinAveMax Example

Dim amin As Single, aave As Single, amax As Single
If Signal = "ENDRUN:" Then
  MinAveMax "Queue for Work Complete 1", amin, aave, amax
  Worksheets("Sheet1").Cells(2, 1).value = amin
  Worksheets("Sheet1").Cells(2, 2).value = aave
  Worksheets("Sheet1").Cells(2, 3).value = amax
End If

ModelSpeed

VBA Declaration:

Sub ModelSpeed(ByVal et As Single)

Adjust the position of the speed scroller - and thus change the speed of the model.

Setting speed zero will stop the model. Speed 100 switches off all graphics and runs the clock in high-speed mode.

ModelSpeed Example

Dim RunNumb As Integer 'Needs to be above SIMUL8_Signal
If Signal = "ENDRUN:" then
  RunNumb = RunNumb + 1
Worksheets("Sheet1").Cells(RunNumb, 1).value = QueueSize("Work Complete 1")
If RunNumb < 10 then
    ResetModel RunNumb
    RunModel 10000
    If RunNumb > 1 then ModelSpeed 100
End If
End If

QueueSize

VBA Declaration:
Function QueueSize (qname As String) As Long

Returns the current number of work items in the queue (storage bin) which has the name qname.

Also works with most other SIMUL8 Objects

QueueSize Example

If Signal = "SET: Attribute 1" Then
    If QueueSize("Storage Area 1") < 10 Then
        SetAttribValue "Attribute 1", 1
    ElseIf
        SetAttribValue "Attribute 1", 2
    End If
End If

ReSetModel

VBA Declaration:
Sub ReSetModel (ByVal randnumberstream As Single)
Reset the clock back to the start of the simulation run and change to random number stream: randnumberstream. If randnumberstream is zero then the random number stream is not changed.

Use StopModel first to ensure the model is not running.

**ResetModel Example**

At the end of the model run put the number of completed work items in column A in the spreadsheet and run the model again using a new random number stream.

```vba
Dim RunNumb As Integer 'Needs for be above SIMLU8_Signal
If Signal = "ENDRUN:" then
    RunNumb = RunNumb + 1
    Worksheets("Sheet1").Cells(RunNumb, 1).value = QueueSize("Work Complete 1")
    If RunNumb < 10 then
        ResetModel RunNumb
        RunModel 10000
    End If
End If
```

**RunModel**

VBA Declaration:

```vba
Sub RunModel (ByVal endttime As Single)
    Set the model running until endtime.
End Sub
```

**RunModel Example**

```vba
If left$(Signal,7) = "LOADED:" then 'Check first 7 characters only
    'OK the model is now loaded
    ConnectSIMUL8 'Ensure SIMUL8 is ready for parameter changes
    BreakDown "Work Center 1", 0
```
RunModel
End If

SaveModel

VBA Declaration:
Sub SaveModel(ByVal fromname$)
    Save the model to disk filename
End Sub

SaveModel Example
Dim RunNumb As Integer 'Needs to be above SIMUL8_Signal
If Signal = "ENDRUN:" then
    RunNumb = RunNumb + 1
    Worksheets("Sheet1").Cells(RunNumb, 1).value = QueueSize("Work Complete 1")
    If RunNumb < 10 then
        ResetModel RunNumb
        RunModel 10000
    If RunNumb > 1 then ModelSpeed 100
    Else
        ResetModel 1
        SaveModel "C:\APPS\S8\MODELS\MYMODEL.S8"
        ExitSIMUL8
    End If
End If

SelectWI_in_Object

VBA Declaration:
Sub SelectWI_in_Object(ByVal Object_Name As String, ByVal Pos_in_Object As Long)
    Change the current work item to be the work item at a position in an object.
SelectWI_in_Object Example

Make speed relative to the number of urgent jobs in the queue.

If Signal = "DISTRIB: Distrib1" Then
    U = 1
    For N = 1 to QueueSize("Storage Bin 1")
        SelectWI_in_Object "Storage Bin 1" , N
        If AttribValue("Attribute   1") = 1 then U = U + 1
    Next N
    SetDistribution "Distrib1", 100/U , 5 , 0 , 0 , "NORMAL"
End If

SetAttributeBatchSize

VBA Declaration:

Sub SetAttributeBatchSize(rname As String, minvalue As Single, maxvalue As Single)

Change a work center's routing-in attribute batch size parameters.

SetAttributeBatchSize Example

In this example attribute 1 contains a product code and we are only prepared to accept
products of code 1, 2 or 5. The work center is set to batch in on attribute 1.

This code gets the work center to take in all items up to but excluding the first work item
which does not have one of the above product codes.

If Signal = "ROUTE_IN_BEFORE: Work Center 1" then
    MX = 0
    For N = 1 to QueueSize("Storage Bin 1")
        SelectWI_in_Object "Storage Bin 1" , N
        PC = AttribValue("Attribute   1")
        If PC = 1 or PC = 2 or PC = 5 Then MX = MX + PC
    Else Exit For
    Next N
SetAttributeBatchSize "Work Center 1", MX, MX
End If

SetAttribValue

VBA Declaration:
Sub SetAttribValue (attname As String, value As Single)
Sets the value of an attribute in the current work item.

SetAttribValue Example

If Signal = "RESET:" Then
For n = 1 To 20
  AddWItoQueue "Work Item Type 1", "Storage Area 1"
  SetAttribValue "Attribute 1", n
Next n
End If

SetCollect

VBA Declaration:
Sub SetCollect (ByVal fromname$, ByVal number As Integer, ByVal toname$)
Change the number of items required by Work Center TONAME from Work Center FROMNAME to NUMBER

SetConveyor

VBA Declaration:
Sub SetConveyor (cname As String, Speed as single, Length as single)
Changes the speed and / or the length of the conveyor cname. If a value (speed or length) is not to be changed set its value to zero.

**SetConveyor Example**

```vba
Dim RunNumb As Integer 'Needs to be above SIMUL8_Signal
If Signal = "RESET:" Then
    RunNumb = RunNumb + 1
    Speed = Worksheets("Sheet1").Cells(RunNumb, 1).value
    Length = Worksheets("Sheet1").Cells(RunNumb, 2).value
    SetConveyor "Conveyor 1", Speed, Length
End If
```

**SetDistribution**

**VBA Declaration:**

```vba
Sub SetDistribution (distname As String, p1, p2, p3, p4 As Single, disttype As String)
Change the parameters of a "named" distribution. P1, P2, P3 and P4 refer to the 1-4 boxes which appear in the named distribution dialog box. (for example, in the case of a NORMAL distribution, P1 is the average, P2 is the Standard Deviation).

disttype should be one of the following:

- FIXED
- AVERAGE
- NORMAL
- NEGEXP
- UNIFORM
- ERLANG
- LOGNORMAL
- WEIBULL
- BETA
- GAMMA
- TRIANGLE
```
If the disttype parameter is left blank (""") then the distribution type will not be changed.
The above spellings should be used - and only capital letters should appear.

SetDistribution Example

mean = Worksheets("Sheet1").Cells(1, 1).value
sd = Worksheets("Sheet1").Cells(1, 2).value
SetDistribution "Distrib1", mean, sd, 0, 0, "NORMAL"

SetMaxLevel

VBA Declaration:

Sub SetMaxLevel (rname As String, value As Single)

Change the maximum size of a queue (or similar parameter for other simulation objects - e.g. replicate level in work centers and total number of resources available in a resource object).

SetMaxLevel Example

If Signal = "TIMECHECK" Then
  If SimTime() > 2400 Then 'Week 2
    SetMaxLevel "Resource 1", 20 'More people in second week
  End If
End If

SetPriority

VBA Declaration:
Sub SetPriority (ByVal WCname$, ByVal Pri As Integer)

Change a work center's priority (assuming at is set to use "Fixed" Priorities).

SetPriority Example

If Signal = "TIMECHECK" Then
    If QueueSize("Queue for Work Center 1") > 5 then
        SetPriority "Work Center 1" , 60
    ElseIf
        SetPriority "Work Center 1" , 40
    End If
End If

SetResourcesRequired

VBA Declaration:

Sub SetResourcesRequired (ByVal resname$, ByVal req As Integer, ByVal wcname$)

Change the number of a particular resource which are required per task at a work center.

SetRouteMode

VBA Declaration:

Sub SetRouteMode(objectname As String, routemode As Long)

Set routing out discipline for a work center.

The variable routemode should be set to one of the following:
Circulate_Route
Uniform_Route
Percent_Route
Priority_Route
Attribute_Route
**SetRouteMode Example**

In this example telephone calls are distributed to a number of call centers using a priority system during the period 8am - 5 pm. Outside this time a uniform rule is used.

```vba
If Signal = "TIMECHECK" Then
    Hour = ((SimTime()/60) Mod 24)
    If Hour >= 8 and Hour < 17 Then
        SetRouteMode "Work Center 1" , Priority_Route
    Else
        SetRouteMode "Work Center 1" , Uniform_Route
    End If
End If
```

**SetRoutePercent**

**VBA Declaration:**

```vba
Sub SetRoutePercent (ByVal fromname$, ByVal PC As Single, ByVal toname$)
Changes the routing out percentages from work center FROMNAME to work center TONAME to percentage PC.
Valid PC values range from 0 to 100. You do not have to check they add to 100% because, after a series of calls to SetRoutePercent you can call AdjustRoutePercent
```

**SetRoutePercent Example**

```vba
If Signal = "TIMECHECK" Then
    Day = SimTime() Mod (8*60)
    If Day = 1 Then
        SetRoutePercent "Work Center 1" , 10 , "Storage Area 1"
        SetRoutePercent "Work Center 1" , 10 , "Storage Area 2"
        SetRoutePercent "Work Center 1" , 80 , "Storage Area 3"
        AdjustRoutePercent "Work Center 1"
End If
```
Else
  SetRoutePercent "Work Center 1", 80, "Storage Area 1"
  SetRoutePercent "Work Center 1", 20, "Storage Area 2"
  SetRoutePercent "Work Center 1", 0, "Storage Area 3"
  AdjustRoutePercent "Work Center 1"
End If
End If

SetWarmUpPeriod

VBA Declaration:

Sub SetWarmUpPeriod(ByVal warm As Single)

Changes the warm up period to warm time units.

SetWarmUpPeriod Example

Having loaded the model, set the warm up period to the value in cell E5 and run the model.

If left$(Signal,7) = "LOADED:" then
  SetWarmUpPeriod Worksheets("Sheet1").Cells(5 , 5).value
  RunModel 10000
End If

SetWITypeLength

VBA Declaration:

Sub SetWITypeLength(witname As String, wilength As Single)

Changes the length of a work item type of name witname to length wilength. This affects the number of work items which will fit on a conveyor.

SimTime

VBA Declaration:
Function SimTime () As Single

Returns the current simulation time in "time units" (as specified in the Clock/Format dialog in SIMUL8).

This is a real number which is set to zero when the "reset clock to start" button is pressed.

**SimTime Example**

In this example SimTime is used to discover if people may be away at lunch.

**IMPORTANT:** Note the use of the variable Hour below. Obviously you can use just about any name here but Excel/VB gets confused if you use Time as a variable name.

```
Hour = (SimTime()/60) Mod 24  "Mod 24" causes Hour to be a number from 0 to 23
If Hour >= 12 or Hour < 24 Then  "It could be lunch
StepModel
```

**VBA Declaration:**

```
Sub StepModel ( )
End Sub
```

Causes a model to move forward in time by one internal simulation event. For a discussion of how much the model will move see: Simulation Monitor.

**How to use: StepModel**

The most common use of StepModel is to add a menu to Excel which can step the model forward a step at a time. Click the tab of a sheet and choose Tools / Menu Editor. Add a menu called STEP and select macro StepModel.

**StopModel**

**VBA Declaration:**

```
Sub StopModel ( )
End Sub
```
Causes a running model to stop.

**StopModel Example**

Dim RowNumb As Integer 'Needs for be above SIMLU8_Signal

If Signal = "TIMECHECK" then
    If QueueSize("Work Complete 1") >= 100 Then
        StopModel
        RowNumb = RowNumb + 1
        Worksheets("Sheet1").Cells(RowNumb, 1).value = SimTime()
    End If
End If

**StopTask**

VBA Declaration:

Sub StopTask (Wname As String)

A currently active work item at the named work center will proceed as if its tasked ended immediately (its scheduled time to finish work is brought forward to the current time).

**StopTask Example**

If Signal = "TIMECHECK" Then
    If (SimTime Mod (8*60)) = 0 Then 'End of Shift'
        EmptyQueue "Queue for Work Center 1"
        StopTask "Work Center 1"
    End If
End If
Utilization

VBA Declaration:

Function Utilization(objectname As String) As Single

Return the percentage utilization of the named object.

Utilization Example
At the end of the run place the utilization of a resource in cell E1.

If Signal = "ENDRUN:" Then
    Worksheets("Sheet1").Cells(5, 1).value = Utilization("Resource 1")
End If
Signals (automatic) from SIMUL8 to Excel

This section is provided for backwards compatibility. Versions of SIMUL8 prior to 4.00 did not support Visual Logic. Places in SIMUL8 where you can now write VL code instead sent “signals” to VBA in EXCEL (or to VB). Models which were built using this feature are automatically updated when opened in version 4 (and above). Direct signals are replaced by single lines of Visual Logic.

For example: Signal VBA ENDRUN:

SIMUL8 automatically sent signals to your Excel or VB extension. You can make use of these signals or ignore them.

DISTRIB: distribution name

The distribution whose name follows the above text is about to be sampled - now is an opportunity to change its parameters (for example).

This message is only sent in the case of "named" distributions with the ASK VBA check box set.

Note there is a space between the colon (:) and the start of the distribution name.

ENDRUN:

The model has just reached the end of the "Results collection time".

LOADED: model name
The model has just been loaded into memory (File / Open or load from command line)

"model name" is the title set in File / Model Details menu. If this is blank then no name is passed in this signal.

**RESET:**

The model clock is being reset to the start time.

**ROUTE_IN_AFTER: work center name**

The current work item has been loaded into a work center. The name of the work center appears after a space which follows the :

Example:
ROUTE_IN_AFTER: Work Center 1

Note there is a space between the colon (:) and the start of the work center name.

**ROUTE_IN_BEFORE: work center name**

The work center whose name follows a space following the : is about to attempt to find a work item to load.

Example:
ROUTE_IN_BEFORE: Work Center 1

Note there is a space between the colon (:) and the start of the work center name.
SET: attribute name

An attribute action unit has been asked to request VB for a value. Set the value of the attribute using SetAttribValue. If you do not call SetAttribValue the attribute's value will be unchanged.

Note there is a space between the colon (:) and the start of the attribute name.

WARM:

The model has finished warming up and the results have just been reset.

Other Signals

See also Time Based Signals (on page 212) and signals from External Distributions (page 213).

Signals from Visual Logic

You can create your own signals from Visual Logic using the “Signal VBA” command. Any text can be sent to the EXCEL SIMUL8_Signals macro.
Time Based Signals

Sometimes there is a need to ask Excel/VB to do a task when there is not a convenient place in SIMUL8 to tell Excel/VB to do (or consider doing) the task.

So that you can give Excel/VB a chance to undertake such tasks SIMUL8 provides the "Clock Extension".

Use the menu CLOCK / EXTENSION to set how often (in simulation time units) you would like SIMUL8 to allow Excel/VB to check the current status of the model (and optionally modify it).

You should use a sensible time interval (one which is often enough to allow Excel/VB to see the things it needs to see - but not so often that your model is slowed down by Excel/VB. Every time Excel/VB is called some code in Excel/VB will have to run - this is quite fast - but unnecessary use of this feature will slow the model down).

Specify a time interval of zero to allow Excel/VB to check status EVERY time ANYTHING in the model changes during running.

The message you place in the LinkExecute box will be passed to your Excel SIMUL8_Signals macro (or VB Form's LinkExecute method) whenever the time interval occurs. Leave this box empty to switch off the Clock Extension.
External Distributions - Signals
You can send a signal to your Excel/VB extension from your SIMUL8 model to tell your macros the context in which it is being asked for data.

The message is text typed into the Signal box External Distribution dialogs.

Excel:

In Excel an External Distribution signal is received in the SIMUL8_SIGNALS macro (like all other signals)

To make use of a signal you had placed in an External Distribution you might want to use code like this:

    ElseIf Signal = "TEST" Then
      Worksheets("Sheet1").Cells(1, 1).Value = 10

Visual Basic:

In Visual Basic an External Distribution signal is received in the Form's LinkExecute Sub (like all other signals)

Without any other code it would look like this:

    Sub Form_LinkExecute (CmdStr As String, Cancel As Integer)
    End Sub

All you need to do is put code into it which responds to the messages you send it from SIMUL8.

For example:

    If CmdStr = "TEST" then Label1 = 10
EXE Programs

Note that, once your Visual Basic Program is built and working you can compile it into a .EXE program which will be automatically run by SIMUL8.

Your users need never know that Visual Basic is being used.

(Use the File / Make EXE command in Visual Basic and enter the program name in your SIMUL8 model's list of "Required Programs" (File / Required Programs in SIMUL8).

If you use this option it means that you do not even need to have Visual Basic on your computer to be able to use Visual Basic programs with SIMUL8.

Object Names from Excel/VB

Names must exactly match the name of the required SIMUL8 object. This IS case sensitive AND ALSO sensitive to the number of spaces between words.

So:

   Storage Area 1

is different from

   Storage area 1

We recommend that you Copy / Paste object names between SIMUL8 and Excel to ensure they are exactly the same (use CTRL-C to copy a highlighted object name in a SIMUL8 dialog box and CTRL-V to paste the name between two quote marks in Excel).
Hiding Visual Basic from the User

Your Visual Basic Form and/or Text Box can be hidden by changing their "Visible" properties in Visual Basic. But it is best not to do this until you know your Visual Basic program is doing everything you want.

See also: Running the Visual Basic Interface without Visual Basic on your computer.

Interfaces to other languages

This help file (and all the SIMUL8 menus and dialogs) are written on the assumption that your own extensions to SIMUL8 will be written using Excel or Visual Basic. However, the interface works using standard WINDOWS DDE and DLL conventions. This means that any language which supports DDE and DLL interfaces can be used.

If you use another language which supports these facilities (e.g. Borland Pascal, C++, Borland Delphi etc.) then take a look at the Visual Basic example programs to see the calling conventions used in the DDE messages and DLL procedures.

SIMUL8’s interface to EXCEL/VBA/Visual Basic uses standard Windows DDE/DLLs. Many software packages and languages support these standards so it is possible to interface SIMUL8 to these other packages.

To understand how read this section in conjunction with the S8DEC.TXT file installed in your SIMUL8 directory. This section assumes you have a good understanding of the principles of DDE and DLLs. Reading the S8DEC.TXT file is easier is you have some understanding of Visual Basic syntax (although most of it is fairly obvious if you have enough programming knowledge to understand DDE and DLLs.).

Overview
You need to use both DDE and the SIMUL8 DLL (S8LIB4.DLL) to add an extension to SIMUL8. This is because 2-way communication has to occur. The DLL is used for the Extension-to-SIMUL8 communication and for the Extension to make inquiries about the current state of the model (e.g. “what is the current simulation time”). DDE is used when SIMUL8 needs to tell the Extension something (in windows terms when an “event” has occurred).

**DLL**

Read the S8DEC.TXT file (and the manual section that describes the EXCEL/VBA/VB interface routines). This should be sufficient to understand the routines available and how to call them.

SIMUL8 must be in memory before the DLL is loaded. So make sure your model is loaded and running before your application loads the DLL. If you use the “Extensions / Visual Basic” dialog in SIMUL8 to specify the name of your .EXE file then SIMUL8 will automatically run it when it is safe to do so. The application in this field does NOT have to be a VB program.

**DDE**

SIMUL8 send “signals” to you extension when it needs the extension to do something. For example, when you have a distribution that is about to be sampled and you may want an opportunity to change its parameters before it is sampled by looking up a database with your extension. In SIMUL8 3.5 and lower these signals were sent automatically either when you had check a box (in the case of a distribution), or always, for example when the model is reset. In version 4 and above Visual Logic replaced the need for most of these signals. The automatic sending was removed to considerably improve performance. In version 4 and above use the Visual Logic command VBA Signal to make signals go to your extension whenever you require.

The manual gives the standard syntax for these signals. However, in version 4 and above there is no need to follow this syntax. For example SIMUL8 sent “RESET:”
whenever the model was reset. As your extension must check for the content of the signal you may use any text.

Your extension application must provide a “DDE SERVICE”. The DDE “Service” and DDE “Topic” are defined in the Extensions/Visual Basic dialog. The “VB Project name” box content is used as the service name. The “Form’s LinkTopic” box is used as the topic. DDE “Items names” are not used unless making use of “External Distributions” (see below).

SIMUL8 sends a xtyp_execute DDE event (“macro execute”) to the service/topic with the data string set to the text in the signal (e.g. RESET: ). Remember that Windows does not guarantee to preserve case through DDE transactions.

On multi-threading operating systems it is possible for applications to run independently of each other and so SIMUL8 might not wait for application to respond to a signal (call the appropriate routines in the DLL to decide what to do and make changes to the model). For this reason SIMUL8 enters a low priority wait state immediately after sending the DDE message until you call S8_Signal_Done in the DLL. If you fail to call S8_Signal_Done SIMUL8 will not respond for a considerable period (there is a timeout set to ensure you can regain control of SIMUL8 if your extension code should fail). The default timeout is 5000 milliseconds, however the timeout period is at least doubled in practice because of additional checking that SIMUL8 does during the time period. If this is too short for your application extend it using DDETimeout=???? In the [startup] section of the S8.INI file. Where ???? is the timeout in milliseconds. (Necessary if your application is, for example, looking up many entries in a database during “RESET:”)

For compatibility reasons S8_Signal_Done is also required on 16 bit machines.

If you use External Distributions with your extension tell the distribution to use VB and set the “Name of label or text box” to your “item name”. In the external distribution dialog, if you enter a string in the optional LinkExecute box this string will be sent as a signal.
Summary

- Enter the name of your .EXE, its DDE service and topic into SIMUL8’s Extensions / Visual Basic dialog.

- Set up your extension application as a service and provide a response to xtyp_execute DDE signals.

- This response must check for the signals from SIMUL8 (most from VBA SIGNAL Visual Logic statements) and process them.

- At the end of processing each (and even if it receives a signal that it decides not to process) it must call S8_Signal.Done.

- Make calls to the S8LIB4.DLL library to instruct SIMUL8 to perform available functions and return available information (see manual for use of EXCEL/VBA/VB interface).

Further advice

Please email info@VisualT.com and ask to be put in contact with one of our SIMUL8 DLL/DDE advisors. In the first email please give a description of the purpose of your extension and the language/system it will use.

"Current" Work Item

The current work item in the context of the Excel Interface is the Work Item which has cause SIMUL8 to ask a question of Visual Basic.
Using Visual Basic with EXCEL

**SIMUL8** can communicate with both EXCEL and Visual Basic at the same time and no special considerations are necessary when doing this.

However, sometimes it is convenient to place much of your data in a spreadsheet in EXCEL and to manipulate it in Visual Basic before passing it to **SIMUL8**. In this case **Visual Basic** must communicate with both EXCEL and **SIMUL8**. One or two tips are needed here because Visual Basic needs to know which way it is talking at what time!

Have a look at the help in Visual Basic to see how it normally talks to EXCEL. You will see something like this:

```vbnet
row = row + 1
If row = 1 then
    Label1.LinkMode = None
    Label1.LinkTopic = "EXCEL|C:\SPREAD.XLS"
    Label1.LinkItem = "R1C1"
    Label1.LinkMode = Automatic
Else
    Label1.LinkItem = "R" + row + "C1"
End If
```

If the value from the spreadsheet is to be passed to **SIMUL8** (or used in a calculation which results in something being passed to **SIMUL8**) you should be aware that Visual Basic does not update Label1 immediately. Automatic mode means "do it for me without me coding it each time BUT do it when you have time"....this is not much good in a high speed simulation!

For this reason you should change the above to:

```vbnet
row = row + 1
If row = 1 then
    Label1.LinkMode = None
    Label1.LinkTopic = "EXCEL|C:\SPREAD.XLS"
    Label1.LinkItem = "R1C1"
    Label1.LinkMode = Manual
Else
    Label1.LinkItem = "R" + row + "C1"
End If
```
Else
    Labell.LinkItem = "R" + row + "C1"
End If
Labell.LinkRequest

LinkRequest tells VB to update the label immediately.

Secondly, note that Visual Basic cannot talk to two packages at the same time using the same control. This means that if Label1 contains a value which has come from EXCEL (as in the above code), you should not use Label1 in SIMUL8.

Instead use, say, Label2 and place

Label2 = Labell

after the LinkRequest line above.

If you have any problem getting Visual Basic and EXCEL to talk to each other, try inserting "DoEvents" calls into your Visual Basic program. This gives Excel time to respond to requests it gets from Visual Basic (See Visual Basic help for more information on DoEvents).

Visual Basic Project Name

When you save a Visual Basic Project you will be asked for a name. The name you give will become the project's name and should be used in SIMUL8's information about Visual Basic Extensions.

Note that, when you change a Visual Basic Project name (with SAVE or SAVE AS in Visual Basic) Visual Basic does not recognize the new name (for purposes of communication with other software) until after you exit from Visual Basic and reload it.
Simple Importing of Data from Visual Basic

If you simply wish to use external data in your SIMUL8 model and that data can replace a SIMUL8 distribution, then use an External Distribution as described on page 112.

If the data is not fixed, or cannot easily be represented in a spreadsheet, and you would like the data to be in a dialog box of your own design which the model user can use to adjust the data, then create an external distribution which uses Visual Basic as the source of the data.

Use the EXTENSIONS / VISUAL BASIC item in the main menu to specify the name of the VB program which your model is to use.

If you have not separately created a Visual Basic program you can use this dialog to let SIMUL8 do it for you. Just enter a new project name into the project name edit box and click CREATE. SIMUL8 will create a new VB project for you and load it onto the screen. A help screen will also appear to take you step by step through using this facility.

The Visual Basic program which is created contains example code showing you how to change a distribution in SIMUL8 from Visual Basic. To change this so that the Visual Basic program created, instead passes data back to SIMUL8 using an “external distribution” proceed as follows:

Delete all the code in the Form “LinkExecute” Sub.

In SIMUL8 create an External distribution. Click the Visual Basic check box and enter TEXT1 in the “Name of label or text box” entry.

Leave the other field blank.
Run the Visual basic program.

When you run the SIMUL8 model it will use the value in the text box as the sample value for this new distribution.

Values returned always have to be in either a label or a text box and must be on a form which has its LinkMode set to Source and its LinkTopic set to the same name as in the SIMUL8 Extensions / Visual Basic dialog. If this does not work make sure you have NOT checked the USE VBA check box in the SIMUL8 Extensions / Excel dialog.

If you want to change the value in the label or text box automatically from within the Visual Basic program either use the Clock / Time Check dialog (see page 212) or:

In the external distribution dialog enter any text into the LinkExecute command (optional) field. This text will be passed to the form’s LinkExecute Sub immediately prior to the value being read from the text box or label.

Into the LinkExecute Sub type some code like this:

```
If CmdStr = "SAMPLINGMYDIST" Then
   Label1.Caption = "22"
End If

or

If CmdStr = "SAMPLINGMYDIST" Then
   If QueueSize("Queue for Work Center 1") > 3 Then
      Label1.Caption = "1"
   Else
      Label1.Caption = "2"
   End If
End If
```
If you have already created a Visual Basic program which you want to link into **SIMUL8** then:

Specify the name of the Project (the default in Visual Basic is PROJECT1) and the name of the Form's LinkTopic (the default in Visual Basic is the same as the name of the form - e.g. FORM1). Make sure that in Visual Basic you set the form’s “link mode” to “source”.

If you turn your Visual Basic program into a .EXE you can select this .EXE file in the **SIMUL8** Extensions / Visual Basic dialog so that **SIMUL8** will automatically load your .EXE when it is needed.
A Free Service from Visual Thinking

The Visual Basic Interface is easy to expand. If any of SIMUL8’s users would like their Visual Basic Programs to be able to know more about what is happening inside your SIMUL8 model - just let us know.

Give us a clear definition of what you would like to know and how you would like the information returned to you and we will add the appropriate extra facility to S8DEC.BAS and send you a new version.

This is a free service.

Similarly, if there is somewhere additional in your model where you would like Visual Basic to be able to determine the actions your model takes, let us know.

Give us a clear definition of the way you would like Visual Basic to be able to control your model. If your need will not make SIMUL8 difficult for other users to use, then we will add the requested facility and send you an update.

This is a free service.
Technical Examples

The following technical example models are all installed to your hard disk when you run the SETUP program on the SIMUL8 distribution disk 1. They are installed in the directory which you specify during the setup procedure.

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATACTION</td>
<td>Attribute Actions and Routing.</td>
</tr>
<tr>
<td>ATTDIST</td>
<td>Shows how a work center can use different distributions for different work items.</td>
</tr>
<tr>
<td>BATCH</td>
<td>Batched Arrivals of Work to do.</td>
</tr>
<tr>
<td>BOTTLES</td>
<td>Shows the use on non-standard icons for products (on a stream of conveyors).</td>
</tr>
<tr>
<td>CLOSEQ</td>
<td>Shows the use of breakdowns which do not stop a work center until a storage bin is empty.</td>
</tr>
<tr>
<td>EXAMPLE2</td>
<td>An example of paper work flows.</td>
</tr>
<tr>
<td>LOOP</td>
<td>Shows how you can run a model of a contained system which does not have any &quot;Work Starts&quot; or &quot;Work Completes&quot; Objects and which uses Storage with some Initial Contents.</td>
</tr>
<tr>
<td>PRDCOUNT</td>
<td>Shows how to use Visual Basic to tell one work center to stop working when another has done a certain amount of work.</td>
</tr>
<tr>
<td>PRIWRKC</td>
<td>Priority for a work center in using labor (resources). The four machines in the center of the screen each require one unit of labor to operate. Only three units of labor are available.</td>
</tr>
<tr>
<td>PRIQ</td>
<td>Certain work items are given priority (in a queue) by using an attribute on the work items to sequence them in the queue.</td>
</tr>
<tr>
<td>TIMEDD</td>
<td>Illustrates how to use Time Dependent Distributions.</td>
</tr>
<tr>
<td>VBDEMO1</td>
<td>A simple example of how to use Visual Basic with SIMUL8.</td>
</tr>
<tr>
<td>VBS8</td>
<td>See: Visual Basic. This is not an example, instead it is a shell for using Visual Basic with SIMUL8 which can be converted to your own needs.</td>
</tr>
</tbody>
</table>
Installation Guide

Standard Windows Installation

To install **SIMUL8** on your computer proceed as follows:

Place **SIMUL8** disk 1 into your computer’s A: drive.

(If you do not have an A: drive, you can use any other drive - just substitute your drive letter for A: in the instructions below)

**In WINDOWS 95 or NT 4**

click Start, Run and enter A:\SETUP

**In WINDOWS 3.1x or NT 3.5x**

from WINDOWS PROGRAM MANAGER choose the FILE / RUN menu and type A:\SETUP into the box, then press the ENTER key or click the OK button.

Read the instructions on the screen. These will take you step-by-step through the installation process.

You will be able to choose the location where **SIMUL8** is stored on your hard disk.

**SIMUL8** requires about 25MB of hard disk space.
When installation is complete, you can click the RUN button (or EXIT back to Windows).

If you click RUN, you will be shown an example SIMUL8 simulation - so you can see what a SIMUL8 model looks like. Choose Help / Getting Started to run the tutorial learn how to build your own models.

If you have any difficulty please call us for support. We want you to enjoy using SIMUL8 to improve your business. See the front of this manual for contact information.

**Network Installation**

This information is provided for network administrators on setting up SIMUL8 to run on a network.

Remember you must have enough licenses to cover the maximum number of computers which could possibly run SIMUL8 simultaneously.

SIMUL8 can be installed from a network drive. Copy all the disks to the same directory on the network and then replace that network drive/directory name in the above instructions where you see “A:\”.

Once you have installed SIMUL8, if it is installed in a shared directory you are advised to write-protect the files in the SIMUL8 directory and its sub-directories. This enables all users to access copies of examples (and other files) which will not have been accidentally changed by other users.
If the directory containing the EXE and DLL files for SIMUL8 is write protected when users are running SIMUL8 then you should ensure the INI file (S8.INI in the same directory as S8.EXE):

DOES NOT contain any lines which start FILE?=

(These will be added if you should run SIMUL8 while the directory is NOT write protected)

DOES CONTAIN a line (in the [STARTUP] section):

RunnableNumber=20

SIMUL8 has been designed to work well from shared network directories.

Support contact information is provided in the front of this manual. The web site contains a support forum.
# INDEX

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy Of Results</td>
<td>39</td>
</tr>
<tr>
<td>Add To The List Of Action Attributes</td>
<td>90</td>
</tr>
<tr>
<td>Add Work Item To Queue Command</td>
<td>158</td>
</tr>
<tr>
<td>Adding Attributes To Work Items</td>
<td>88</td>
</tr>
<tr>
<td>Additional Graphics</td>
<td>127</td>
</tr>
<tr>
<td>Addwitoqueue</td>
<td>181</td>
</tr>
<tr>
<td>Adjustroutepercent</td>
<td>181</td>
</tr>
<tr>
<td>Aligning Objects On The Display</td>
<td>72</td>
</tr>
<tr>
<td>Animation</td>
<td>95</td>
</tr>
<tr>
<td>Antithetic</td>
<td>102</td>
</tr>
<tr>
<td>Ask Vba Check Box</td>
<td>107</td>
</tr>
<tr>
<td>Attribute Actions</td>
<td>89</td>
</tr>
<tr>
<td>Attribute Based Distributions</td>
<td>113</td>
</tr>
<tr>
<td>Attribute Batching</td>
<td>81</td>
</tr>
<tr>
<td>Attribute Text - Setting</td>
<td>91</td>
</tr>
<tr>
<td>Attribute Value</td>
<td>91</td>
</tr>
<tr>
<td>Attributes Containing Numbers</td>
<td>89</td>
</tr>
<tr>
<td>Attributes Containing Text</td>
<td>89</td>
</tr>
<tr>
<td>Attributes Control Work Item Images</td>
<td>92</td>
</tr>
<tr>
<td>Attributes List</td>
<td>89</td>
</tr>
<tr>
<td>Attributes</td>
<td>88</td>
</tr>
<tr>
<td>Atttribvalue</td>
<td>182</td>
</tr>
<tr>
<td>Auto Change</td>
<td>101</td>
</tr>
<tr>
<td>Avesize</td>
<td>182</td>
</tr>
<tr>
<td>Avetimein</td>
<td>183</td>
</tr>
<tr>
<td>Avetimeinbycategory</td>
<td>183</td>
</tr>
<tr>
<td>Batching By Type</td>
<td>82</td>
</tr>
<tr>
<td>Beep</td>
<td>158</td>
</tr>
<tr>
<td>Breakdown</td>
<td>158</td>
</tr>
<tr>
<td>Breakdown</td>
<td>184</td>
</tr>
<tr>
<td>Breakrestart</td>
<td>159</td>
</tr>
<tr>
<td>Breakrestart</td>
<td>184</td>
</tr>
<tr>
<td>Build Model, Then Refine It</td>
<td>32</td>
</tr>
<tr>
<td>Building Visual Logic</td>
<td>145</td>
</tr>
<tr>
<td>Building Your First Simul8 Model</td>
<td>19</td>
</tr>
<tr>
<td>Business Process Re-Engineering</td>
<td>29</td>
</tr>
<tr>
<td>Changing The Work Item During A Task At</td>
<td></td>
</tr>
<tr>
<td>A Work Center</td>
<td>83</td>
</tr>
<tr>
<td>Circulate Routing Rule</td>
<td>84</td>
</tr>
<tr>
<td>Clearallresourcesrequired</td>
<td>185</td>
</tr>
<tr>
<td>Clock</td>
<td>75</td>
</tr>
<tr>
<td>Closing A Sub-Window</td>
<td>68</td>
</tr>
<tr>
<td>Collect Work Items</td>
<td>79</td>
</tr>
<tr>
<td>Collectcomplete</td>
<td>186</td>
</tr>
<tr>
<td>Combination Distributions</td>
<td>113</td>
</tr>
<tr>
<td>Combining Time Graphs</td>
<td>125</td>
</tr>
<tr>
<td>Conclusion</td>
<td>48</td>
</tr>
<tr>
<td>Connectsimul8</td>
<td>186</td>
</tr>
<tr>
<td>Contents Button</td>
<td>128</td>
</tr>
<tr>
<td>Converting Resources to Different Types</td>
<td>63</td>
</tr>
<tr>
<td>Conveyors</td>
<td>59</td>
</tr>
<tr>
<td>Copy</td>
<td>70</td>
</tr>
<tr>
<td>Current Work Item</td>
<td>142</td>
</tr>
<tr>
<td>Current Work Item</td>
<td>208</td>
</tr>
<tr>
<td>Definition</td>
<td>14</td>
</tr>
<tr>
<td>Delete Work Item</td>
<td>159</td>
</tr>
<tr>
<td>Deleting A Route Arrow</td>
<td>78</td>
</tr>
<tr>
<td>Deleting Objects</td>
<td>65</td>
</tr>
<tr>
<td>Detail / Auto Efficiency</td>
<td>99</td>
</tr>
<tr>
<td>Detailed Description And Examples Of Excel Interface Library</td>
<td>181</td>
</tr>
<tr>
<td>Diagnostic Facilities</td>
<td>128</td>
</tr>
<tr>
<td>Display</td>
<td>26</td>
</tr>
<tr>
<td>Distrib: Distribution Name</td>
<td>202</td>
</tr>
<tr>
<td>Distribution Detail</td>
<td>104</td>
</tr>
<tr>
<td>Distribution Parameters</td>
<td>103</td>
</tr>
<tr>
<td>Distribution Type</td>
<td>104</td>
</tr>
<tr>
<td>Distributions Used For Efficiency</td>
<td>99</td>
</tr>
<tr>
<td>Documenting Models</td>
<td>29</td>
</tr>
<tr>
<td>Drawing The Routing</td>
<td>77</td>
</tr>
<tr>
<td>Due Attribute</td>
<td>93</td>
</tr>
<tr>
<td>Efficiency And Breakdown</td>
<td>99</td>
</tr>
<tr>
<td>Empty Simulation Object</td>
<td>160</td>
</tr>
<tr>
<td>Emptyqueue</td>
<td>186</td>
</tr>
<tr>
<td>Enable/Disable Undo</td>
<td>73</td>
</tr>
<tr>
<td>Endrun:</td>
<td>202</td>
</tr>
</tbody>
</table>
Stoppages - Special Cases 99
StopTask 200
Storage Bins 57
Structure Of Your Model 49
Sub-Simulation Windows 67
Switch Route Lines On/Off 78
Tailored Interactive Aspects 27
Technical Examples 215
The Main Concepts 49
The Process 16
The Simul8 / Excel Extension 174
The Simulation Time 75
The Structure Of Visual Logic In Your Simulation Model. 135
The Structure Of Visual Logic Itself. 138
Time Absolute Distributions 114
Time Based Signals 205
Time Dependent Distributions 107
Time Graph Button On The Tool Bar 125
Time Graphs 124
Timing And Distributions 103
Traveling Time 86
Trials 126
Undo 72
Uniform Distributions 106
Uniform Routing Rule 84
Until 172
Using Visual Basic With Excel 208
Utilization 200
Verification 34
Visual Basic Project Name 210
Visual Exploration 38
Visual Logic Basics 141
Visual Logic Command Details 157
Visual Logic Commands By Category 155
Visual Logic Extension User Guide 133
Visual Logic Introduction 133
Visual Logic Structure 135
Visual Logic 29
Vl Blocks 141
Warm Up Period 75
Warm Up Time 35
What: 204
What Next 30
When Are Resources Released? 63
When Are Resources Required? 64
Where Random Numbers Are Used 101
While 172
Work Centers 55
Work Closely With Your Client 33
Work Entry Points 59
Work Exit Points 60
Work Items 51
Zoom 29

234