Using **Extend**

**Extend** is a complete simulation environment that runs under Windows.

It supports both continuous and discrete simulation.

It employs a multi-window graphical user interface for assembling model structure. The primary technique for laying in a model is "drag and drop" using blocks from supplied libraries.

Block elements include a dialogue, iconics, and procedural routines. The procedural routines are written in a "C-like" language called **ModL**.

Library blocks, including their procedural routines are fully accessible and can be copied and modified for special purposes.

A basic graphical layout for a *single-server queue* model looks like

- "Input Random" block (in the *Generic.lix* library) for sampling from probability distributions
- "Model "Executive" block (in the *Discrete Event.lix* library)
- Configurable "Generator" block (in the *Discrete Event.lix* library) for bringing items into the model
- "Queue FIFO" block (in the *Discrete Event.lix* library) for receiving items in the model flow
- "Exit" block (in the *Discrete Event.lix* library) for removing items from the model
- "Activity Delay" block (in the *Discrete Event.lix* library) for modeling activities such as a server performs

---

*Image description*: A diagram showing a single-server queue model with blocks and connections, including "Executive", "Generator", "Queue FIFO", "Exit", and "Activity Delay".
The simulation is started by executing "run" from the run tab or by clicking the icon on the task bar (a simulation, even in its initial set up will almost always run, but until configuration details and output display are in place it will not do anything particularly meaningful).

By turning on "Show Animation" under the run tab, the simulation graphics will show simulation progress as the simulation executes, which is useful in initial stages of simulation construction.

To bring up a block's dialogue, just double-click on it. The dialogue is where any permitted configuration values are set (in some cases, configuration values can be adjusted "on the fly" by use of prescribed "block inputs").

For example, to configure the Generator block (Discrete Event.lix), double-click on it to bring up its configuration dialogue. Note that for animation, the animation figure can be any one of a number of different representations.
The distribution is selected (including empirical) just by clicking on its dialogue input box. Any parameters required by the distribution can then be set.

Similarly, by double-clicking on an “Input Random” block (Generic.lix), the distribution (including empirical) to be sampled from can be established.

The overall simulation can be set up as a "timed" simulation via the **Run tab … Simulation Setup/Discrete event** tab. If the setup is configured for "hours" and the **Generator** blocks (Discrete Event.lix) and **Input Random** blocks (Generic.lix) are configured for "minutes", **Extend** will automatically take care of the time conversions. Note that the **Simulation Setup/Time units** tab provides the means for changing how time conversions are done (e.g., to have 8 hour days, rather than 24 hour days).

A simulation model can easily be changed from being timed to being one terminated on a count value. This is one of the options provided in the dialogue of the **Executive** block (Discrete Event.lix), permitting the stop criteria to be either an event count, or a value input through the **count** connector. Note that the basic single server queue model has two events associated with each item … the arrival event and the departure event, which illustrates that this way of stopping a simulation is likely to be problematic. The specified stop amount is treated as a count when there is a connector into the count tab.

A display or I/O box in a block's dialogue can be "cloned" onto the model display (actually residing in the "Clone" layer) by using the **clone tool** to drag a copy from the dialogue onto the model display.

There are actually three "layers" on the display:

- the **Block/Text layer**
- the **Draw layer**
- the **Clone layer**

each having its own tool. There is also an "all layers" tool for positioning items for best line-up across layers.

Once a dialogue box is cloned onto the display, the “all layers” tool can be used to resize and position it appropriately. As the simulation executes, a cloned display box changes just as it does in the dialogue (which can be seen if the dialogue is left open).
Other useful blocks:

One of the things we look for in a simulation is feedback as the simulation runs that gives us some sense of what it is doing.

Under the **Model ... Controls** tab there are 3 controls:

- **Slider** - a graphical display for producing a value in a \((\text{min}, \text{max})\) range
- **Switch** - a graphical display for producing a 0 or 1 value (true or false)
- **Meter** - a graphical display for exhibiting a value

The “**Status**” block (\textit{Discrete Event.lix}) is used to display the value of another block’s output. If items are input, the interarrival time from the previous item to the current one is placed at the output labeled \textit{I}.

The “**Mean & Variance**” block (\textit{Generic.lix}) is used to obtain statistical for the input to the block (the mean is placed at the output labeled \textit{M}).

The “**Show Times**” block (\textit{Discrete Event.lix}) shows the event times for blocks that produce events. It also shows the current simulation time.

When arithmetic calculations are needed, the “**Constant**” block (\textit{Generic.lix}) is used to make a constant available in calculations, and “**Add**”, “**Subtract**”, “**Multiplication**”, and “**Divide**” blocks (\textit{Generic.lix}) are used to perform standard binary operations on the values input.

There is also a general "**Exponent**" block (\textit{Generic.lix}) for calculating \(x^y\).

There is a general "**Equation**" block (\textit{Generic.lix}) for calculating expressions of the form

\[
\text{result} = \text{formula};
\]

for as many as 5 inputs. **Extend**'s operators are:

+ , - , * , / , ^ (exponentiation), MOD or \(\%\) (modulo), AND or \&\& , OR or \|\| , NOT or \! , == (equals), != or <> (not equal), < , <= , > , >=
Connecting Blocks:

There are two basic types of connectors for blocks in *Extend*:

1. value input and output connectors (enlarged)
2. item input and output connectors (enlarged)

There is also a “universal input connector” (enlarged) that can take as input either values or items.

**Items** are simply the basic discrete event units that are passed between blocks. **Values** are actual data generated or used in the model as it operates.

The idea is to connect from value output to value input (e.g., connect the output from an Input Random block to the input of an Activity Delay block) or to connect from item output to item input (e.g., the output of a Generator Block to the input of a Queue FIFO Block).

On the model this is done using the “drawing pen”, which appears when the block/text layer tool is placed over a connector and the left mouse button is clicked and held.

Using standard “click and drag” from one connector to another of opposite type causes a tentative connection line which darkens when a connection match is made. Releasing the mouse button causes the connection line to affix between the two points.

Different types of connection lines can be configured by use of the *Model … Connection Lines* tab.

Reporting:

*Extend* provides a crude report generator under the Run tab. The report information is generated block by block for those blocks the user has selected (to select multiple blocks at once, click on them while holding the shift key - under the Run tab click on "Add Selected to Report"). Reporting only gives final values. There is a (voluminous) model tracing capability if intermediate values are of interest. The "Dialogs" report type includes the values and setting for all parameters of a block (useful for documentation purposes). The "Statistics" report type reports on final values of output parameters only.

Reports are written to a text file from which model outcomes can be extracted for a more polished report and analysis of a simulation. At the end of the simulation run, the report is automatically opened and displayed from within *Extend*. 