



GUIA DE SIMULAÇÃO

Versão do FlexSim 7.3

**Data da Publicação no Brasil
10 de Outubro de 2014
Por Michael Machado e Flávio Brito.**

Tabela de Conteúdo

Iniciando o uso

- Iniciando com o FlexSim
- Interagindo com o FlexSim
- Interagindo com o Teclado
- Terminologia do FlexSim
- Primeiro Modelo
- Ativação de Licença
- Conceitos de Ativação de Licenças
- Exemplo de Ativação de Licença
- Referência de Ativação de Licença

Conceitos do FlexSim

- Overview dos conceitos do FlexSim
- Flowitems
- Portas
- Itemtype
- Labels
- Item e Current
- Return Values
- Picklists
- Template Code
- Model Tree View

Tutoriais

- Introdução Exercícios
- Exercício 1
- Exercício 1 Introdução
- Exercício 1 Construção Passo a Passo
- Exercício 2
- Exercício 2 Introdução
- Exercício 2 Construção Passo a Passo
- Exercício 2 Extra Mile
- Exercício 2 Extra Mile Introdução
- Exercício 2 Extra Mile Construção passo a passo
- Exercício 3
- Exercício 3 Introdução
- Exercício 3 Construção do Passo a Passo
- Labels Tutorial
- Labels Introdução
- Labels Construção do Modelo Passo a Passo
- Global Modeling Tools

Ferramenta de Modelagem

- Animation Creator
- Animation Creator Concepts
- Animation Creator Example
- Animation Creator Reference
- AVI Maker
- AVI Maker Concepts
- AVI Maker Example
- Flowitem Bin
- Flowitem Bin Concepts
- Flowitem Bin Reference
- *Global Tables*
- Graphical User Interfaces
- Graphical User Interfaces Concepts
- Graphical User Interfaces Example
- Graphical User Interfaces Reference
- Media Files
- Model Background
- Visio Importer

Gráficos e Relatórios

- Dashboard
- Dashboard Concepts
- Dashboard Example
- Dashboard Reference
- Dashboard Graphs
- Dashboard Date and Time Display
- Dashboard General Pages
- Dashboard HTML Statistic
- Dashboard Objects Page
- Dashboard Statistics Page
- Dashboard Tracked Variables
- Dashboard Utilization Analysis Page
- Model Input Properties
- FlexSim Chart
- FlexSim Chart Concepts
- FlexSim Chart Application
- FlexSim Chart Summary Report
- FlexSim Chart State Report
- FlexSim Chart Object Comparison Chart
- FlexSim Chart Time Plot
- FlexSim Chart Financial Report
- FlexSim Chart Object Gantt Chart
- FlexSim Chart Single Object Chart

- Global Modeling Tools Introdução
- Global Modeling Tools Construção do Modelo Passo a Passo
- User Events
- User Events Introdução
- User Events Construção do Modelo Passo a Passo
- TimeTables
- TimeTables Introdução
- TimeTables Construção do Modelo Passo a Passo
- Kinematics
- Kinematics Introdução
- Kinematics Construção do Modelo Passo a Passo
- Task Sequence
- Task Sequence Introdução
- Task Sequence Construção do Modelo Passo a Passo
- Task Sequence 2
- Task Sequence 2 Introdução
- Task Sequence Construção do Modelo Passo a Passo
- Task Sequence 3
- Task Sequence 3 Introdução
- Task Sequence Construção do Modelo Passo a Passo
- SQL Tutorial
- SQL Tutorial Introdução
- SQL Tutorial Construção do Modelo Passo a Passo
- Fluido Objetos
- Fluido Objetos Introdução
- Fluido Objetos Construção do Modelo Passo a Passo

- FlexSim Chart Flowitem Gantt Chart
- FlexSim Chart DatGuiase Tables
- FlexSim Chart Interacting with Graphs
- FlexSim Chart Editing Chart Variables
- Reports and Statistics

Experimenter Optimizer

- Experimenter
- Optimization in FlexSim
- Experimenter Optimizer Example

3D Media

- Importing 3D Media
- Peparing a 3D File
- Importing AutoCAD Drawings
- Shape Factors
- Shape Frames
- Level Of Detail

Visualização dos Modelos

- Orthographic Perspective View
- Tree Window
- Travel Networks
- Library Icon Grid
- View Settings
- Light Source Editor
- Quick Properties
- Edit Selected Objects
- Find Objects
- Views
- Groups
- Model Layouts
- Measure Convert

Menu Principal e Toolbar

- [File Menu](#)
- [Edit Menu](#)
- [View Menu](#)
- [Build Menu](#)
- [Execute Menu](#)
- [Statistics Menu](#)
- [Tools Menu](#)
- [Debug Menu](#)
- [Help Menu](#)
- [FlexSim Toolbar](#)
- [Simulation Run Panel](#)

[Biblioteca de Objetos do FlexSim](#)

- [FlexSim Object Library Overview](#)
- [FixedResources](#)
- [FixedResources Concepts](#)
- [TaskExecuters](#)
- [TaskExecuters Concepts](#)
- [Travel Networks](#)
- [NetworkNode](#)
- [TrafficControl](#)
- [VisualTool](#)
- [VisualTool Overview](#)
- [VisualTool Example](#)
- [Fluid Library](#)
- [Fluid Library Concepts](#)

FlexSim

3D Simulation Software

Bem vindo ao guia de simulação com o FlexSim

Se esta é a sua primeira vez utilizando o FlexSim, experimente algumas das seguintes sugestões:

1. Visite o [Canal Youtube da FlexSim, USA](#) e o [Canal Youtube da FlexSim Brasil](#) e assista como construir seus primeiros modelos, vídeos com dicas sobre como usar o FlexSim. Usando o Mozilla Firefox para assistir os vídeos no [canal Youtube da FlexSim Brasil](#), você pode ativar as legendas para acompanhar a descrição do vídeo em português. Agora também temos o [canal do Vímeo da FlexSim Brasil](#), com a descrição dos vídeos em português apenas em “Read More”. Aproveite!
2. Faça os exemplos deste guia na versão gratuita do Flexsim. Baixe a versão gratuita do Flexsim em <http://www.flexsim.com/br/account/signup/>
3. Leia a seção Iniciando o Desenvolvimento de Modelos (Getting Started section) e siga os passos a passo para o desenvolvimento de modelos. Somente os três primeiros exercícios é possível de ser completado com a versão demo.

Divirta-se ao desenvolver o seu primeiro modelo do princípio!

Sinta-se à vontade para contatar nosso suporte técnico se você tiver alguma dúvida enquanto estiver avaliando o software. O suporte técnico está disponível de segunda à sexta-feira, das 08h30min até as 17h00min.

Você pode telefonar para +55 (19) 3308 – 7833 (FlexSim Brasil) ou ainda +1 801-224-6914 (FlexSim, USA), enviar um e-mail sobre sua dúvida utilizando nosso formulário web disponível em <http://www.flexsim.com/br/account/login/>, ou postar sua dúvida na comunidade mundial da FlexSim, o FlexSim Community Forum (<http://www.flexsim.com/community/forum/>). Esperamos que você goste da avaliação com o FlexSim e que ele possa ter ajudado você na otimização do fluxo de seu processo. Você pode também ir em “O que há de novo” para ver o que foi adicionado nas últimas atualizações do FlexSim.

O que há de novo no FlexSim

A seguinte lista de recursos e correções foram incluídas na última versão do FlexSim.

FlexSim 7.3.4

- Arrumado alguns exceptions sendo disparados quando usava o Debugger.

FlexSim 7.3.2

- Corrigido um bug na Time Table e no MTBF/MTTR para permitir a configuração de indisponibilidade.
- Corrigido um bug com o Optimizer GUI quando selecionava a opção Manual stop only.
- Corrigido um bug no merge sort que não estava resetando a tabela de dados de entrada corretamente.
- Corrigido a unidade Gallons para não mais ficar aparecendo a mensagem como Fluid Ounces.
- Corrigido a questão de modelos antigos não estarem abrindo corretamente Global Tables do toolbox.
- Corrigido um bug no queue quando a capacidade máxima não for um número inteiro.
- Retorno do ExpertFit.

FlexSim 7.3.0

- Adicionado o recurso de salvamento automático em global preferences.
- Adicionado a opção de sombreamentos suaves ou de alta intensidade.
- Melhora no stereographic 3D incluindo o suporte a sombreamento.
- Adicionado um floor object.
- Added alignment property to dashboard model input widgets.
- Added a dashboard GUI class model input widget.
- Added support for linking to groups in dashboard statistics widgets.
- Added support for selecting multiple dashboard widgets.
- Dashboard dynamic model input fields and edit fields can be tied to global variables.
- Improved support for Full Screen mode including stereographic 3D and multiple monitors (F11 to switch in and out of full screen).
- Melhorar no recurso do mouse com imagens 3D.
- Removido o menu Tools e adicionado o Toolbox.
- Adicionado o modo Data Based nas Time Tables.
- Added optional stateprofile parameter to stopobject and resumeobject commands.
- Adicionado Stop Date & Time ao Experimenter.
- Arrumado as classes GUI disponíveis em Reports e Statistics.
- Fixed an issue with adding/editing triggers in the dashboard causing FlexSim to crash.
- Fixed an issue with exporting multiple TrackedVariables statistics to CSV files.
- Fixed an issue with aligning model input objects to the left/top of the dashboard.
- Fixed Content vs Time graph not collecting data when defined for a time interval.
- Fixed an issue with setting long object names/paths in bundles and dashboard statistic objects.
- Fixed tabbing in Quick Properties causing FlexSim to crash.
- Added flexscript commands for use with Stat::Fit.
- Added search results to Command Helper window.
- Added copy/paste functionality to Dashboard (Model Input Widgets only).
- Updated the Automatic Data Distinction in the Excel Importer to more correctly import numbers.
- Added sql query() support for LIKE, GROUP BY, and non-trivial expressions in the SELECT column clause.
- Update the Excel Import/Export to handle importing into bundles and export bundle data.

Backwards Compatibility Note: the following changes may slightly change the way updated models behave.

- Expressions can now be used in select columns of an sql query. In order to do this we needed to change our sql grammar to be more compliant with standard SQL. This means you can no longer use the AS clause in a WHERE or ORDER BY statement. You can now only use AS in SELECT and FROM clauses.
- Some parameter types for the query command have changed in c++ for more flexible type definitions. This means if you use the query() command in a c++ dll, you'll need to update your dll's headers and recompile the dll. If you use straight c++ compiled code or flexscript for your query() command, then this should just update.
- Removed expertfit. Use Stat::Fit.

FlexSim 7.1.4

- Fixed an issue with the Excel Import not properly handling importing multiple sheets from the same file.
- Fixed Basic Conveyors and Merge Sorts not drawing with shaders on.
- Fixed loading SimpleDataType nodes in modules in the Experimenter.

FlexSim 7.1.2

- Fixed a memory leak with the 3D view.
- Fixed GlobalTables so they save their cell width and cell height.
- Fixed a bug in the MTBF/MTTR that caused the downtime value in the down function to always be 0.
- Fixed a bug causing Keyframe triggers to not be fired in operator animations.
- Fixed Animation Editor > Draw Surrogate > Main Object Content.
- Fixed Animation Editor > Component > Rotational Centroid fields.
- Fixed issues with editing animations in models where units were not meters.
- Fixed an issue causing global variables to be renamed to NULL when creating multiple global variables at the same time.
- Fixed exceptions being thrown when you try to use shaders in compatibility mode.
- Fixed issues displaying skp files while using the A* Module.
- Fixed an issue with not being able to select views in the GUI builder.
- Fixed DLL loading to properly load DLLs from the model directory.

FlexSim 7.1.0

- Added Financial Analysis Dashboard Widget.
- Added Custom Dashboard Widget to allow any numeric data to be displayed in the dashboard, including table data, bundle data, global variables, etc.
- Added State Gantt Chart Dashboard Widget.
- Added a FlowItem Trace Gantt Chart Dashboard Widget.
- Dashboard table data can display current state values as strings.
- Added functionality to export dashboards to HTML.
- Moved the statistics tab out of object properties windows and into the Quick Properties window.

- Labels, tables, statistics, global variables and tracked variables can be "pinned" to the dashboard.
- More options to customize Dashboard Widgets (font size, bar size, custom display names, etc).
- More Model Input objects in the Dashboard including Radio Buttons, Listboxes, Trackers and Tables.
- Added picklist options for starting and stopping animations.
- Added picklist option to display labels on FlowItems in the 3D view.
- Moved the User Manual into FlexSim as a dockable window.
- Redesigned Robot GUI.
- Animation variables can point to components in an animation for quick referencing, not just surrogates.
- Improved User Toolbar items for Dashboard, GlobalTables, TimeTables and MTBF/MTTR buttons to allow for opening the objects.
- Improvements to shadows, shaders and mesh drawing.
- Improvements to the SKP Reader.
- Removed Model Views utility and placed it in the Quick Properties window.
- Added a "Headlight" feature to light sources.
- Updated Excel Importer/Exporter to handle relative paths of workbooks to the model.
- Improved the Excel Import's excelreadstr() and Automatic Data Distinction to more accurately read in values from excel, including dates and times.
- Added double click to open colors panel to all color wells.
- Fixed model unit scaling with flowitems when creating new models.
- Fixed issues with copying network nodes in containers.
- Fixed a bug in the Excel Import on Reset not handling multiple workbooks.
- Fixed a bug in the timetables causing the duration passed in to picklists to sometimes be negative.
- Fixed issues with closing/reopening dashboard widgets.
- Fixed sizing issues with dashboard widgets.
- Fixed Rack shelf tilt.
- Fixed an issue with having TaskExecuters using navigator logic for offset travel.
- Updated Move into Highlighted object to move into the model if no object is highlighted.
- Fixed a bug with running flypaths using model run speed.
- Fixed issues with adding flowitems to User Libraries.
- Fixed sizing issues with the Crane.
- Fixed a problem where Global Variables were not being properly loaded using save/load state.
- Fixed an issue with saving views in Full Screen Mode.
- Fixes to the Conveyor's drawing.
- Fixed issues with the Shape Frame tab of the Quick Properties window.
- Fixed a bug with the Basic Conveyor causing flowitems to not always exit when they're supposed to.
- Removed FlexSim Chart and save full history.

Backwards Compatibility Note: the following changes may slightly change the way updated models behave.

- The order in which MTBF/MTTR triggers is fired has been changed to reflect the documentation. Now the down/up trigger fires AFTER the down/up function. This may cause problems in models that depend on the down/up triggers firing before the down/up functions.
- Excel Multi Table Import using Automatic Data Distinction will import empty cells as string data, rather than as the number 0.

FlexSim 7.0.6

- Fixed an issue in the installer regarding Optquest dll registration.

- Added support for compiling with Visual Studio and Visual Studio Express 2013.
- Fixed the duration passed into the TimeTable down function when the last row is combined with the first row.
- Fixed a bug with time tables when the model start time is on a Sunday.
- Changed the Event List to save its filters when the model is saved.
- Fixed an issue with global variables when saving or loading state files.
- Fixed a bug in Display Date and Time pick option.
- Fixed a bug with exporting bundle string data.
- Fixed an exception when documenting user commands with improperly formatted descriptions.
- Some fixes to the query() command.

FlexSim 7.0.4

- Fixed a bug with the refresh rate overlay showing for educational users.
- Fixed an issue with library installcomponents not getting installed.
- Fixed a bug with the model start time getting reset incorrectly.
- Fixed issues with Labels not showing up in Dashboard Properties window.
- Fixed issues with saving open properties windows and top level windows.
- Refactored TimeTables to minimize the number of events created.
- Fixed issues with recursive debugging hiding the debug toolbar.

FlexSim 7.0.2

- Fixed a bug with the Pick Operator with Animation picklist.
- Added scroll bar to Labels page Tree View.
- Fixed bug in Processor that showed item conveying when stopped using STATE_BLOCKED.
- Added code to read texture repeats/offsets into the assimp importer so that .ac file textures render more accurately.
- Fixed some import and display issues with SKP files.
- Fixed a bug that was occasionally causing WebKit to crash.
- Fixed a bug in the Presentation Builder that was causing the first flypoint to jump beyond the second flypoint.
- Fixed issues with the TimeTable repeating daily.
- Fixed parqty() issue on nodefunction, user commands, function_s, etc.
- Fixed bug in Dashboard to display current content of Fluid Objects.
- OnCreate now gets fired for all objects inside a container when the container is copied.
- Fixed an exception in the Startup Page's OnPreLoad when there was an invalid recent models path.
- Updated Experimenter PFM graph to draw the box plot on top of replication points.
- Made it so foreign languages' dashboard statistic names will be properly associated with the visible name that they're dragged from.
- Fixed bug with changing the model start time and it not being reflected in the model stop time.
- Fixed reset exceptions on presentation slide.
- Fixed a bug that crashes FlexSim when you call startanimation with a rank that doesn't exist.
- Fixed bug causing FlexSim to crash when copying NetworkNodes.
- Fixed round() to work properly with negative numbers and large numbers.
- Fixed triangular distribution from dividing by 0.
- Fixed issue with debugging on a script window script when the first line is commented.
- Other various bug fixes from the development list.

FlexSim 7

- Novo manual do usuário com uma estrutura mais fácil de leitura.
- Versão 64-bit (faz com que o FlexSim use mais memória RAM).
- Revisão da interface de janelas para usar uma estrutura ancorada.
- Criada uma janela rápida de propriedades que é sensível ao contexto. A janela exibirá as propriedades mais utilizadas com base na seleção atual ou na janela do documento ativo.
- Encontrar/Substituir no Tree está agora integrado com a janela de propriedades rápidas e tem suporte para buscas case-insensitive assim como para buscas por nome dos nodes.
- Melhorias no Library Icon Grid que incluem filtros, grupos configuráveis e modo de edição.
- Library Icon Grid é context sensitive e altera sua exibição baseado na seleção atual ou na janela do documento ativo.
- Adicionado um botão de amostragem/dosador que é colocado em todo o software para permitir aos usuários provarem imagens, media 3D, objetos, nodes, números, strings e cores. Este recurso permite eliminar algumas necessidades de escrever códigos.
- Página de downloads que permite o recurso de realizar download e instalar módulos, formas em 3D, imagens e modelos.
- Adicionado uma ferramenta de medida/conversão.
- Melhorias no console script, permitindo aos scripts serem salvos ambos em modelos individuais e para o ambiente do usuário. Você pode agora debugar seu código de console script.
- Melhorias na interface do Presentation Builder.
- Melhorias na interface do Flowitem Bin, incluindo modos de embalagem para os flowitems container, visíveis e editáveis. As formas dos flowitem podem ser alterados através do método de arrastar e soltar diretamente da área de bibliotecas.
- Melhorias no emprego de shape frames nos objetos FR e Flowitems.
- Adicionado uma No Selecte flag para todos os objetos.
- Adicionado a exportação multi-tabela do Excel e reformulado a interface do Excel para combinar com o MTEI. O novo MTEI inclui uma opção para automaticamente reimportar tabelas no on reset.
- Melhorias no Animation Creator, incluindo animações dinâmicas usando variáveis, mais detalhes na edição de keyframes e nas triggers keyframe. Formas em 3D podem ser adicionadas para uma animação através do método de arrastar e soltar diretamente da área de bibliotecas.
- Criamos uma interface global para configurar a data/hora de início da simulação, linkado com as Time Tables. Uma data/horário de parada pode ser especificada.
- A janela Time Table foi melhorada. Uma programação diária ou semanal pode agora ser importada através do MTEI.
- Adicionado e atualizado diversos popups picklist removendo todas as opções de picklist baseado em texto.
- Melhorias no editor de código e em outras áreas onde a lógica é definida através de construções por arraste na grade de bibliotecas e nos botões de amostragem através de popups e picklist widgets para automaticamente adicionar códigos, etc. Os comandos do FlexSim também mostram certas opções quando digitados no editor de códigos.
- Os campos dos picklists e muitos campos do popup tem códigos selecionáveis e que se auto completam.
- Adicionado algumas implementações de Flexscript para as expressões lambda.
- Melhor debugger que permite você acessar o tree e outras áreas do FlexSim enquanto no modo de debug. Passando o mouse sobre variáveis durante o modo debug irá mostrar o seu valor atual.
- Melhorias no painel de desenvolvimento do GUI.
- Adicionado uma nova forma de construção de dashboard que irá substituir muitos passos no desenvolvimento dos GUI. Usuários podem agora realizar input do modelo através dos dashboards ao invés de simplesmente usar o construtor do GUI. Múltiplos dashboards podem ser criados.
- Agora você pode escolher qual navegador a TE está conectado através da sua página de propriedades

- Novas teclas/atalhos. Ctrl+K e Ctrl+L para redimensionar os objetos em 5%. Ctrl+W para fechar a janela do documento ativo ou a janela ativa flutuante. Atualização da Ctrl+Tab e Ctrl+Shift+Tab para movimentar-se entre as guias da janela do documento ou as flutuantes ativas.
- Adicionado um objeto FluidConveyor para a biblioteca padrão de fluídos.
- Pode agora ser visualizado um evento do objeto ao clicar com o botão direito do mouse no modo de visualização 3D e selecionar View/View Object Events.
- Melhoria completa no OptQuest (inclui suporte multi-core e integração com o experimenter).
- Melhor suporte na importação de formatos em 3D. FlexSim agora suporta os seguintes formatos: *.wrl; *.3ds; *.dxf; *.stl; *.skp; *.dae; *.obj; *.ac; *.x; *.ase; *.ply; *.ms3d; *.cob;*.md5mesh; *.irr; *.irrmesh; *.ter; *.lwo; *.csm; *.scn; *.q3o; *.q3s; *.raw; *.off; *.mdl; *.hmp; *.scn; *.xgl; *.zgl; *.lwo; *.lvs; *.blend
- Foi adicionada uma nova classe para desenho em OpenGL.
- Renderização em 3D estereográfica (requer estação de trabalho Quadro ou cartão FireGL para renderização frame-sequencial).
- Compatibilidade gráfica aprimorada com placas integradas Intel.
- Renderização em 3D melhorado, incluindo renderização de sombras, reflexos especulares em objetos 3DS, mapas de relevo, mapas de paralax, etc.
- Desenvolvimento do módulo SDK, incluindo:
 - Adicionado tipo de dado, que é uma classe abaixo para a agregação rápida e eficiente em memória de dados e para melhor codificação do módulo orientado a objeto, com um mecanismo simples para economizar na árvore.
 - Atualizado o wizards do visual studio que funciona com o VS 2012
 - Um modulo de tutorial
 - Mais documentação.

Nota sobre compatibilidade com versões anteriores: as seguintes mudanças podem alterar a forma como os modelos atualizados se comportam.

- Removido as formas 2D dos objetos.
- Removido a visualização planar.
- Muitos comandos OpenGL foram descontinuados. O mecanismo de atualização do modelo tenta substituir todos os usados antigamente. Geralmente isto deve funcionar, mas em alguns casos pode não acontecer. Note que glBegin(), glEnd(), glVertex(), glNormal(), glTexCoord() estão totalmente obsoleto e eventualmente não irão funcionar. Daqui em diante você deve usar o mesh api.
- O comando spaceobject() tornou-se obsoleto e não funciona mais. Ao realizar a otimização da ferramenta de gráficos, descobrimos a melhoria de 5-20% na taxa de atualização, simplesmente removendo a funcionalidade que está exclusivamente lá para fazer o comando spaceobject() funcionar. Então, nós depreciamos este comando. Ele ainda está na lista de comando para os modelos que serão atualizados, mas os modelos atualizados que utilizam terá formas estranhas aparecendo no modelo.
- OnLoad não é mais enviado para estrutura quando um projeto / tree é carregado. Se você tem objetos personalizados que dependem deste evento, você vai precisar usar algum outro mecanismo para disparar a lógica quando os modelos são carregados, ou seja, através da OnModelOpen, ou através OnModelOpen das bibliotecas do modelo.
- A funcionalidade Kinematics mudou de modo que, Kinematics padrão será removido automaticamente conforme passar os seus tempos finais individuais. Note que este comando exigia uma reestruturação dos dados Kinemáticos, por isso, se os seus modelos não o executarem ao resetar o modelo de qualquer maneira, você precisará reinicializar o kinematics nos modelos que são atualizados a partir de versões mais antigas.
- Nas versões anteriores, getdatastat() foi documentado incorretamente para o parâmetro p2 (graus de liberdade). Estava na verdade interpretando p2 como o número de amostras em um conjunto, não graus de liberdade. Nós corrigimos isso simplesmente não usando os parâmetros e inferindo o número de amostras/graus de liberdade de outros parâmetros. Isto significa que se você usou este comando anteriormente, irá retornar diferentes/melhores resultados nesta versão. Também alteramos a forma

como o intervalo de confiança contabiliza em percentual de forma ao resultado ficar mais seguro. Novamente, isto afeta os valores que foram retornados nas versões anteriores vs esta versão. Veja a documentação do comando `getdatastat()` para mais informações.

- Nós trocamos o nome da classe `FlexsimObject` para `FlexSimObject` in-line com nosso esquema de nomeação seguindo adiante. Nós implementamos um script atualizado que substitui todas instâncias do "FlexsimObject" com "FlexSimObject" nos modelos atualizados. Isto significa que se certas coisas em seu modelo são dependentes dos nome "FlexsimObject" (Não sabemos o que pode ser, exceto algum código dll que usa o nome `FlexsimObject`) pode haver problemas com a atualização.
- O importador 3ds importer está traduzindo alguns arquivos (chamados como robot clamps) de forma uma forma que é diferente dos nossos antigos importadores 3ds. É possível que outros arquivos 3ds pode precisar de seus índices ajustados manualmente após a atualização.
- Corrigimos o limite de recebimento por evento de 50 flowitem no queue. Isso vai quebrar modelos antigos se o seu modelo é dependente deste bug.
- Corrigimos um bug na desaceleração sobre um network, quando um task executer está bloqueando o espaço e é fornecido duas tarefas de viagem em uma fila (isto pode alterar os modelos mais antigos)
- Alguns atributos foram removidos, então se você usa estes atributos em seu modelo, então você precisará atualizar seu modelo adequadamente:
 - Atributos Removidos: `assertshape`, `asserttexture`, `billboard`, `distcutoff`, `events`, `instances`, `OnCaptured`, `OnCollision`, `OnInterrupted`, `state_graph`, `state_histo`, `state_percent`, `stats_contenthisto`, `stats_customgraphs`, `stats_throughputgraph`, `stats_throughputgraphmaxpoints`, `stats_throughputhisto`, `tables`, `textureaxis_s`, `textureaxis_t`, `travelstarttime`, `traveldirection`, `travelendtime`, `travelttimealpha`, `travelttimebeta`, `travelvpeak`, `traveldistance`, `travelstartx`, `travelstarty`, `travelstartz`, `travelendx`, `travelendy`, `travelendz`, `travelvmax`, `travelacc`, `traveldec`
 - Atributos de Desenho Removidos: `ignoredraw`, `ignorezbuffer`, `nochildrotate`, `nochildscale`, `noondraw`, `nopredraw`, `shapetype`, `reflective`, `luminous`
 - Comandos Removidos: `travelto`, `traveltoupdate`, `ntravelto`, `ntraveltoupdate`

FlexSim 6

- Melhorias no recurso do experimenter para melhor utilização do processamento de cenários.
- Adicionado uma nova interface para o GUI – Graphical User Interface
- Adicionado um novo Dashboard window com gráficos e informações estatísticas.
- Reprojeto a interface do Experimenter para integrar com novas estatísticas dos objetos
- Foi adicionado a opção de adicionar uma nova variável no experimento "number of objects in group" e "number of task executers."
- Desenvolvido opção de acesso via web: Abrir, configurar, rodar e visualizar modelos pela internet (usando um web browser ou smartphones)
- Foi adicionado uma nova opção de importer arquivos AutoCAD dwg.
- Adicionado janela para definir e converter unidades. Quando você constrói um novo modelo, uma tela irá aparecer perguntando qual unidade você irá utilizar durante o desenvolvimento de seu modelo. Este recurso está presente no menu Tools do modelo.
- Foi modificado os recursos do picklist e trigger para ficarem mais fácil de usar.
- Foi criado um novo node datatype (`DATATYPE_BUNDLE`) para armazenar uma maior quantidade de informações de forma eficiente (consulte a documentação de pacotes de comando).
- Foi implementado um compactador de media (3D formatos e bitmpas) dentro do arquivo do modelo para que você possa somente distribuir um arquivo ao invés de todo diretório de arquivos.
- Adicionado uma janela de documentação do comando embutido que pode ser aberto e destacado, pressionando F1 na janela de código ou no Tree Node.
- Adicionado uma rotina Flexscript para o depurador
- Adicionado a função por etapas nos comandos Flexscript do usuário e `nodefunctions` no depurador.
- Adicionado uma interface para construção lógica no Flexscript sem escrever código. (Você pode alterar o editor padrão de volta para Código desmarcando indo em "File > Global Preferences > Environment > Use Logic Builder by Default".)

- Melhorado a flexibilidade do pull logic e foi adicionado um novo comando pullitem() para uso na trigger "Pull Strategy".
- Melhorado os templates para permitir abertura de janelas popup do gui em opções de picklist.
- Foi adicionado um mecanismo para múltiplos perfis de estado dos objetos.
- As tabelas foram alteradas para sempre mostrar os cabeçalhos apesar de rolagem.
- Foi corrigido um bug que estava causando problemas na operação de salvar quando a memória do computador estava ocupada.
- Alternada a opção de large-address-aware para que o Flexsim possa alocar mais memória.
- Foi adicionado os comandos floor() e ceil().
- Adicionado um botão atualizar os nomes dos nodes e subnodes para pesquisa rápida.
- Adicionado uma opção de menu do botão direito do mouse em Labels de modo que os valores das labels são definidas em reset. Esta opção é marcada por padrão.
- Várias outras correções foram realizadas no developments list.

Nota sobre compatibilidade com versões anteriores: as seguintes mudanças podem alterar a forma como os modelos atualizados se comportam.

- Corrigido um bug crítico no cálculo do curved network length (configuração da distância entre os NNs).
- Corrigidos alguns pontos do gettenetnode() e distancetotavel() nos TEs (transportes) que usam a opção "do not travel offsets and block network space" nas propriedades.
- Foi feita uma alteração em distancetotavel() para informar em "back-to-node" a distância do centro do objeto ao invés da localização do objeto.
- Alterado a forma de configuração para produção puxada de forma a não sobrepor a produção empurrada. Ambos, não devem estar selecionadas para transferir um item do modelo.
- Foi alterado o receiveitem() para que não se comportasse com se estive sendo puxado
- Nas propriedades dos Rack's o comando OnEntry em trigger foi alterado para executar antes do dwell time.
- Foi adicionado um novo gerador de números aleatórios para gerar sementes baseado na quantidade de replicação para o gerador de número aleatório existente. Envie-nos um e-mail se você precisar de um script para rodar simulações de um modelo específico com valores usados nas versões anteriores do FlexSim.
- Foi adicionada uma nova sobrecarga para o commando tonode() para lidar com direcionamento de memória de grande porte. Dlls precisarão ser recompilados com cabeçalhos atualizados para obter essa correção.
- Corrigido um bug onde um endspeed de 0 não estava dizendo corretamente ao executor de tarefas para continuar na velocidade máxima no final de uma tarefa de deslocamento ou viagem. O comportamento agora corresponde corretamente a documentação

FlexSim 5

- Novo recurso de fazer/desfazer ação como no Windows.
- Novas capacidades de debuggind, incluindo:
 1. Um depurador Flexscript em linha, passo-a-passo. Basta ir ao editor de código e clique na margem esquerda para definir um ponto de ruptura.
 2. Regras de sintaxe mais rígidas para Flexscript, incluindo parâmetro de cardinalidade e verificação de tipo.
 3. Um log de eventos que permite que você veja todos os eventos que dispararam no modelo, filtrar esses eventos, exportá-los para csv, etc

4. Uma lista de eventos que lhe dá uma visão da lista de eventos pendentes.
 5. Um perfil de código Flexscript que lhe permite ver o que a funcionalidade Flexscript está sendo chamado, quantas vezes e quanto tempo está sendo tomado
- Um novo criador de animação que permite você criar movimentos em sub-componentes dos objetos assim como customizar animações para os mesmos sub-componentes.
 - Nova opção de importar arquivos .skp 3D para utilizar arquivos do Google Sketchup e Google 3D Warehouse.
 - Melhoria na taxa de resolução em 3D.
 - Melhorado a compatibilidade com placas de vídeo ATI, NVIDIA.
 - Nova forma de licenciamento do software usando Flexnet.
 - Forma de compilar os processos está mais rápido e confiável, sem a necessidade da criação de uma nova instância dentro do FlexSim.
 - Corrigido alguns temas de compilação com o Windows Vista e Windows 7
 - Mais detalhado, amigável e mais fácil de configurar ficou o relatório do Experimenter, com histograma adicional e gráficos de correlação.
 - Suporta arquivos .png, ico e gif para dar aparência sobre os objetos 3D, permitindo transparência sem a necessidade de requisição dos arquivos .tmp e .tpg.
 - Novo editor de códigos all-in-one com botão de aplicar e opção de tabulamento.
 - Melhorado a aparência na bibliotecas de objetos padrão.
 - Novo robô de 6-eixos (Obrigado Kris Geisberger).
 - Melhorado a interface para definição de layout dos Conveyor.
 - Melhorias no AVI Maker para o Windows Vista e Windows 7.
 - Novo e melhorado recursos para salvar arquivos .xml, permitindo que vários desenvolvedores trabalhem no mesmo modelo simultaneamente, bem como uma melhor integração com sistemas de gestão de controle de versões.
 - Várias melhorias para modeladores avançados e desenvolvedores.
 - Melhorias na opção de visualização em 3D usando o botão direito do mouse.
 - Forma mais intuitiva para manipular objetos na forma de visualização em 3D.
 - Usuários podem agora clicar sobre os objetos na grade do plano
 - Suporte melhorado para acessar painel de controle com o teclado.
 - Melhorado o recurso de encontrar e substituir em Tree View.
 - Melhorado a taxa de atualização nas tabelas de visualização.
 - Corrigido um problema com sugestões de preenchimento automático nos códigos
 - Corrigido um bug com renomeação de labels
 - Corrigido um problema com pontos do spline ser invisível se a visão em 3D tem um fundo preto
 - Vários novos comandos Flexscript para cadeia de análise.
 - Você pode agora utilizar Ctrl-Tab para alternar entre janelas dentro Flexsim
 - Mais recursos / comandos para personalizar a lógica do objeto TrafficControl.
 - Visualização em 3D do Flexsim agora tem um forma de destaque para que você possa ver o objeto que está sob o mouse.
 - Tratamento de exceção agora está de forma descritiva.
 - Adicionado os comandos TASKTYPE_STARTANIMATION, TASKTYPE_STOPANIMATION, e TASKTYPE_FREEOPERATORS.
 - Animações da visualização em 3D continua enquanto coloca-se na visão com zoom.

Iniciando os estudos

1. Iniciando o entendimento
2. Interagindo com o FlexSim
3. Interagindo com o Teclado
4. Terminologia
5. Primeiro Modelo
6. Ativação das Licenças

Iniciando simulação com o FlexSim

O que é o FlexSim?

Flexsim é uma poderosa ferramenta de análise que ajuda os engenheiros e profissionais de planejamento a tomar decisões inteligentes no projeto e operação de um sistema. Com o Flexsim, você pode construir um modelo computacional em 3D representando um sistema real, em seguida, estudar esse sistema em um período de tempo reduzido ou por um custo menor do que realizando experimentos e testes em um sistema real.

Como uma ferramenta de análise "e se", Flexsim fornece um feedback quantitativo de uma série de propostas de soluções para ajudá-lo rapidamente sobre a solução ideal para o seu processo. Com animação gráfica realista do Flexsim e relatórios de desempenho extensos, é possível identificar problemas e avaliar soluções alternativas em um curto espaço de tempo. Usando Flexsim para modelar um sistema antes de ser construído, ou para testar políticas operacionais antes de serem realmente implementadas, você vai evitar muitas das armadilhas que são frequentemente encontrados na partida de um novo sistema. Melhorias, que teve anteriormente, que levava meses ou anos de experimentação, de tentativa e erros, pode agora ser alcançado em questão de dias e horas usando Flexsim.

Modelagem

Em termos técnicos, o FlexSim é classificado como um software de simulação de eventos discretos. Isto significa que é utilizado para modelar sistemas que mudam de estado em pontos discretos no tempo como um resultado de eventos específicos. Estados comuns pode ser classificados tais como ocioso, ocupado, bloqueado ou para baixo, e alguns exemplos de eventos seria a chegada de pedidos de clientes, movimentação de produtos e avarias de máquinas. Os itens que estão sendo processados em um modelo de simulação de eventos discretos muitas vezes são produtos físicos, mas eles também podem ser clientes, documentos, desenhos, tarefas, telefonemas, mensagens eletrônicas, etc Esses itens são processados através de uma série de processamento, filas e etapas de transporte no que é chamado de fluxo de processo. Cada passo do processo pode exigir um ou mais recursos, tais como uma máquina, um transportador, um operador de um veículo ou de uma ferramenta de algum tipo. Alguns desses recursos são estacionários e alguns são móveis, alguns recursos são dedicados a uma tarefa específica e outros devem ser compartilhados entre várias tarefas.

Flexsim é uma ferramenta versátil, que tem sido usado para modelar uma variedade de sistemas em diversas indústrias. Flexsim é utilizado com sucesso por pequenas e grandes empresas. Cerca de metade de todas as empresas da Fortune 500 são clientes Flexsim, incluindo nomes tão notáveis como a General Mills, Daimler Chrysler, Northrop Grumman, Discover Card, DHL, Bechtel, Bose, Michelin, FedEx, Seagate Technologies, Pratt & Whitney e TRW.

Existem três problemas básicos que podem ser solucionados com o FlexSim:

1. Problemas com serviços - a necessidade de atender os clientes e seus pedidos com o mais alto nível de satisfação e com o menor custo possível.
2. Problemas de fabricação - a necessidade de fabricar o produto certo na hora certa com o menor custo possível.
3. Problemas logísticos - a necessidade de levar o produto certo para o lugar certo na hora certa com o menor custo possível.

Exemplos de como o FlexSim é usado

Para dar-lhes uma idéia para possíveis projetos, o FlexSim tem sido utilizado para:

- Melhorar a utilização de equipamentos
- Reduzir tempos de esperas e tamanho de filas
- Alocar os recursos eficientemente
- Eliminar problemas de falta de estoque
- Minimizar efeitos negativos de quebras de máquinas
- Minimizar efeitos negativos de rejeição e desperdícios
- Estudar alternativas e ideias de investimentos
- Determinar a produtividade
- Estudar planos de redução de custos
- Estabelecer tamanhos de bateladas ótimos e sequenciamento de produção
- Resolver algumas questões de manuseio de materiais
- Estudar os efeitos de tempos de setup e trocas de ferramentas
- Otimização de priorização e lógica de despacho para produtos e serviços
- Treinar operadores sobre o comportamento geral do sistema e performance relacionado aos trabalhos
- Demonstrar novas ferramentas de design e capacidades
- Gerenciar as decisões de operação diariamente

Flexsim tem sido utilizada com sucesso em ambos os estudos de fabricação de um novo sistema (produtos) e na gestão de movimentação de produtos durante seu processamento diário. Flexsim também tem sido utilizado para fins de treinamento e educação. Um estudo com modelagem utilizando o Flexsim pode fornecer insights sobre as dependências complexas e dinâmicas de um sistema de vida real. Ele pode ajudar os operadores na gestão de como um sistema funciona, mas saber o que acontece quando os procedimentos alternativos são implementadas. Flexsim tem sido usado para construir modelos interativos que podem ser manipuladas enquanto o modelo está sendo executado, a fim de ajudar a ensinar e demonstrar a causa e os efeitos inerentes ao sistema de gestão.

Visualização

Flexsim é uma tecnologia altamente revolucionário que pode ser usado por profissionais com visão de futuro para elevar a imagem da sua empresa e declarar para o mundo todo que seu fabricante orgulha-se de como ele funciona.

É surpreendente o quão eficaz um modelo de simulação de animação pode ser para chamar a atenção da administração e influenciar sua maneira de pensar. A animação exibida durante uma simulação fornece uma ajuda visual excelente para demonstrar como o sistema realmente funciona.

Interagindo com o FlexSim

Tópicos

- Criando um objeto
- Nomeando um objeto
- Editando objetos
- Conectando objetos
- Navegação e visualização

Criando um Objeto

Objetos podem ser criados ao entrar no modo **Create Objects**, ou arrastando e soltando:

1) Entre no modo **Create Objects** clicando e soltando o mouse sobre um objeto na janela de bibliotecas. Clique novamente na visualização em 3D para criar um objeto.

2) Entre no modo de criação de objetos clicando no botão  localizado em main toolbar. Então clique no objeto que você deseja criar e que encontra-se na biblioteca (**Library**). Clique novamente em model view (onde monta-se o modelo) e escolha o local onde você quer que o objeto seja criado.

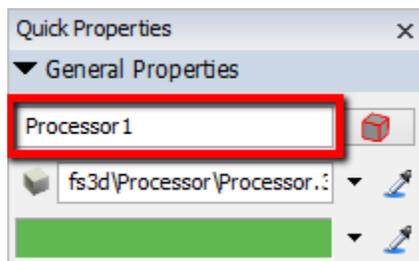
3) Clique e mantenha o botão esquerdo mouse pressionado sobre um objeto em **Library**, então arraste o para a posição que você quer colocar na superfície de simulação e solte o botão do mouse.

Nomeando um Objeto

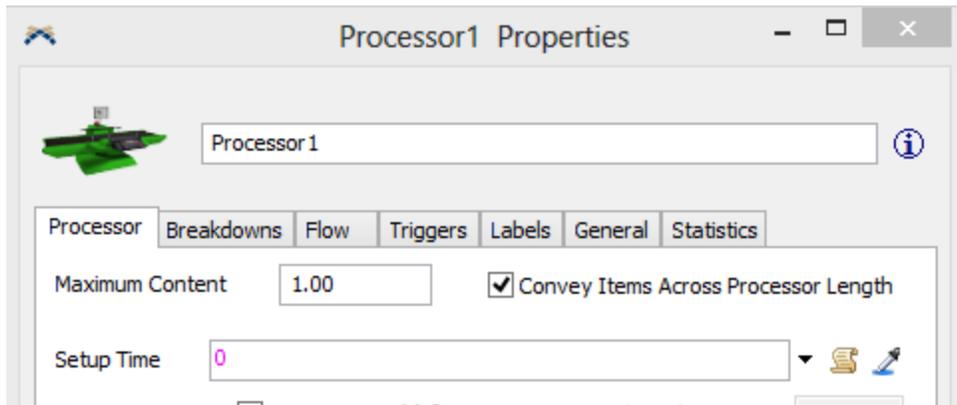
Conforme os objetos são criados, são dados nomes padrão para cada um como Source#, onde # é a numeração de cada objeto criado desde que o FlexSim foi aberto.

Existem dois métodos para renomear um objeto:

1) Clique sobre um objeto na sua forma 3D para mostrar as propriedades de um objeto na janela de propriedades do lado direito da tela (conhecida como Quick Properties Window). Então edite o nome na parte superior da janela de propriedades (Quick Properties).



2) Para nomear um objeto, clique duas vezes sobre o objeto para abrir a tela de propriedades. Então edite seus nomes na parte superior da janela e clique em **Apply** ou **OK**.

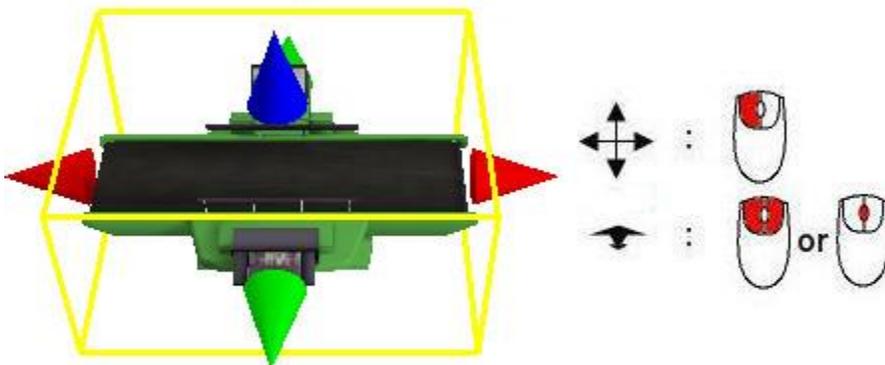


Editando Objetos

Movimentando Objetos – Para mover um objeto em torno do modelo, clique nele com o botão esquerdo do mouse e arraste para a posição desejada. Você também pode mover o objeto para cima e para baixo na direção z usando a roda do mouse ou segurando ambos os botões esquerdo e direito do mouse no objeto e arraste o mouse para frente e para trás

Dimensionamento e Rotação – Para editar o tamanho e rotacionar o objeto, selecione a opção do menu principal em Edit> Resize and Rotate Objects. Quando você clicar no objeto, você deve ver três setas coloridas ao longo de cada eixo do objeto. Para redimensionar o objeto, à esquerda, clique no eixo que deseja redimensionar, e arraste o mouse para cima ou para baixo. Para editar a rotação do objeto, clique na seta correspondente ao eixo que deseja girar ao redor, e arraste o mouse para frente ou para trás. Você pode querer aumentar ou diminuir um objeto em 5% mantendo a tecla Ctrl pressionada e segurando as teclas K ou L.

Nota: Você pode querer alternar entre redimensionamento e rotação dos objetos para estar ativado ou desativado, ao selecionar a opção do menu principal em **Edit > Resize and Rotate Objects**.



Propriedades – Todos os objetos do Flexsim tem um número de páginas ou Guias onde estão as variáveis de informação que podem ser alteradas pelo modelador com base nos requisitos atuais do modelo.

Destruindo ou deletando objetos – Para deletar um objeto, simplesmente selecione-o e pressione a tecla delete.

Conectando Objetos

Portas são criadas e conectadas de uma das duas maneiras possíveis:

1) Ao clicar em um objeto e arrastando em direção a outro objeto, mantendo pressionada diferentes letras do teclado. Se a letra "A" é pressionada enquanto clicar e arrastar, uma porta de saída será criado no

primeiro objeto e uma porta de entrada será criada no segundo objeto. Estas duas novas portas serão, então, conectadas automaticamente. Mantendo pressionada a tecla "S" vai criar uma porta central, em ambos os objetos e ligar as duas novas portas. As ligações são quebradas e portas eliminadas, mantendo-se o "Q" selecionado para as portas de entrada e saída e a tecla "W" para as portas centrais. A tabela a seguir mostra as letras de teclado usados para fazer e quebrar os dois tipos de ligações de portas.

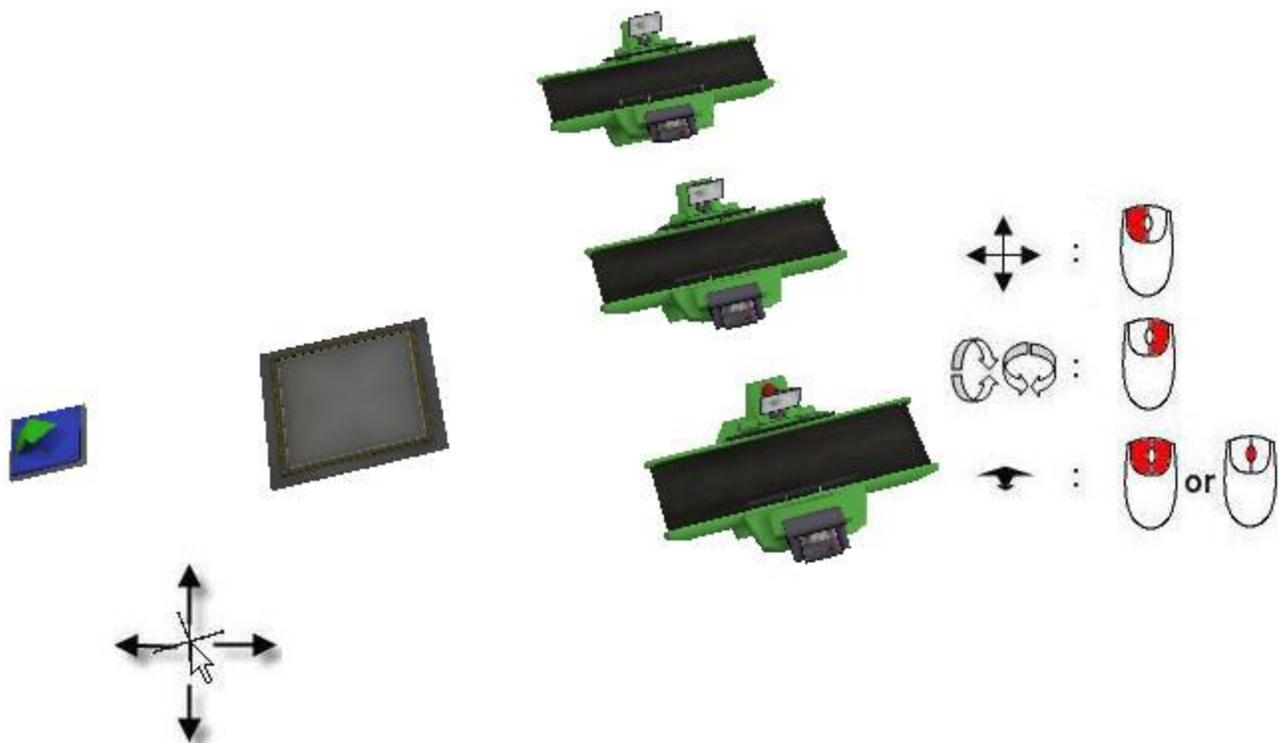
	Portas de Saída-Entrada (Output – Input)	Porta Central
Desconectar	Q	W
Conectar	A	S

2) Ao entrar no modo de conexão mantendo a tecla A pressionado . Uma vez no modo de conexão, há um par de maneiras de fazer uma conexão entre dois objetos. Você também pode clicar em um objeto, em seguida, clique em outro objeto, ou você pode clicar e arrastar de um objeto para o próximo como no primeiro método. De qualquer forma, tenha em mente que a direção do fluxo de uma conexão é dependente da ordem em que você faz a conexão. Fluxo vai desde o primeiro objeto para o segundo objeto.

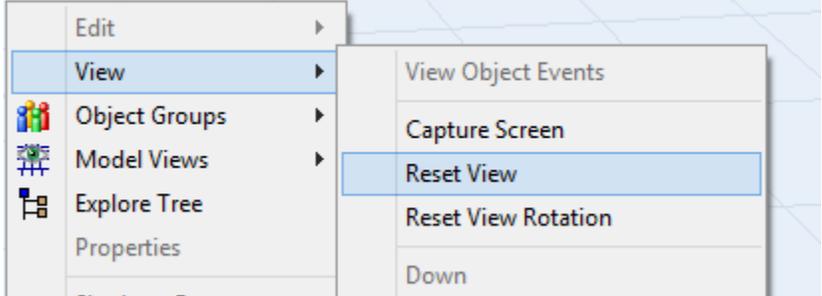
Aliás, as ligações podem ser quebradas, mantendo a tecla Q pressionado  e em seguida, clicar ou arrastar de um objeto para outro, da mesma maneira como quando realizando conexão entre eles. Conexões de portas centrais podem ser realizadas mantendo o botão S do teclado pressionado e em seguida, clicar ou arrastar de um objeto para outro. Para desfazer a conexão da porta central, basta manter o teclado em W pressionado e desfazer a conexão de acordo com o sentido que deve ser desfeito.

Navegação e visualização

Navegação Básica - Para mover o ponto de vista do modelo, clique em uma área vazia do ponto de vista com o botão esquerdo do mouse e arraste o mouse. Para girar o ponto de vista do modelo, clique em uma área em branco com o botão direito do mouse e arraste o mouse. Para ampliar ou reduzir, utilize a roda do mouse ou segure os botões direito e esquerdo do mouse pressionado e arraste o mouse.

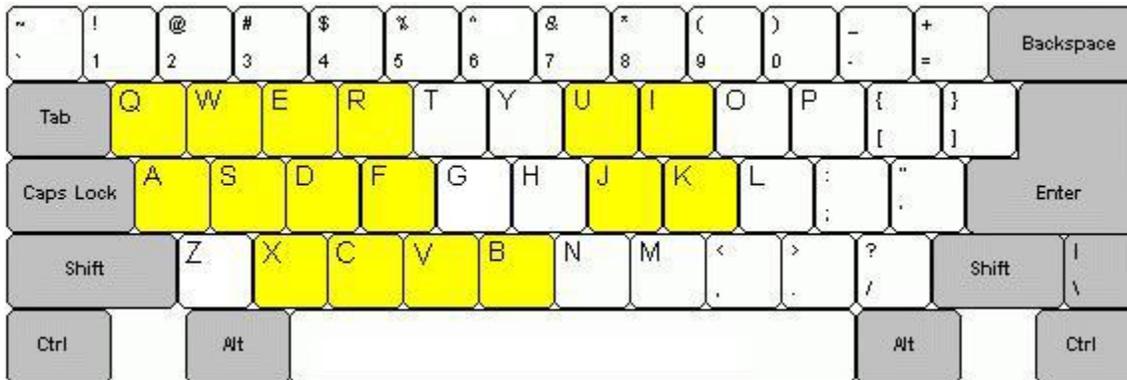


Reset View – Você pode resetar a visualização para o ponto de visualização padrão ao clicar com o botão direito do mouse no espaço da superfície de simulação vazio, e selecionar o menu em **View > Reset View**.



Interação com o Teclado

Quando você está trabalhando em ortographic ou perspective view, você vai usar várias teclas do teclado para construir, personalizar e obter informações a partir do modelo. A figura Guiaixo mostra o layout do teclado. Teclas que estão destacadas em amarelo têm significado ao interagir com Flexsim.



A, J: Conexões context sensitive

A tecla A é utilizada para conectar dois objetos, dependendo do tipo de objetos. Mantenha pressionada a tecla A, clique em um objeto, segure o botão do mouse pressionado e arraste para outro objeto, soltando o botão do mouse sobre o objeto. Normalmente, desta forma, liga-se as portas de saída de um objeto com as portas de entrada do outro objeto. Para NetworkNodes, no entanto, a tecla A, liga um NetworkNode a um TaskExecuters, como viajantes, ao FixedResources como gateways de viagem, e ao NetworkNodes como percurso de viagem. Você também pode usar a tecla J, se você é canhoto. Se você conectar dois objetos com a tecla A, e não ver qualquer mudança, primeiro verifique se as configurações de visualização não escondem conexões. Se ainda não houver mudança, esses objetos provavelmente, não foram conectados com a tecla A.

Q, U: Desconexão context sensitive

A tecla Q é utilizada para desconectar dois objetos, dependendo do tipo de objetos. Mantenha pressionada a tecla Q, clique em um objeto, segurando o botão do mouse pressionado e arraste para outro objeto, soltando o botão do mouse sobre o objeto. Geralmente isso desconecta as portas de saída de um objeto a partir das portas de entrada de outro objeto. Para NetworkNodes, no entanto, a tecla Q desconecta um NetworkNode do TaskExecuters como viajantes, do FixedResources como gateways de viagem, e define um caminho de viagem para "nenhuma conexão" (vermelho). Você também pode usar a tecla U, se você é canhoto.

S, K: Conexão de portal central

A tecla S é usada para conectar as portas centrais de dois objetos. Portas centrais são utilizadas para fins de referência, utilizando o comando centerobject (). Segure a tecla S, clique em um objeto, mantenha o botão do mouse pressionado e arraste para outro objeto, soltando o botão do mouse sobre o objeto. Você também pode usar a tecla K se você é canhoto.

W, I: Desconectar portal central

A tecla W é usada para desconectar portas centrais de dois objetos. Mantenha pressionada a tecla W, clique em um objeto, segurando o botão do mouse pressionado e arraste para outro objeto, soltando o botão do mouse sobre o objeto. Você também pode usar a tecla I, se você é canhoto.

D: Conexão context sensitive

A tecla D é o segundo teclado para conexão context sensitive. O NetworkNode e o TrafficControl usam este tipo de conexão.

E: Desconexão context sensitive

A tecla E é o segundo teclado para desconexão context sensitive. O NetworkNode e o Traffic Control usam esta conexão.

X: Alternar/clicar context sensitive

A tecla X é usado para alterar um objeto ou ver as informações sobre o objeto, dependendo do tipo de objeto. Segure a tecla X e clique no objeto. O NetworkNode irá mudar toda a rede através de diferentes modos de visualização. A tecla X também cria novos pontos de spline em um caminho. Racks também vai alternar entre diferentes modos de visualização. O conveyor irá reposicionar os conveyor a jusante para ficar alinhada com os conveyors que encontram-se no final da linha.

B: Alternar/clicar context sensitive

A tecla B é uma tecla adicional usada para alterar um objeto ou ver as informações sobre o objeto, dependendo do tipo de objeto. Segure a tecla B, e clique sobre o objeto. O NetworkNode irá alterar através de diferentes modos de visualização. O TrafficControl também usa a tecla B.

V: Visualizar conexões de porta de entrada/saída

A tecla V é usada para exibir as conexões de porta de entrada/saída de um objeto. Segure a tecla V e clique em um objeto, segurando a tecla V e o botão do mouse apertado. Se o botão do mouse é liberado em primeiro lugar, em seguida, a informação irá desaparecer, mas se a tecla V é solta primeiramente, a informação vai persistir.

C: Visualizar conexões de porta central

A tecla C é utilizado para visualizar as conexões de portas centrais de um objeto. Mantenha a tecla C apertada, e clique em um objeto, segurando a tecla V e o botão do mouse apertado. Se o botão do mouse é liberado em primeiro lugar, em seguida, a informação irá desaparecer, mas se a tecla C é solta primeiramente, a informação vai persistir.

F: Criar biblioteca de objetos

A tecla F é usada para criar rapidamente objetos de biblioteca. Selecione um objeto no ícone de rede biblioteca, clicando sobre ela. Em seguida, clique na vista orto / perspectiva, e pressione e segure a tecla F apertada. Em seguida, clique na vista orto no local que você gostaria de criar o objeto. O objeto será criado nessa posição.

R: Criar e conectar biblioteca de objetos

A tecla R funciona como a tecla F, exceto que ele também conecta objetos criados consecutivamente com uma conexão.

Teclas de Atalhos e dicas:

- Ctrl + Space** - Inicia e para de rodar o modelo de simulação.
- Ctrl + Left Arrow** – Reseta o modelo.
- Ctrl + Right Arrow** – Pula para o próximo evento do modelo.
- Ctrl + Up Arrow** – Aumenta a velocidade de simulação.
- Ctrl + Down Arrow** – Diminui a velocidade de simulação
- Ctrl + F** – Encontra o texto na visualização aberta.
- Ctrl + H** – Encontra e substitui o texto na visualização aberta.
- Ctrl + C** - Copia o objeto selecionado para o clipboard.
- Ctrl + X** - Recorta o objeto selecionado para o clipboard.
- Ctrl + V** - Cola o objeto para o clipboard.
- Ctrl + Tab** - Troca para a próxima guia da janela.
- Ctrl + Shift + Tab** - Retorna para a guia da janela anterior.
- Ctrl + L** - Aumenta a escala do objeto selecionado em 5%.
- Ctrl + K** - Diminui a escala do objeto selecionado em 5%.
- Ctrl + D** - Adiciona um keyframe para o presentation builder.
- Ctrl + W** - Fecha a janela do documento ativo.
- F1** - Abre o comando de ajuda e pesquisa para o item selecionado.

Tecla de Atalhos para Janela Tree

Os seguintes atalhos estão disponíveis para a janela Tree:

- Spacebar** – Insere um novo node logo Guiaixo do atual.
- Enter** – Insere um novo node(sub node) dentro do node selecionado.
- N** - Adiciona dados numéricos no node em destaque.
- T** - Adiciona dados como string (texto) no node em destaque.
- O** - Adiciona dados de objeto no node em destaque.
- P** - Adiciona pointer data no node em destaque.

Terminologia no FlexSim

Antes de iniciar o seu primeiro modelo, seria útil entender algumas terminologias básicas do software.

Objetos no FlexSim

Os objetos do Flexsim simulam diferentes tipos de recursos na simulação. Um exemplo é um objeto de fila, que atua como uma zona de armazenamento ou um buffer. A fila pode representar uma fila de pessoas, uma fila de processos ociosos em uma CPU, uma área de armazenamento no chão de uma fábrica, ou uma fila de chamadas em espera em um centro de atendimento ao cliente. Outro exemplo de um objeto no Flexsim é o objeto processor (equipamento), o qual simula um atraso ou tempo de processamento. Este objeto pode representar uma máquina em uma fábrica, um caixa de banco atendendo um cliente, um funcionário da triagem dos correios, etc

Os objetos do FlexSim são encontrados na Biblioteca padrão do software e fica do lado esquerdo ao abrir o software. Está organizado em biblioteca de eventos discretos e objetos de fluídos. Pode-se criar facilmente uma biblioteca padrão de equipamentos de sua empresa de forma fácil e rápida.

Flowitems

Flowitems são os objetos (produtos por exemplo) que se movem através de seu modelo, também conhecido como entidades. Flowitems pode representar peças, paletes, conjuntos, papel, embalagens, pessoas, telefonemas, pedidos ou qualquer coisa que se move através do processo que você está simulando. Flowitems podem ter processos realizados sobre eles e podem conduzidos através do modelo por recursos de manuseio de materiais. No Flexsim, flowitems são gerados pelo Source. Uma vez flowitems passaram pelo modelo, eles são enviados para um objeto de saída do modelo, o Sink.

Labels

Labels são strings (textos) ou números que são estocados dentro dos Flowitems e objetos. As Labels podem ser dinamicamente alterados através do curso de um fluxo de processo. As Labels podem ser úteis para armazenar informações como custo, tempo de processamento e outras informações. As labels podem ser acessadas através da janela de propriedades rápidas (quick properties) ou em suas guia Label na janela de propriedade dos objetos ou flowitems. Podem também ser chamados de atributos das entidades e objetos.

Itemtype

O itemtype é uma etiqueta (label) que é colocada sobre um flowitem que pode representar um número de código de barras, o tipo de produto, ou o número da peça. Flexsim está configurado para usar o itemtype como referência no roteamento dos flowitems

Portas

Cada objeto no Flexsim tem um número ilimitado de portas através do qual eles se comunicam com outros objetos. Existem três tipos de portas: portas de entrada, portas de saída, e central.

- **Portas de entrada e de saída (input/output ports):** Portas de entrada e de saída são usados no encaminhamento de flowitems. Por exemplo, um selecionador de pacotes de correio coloca o pacote sobre uma das várias esteiras transportadoras de acordo com o destino do pacote. Para simular isso no Flexsim, você iria ligar a porta de saída de um processor (neste caso, o selecionador de pacotes) com a porta de entrada várias esteiras transportadoras, ou seja, uma vez que o processador (selecionador de pacotes) tenha concluído o processamento do flowitem (ou pacote), ela envia para uma esteira transportador

Porta Central (Central Port): Portas centrais são usados para criar referências de um objeto para outro. Um uso comum para as portas centrais é fazer referência a objetos móveis, tais como operadores, empilhadeiras, guindastes e de recursos fixos, como máquinas, filas ou transportadores.

- [Clique aqui para aprender um pouco mais sobre portas.](#)

Primeiro Modelo

Descrição

Neste modelo, vamos olhar para o processo de fabricação de três tipos de produtos diferentes de uma fábrica. No nosso modelo de simulação, vamos associar a um valor itemtype para cada um dos três tipos de produtos. Estes três tipos de produtos chegam de forma intermitente de outra parte da fábrica. Há também três máquinas em nosso modelo. Cada máquina pode processar um tipo de produto específico. Uma vez que os produtos são processados em suas respectivas máquinas, os três tipos de produtos devem ser testados em uma única estação de teste compartilhado para correções. Se eles foram fabricadas corretamente, eles são enviados para uma outra parte da instalação, deixando o nosso modelo de simulação. Se fossem fabricados incorretamente, devem voltar ao início do modelo de simulação para ser reprocessados pelas respectivas máquinas. O objetivo da simulação é encontrar onde está o gargalo

É importante a quantidade de espaço do buffer antes da estação de teste?

Aplicando o Modelo para Diferentes Indústrias

Enquanto estivermos utilizando uma indústria para este exemplo, o mesmo modelo de simulação pode ser aplicado a outras aplicações semelhantes. Dê uma copiadora, por exemplo. Uma loja com copiadora possui três serviços principais: cópias preto e branco, cópias coloridas e impressões especiais. Durante o horário comercial, há três funcionários trabalhando. Um funcionário lida com os trabalhos de cópia em preto e branco, outro funcionário é responsável pela cópia a cores, e o terceiro cuida das impressões especiais. Há também uma caixa para cobrar os serviços finalizados. Cada cliente que entra na copiadora dá um trabalho para o funcionário responsável por cada uma de suas tarefas. À medida que cada trabalho é concluído, ele se deslocado para o caixa para finalizar a venda e entregar ao cliente a cópia ou impressão finalizada. No entanto, às vezes o cliente não está satisfeito com o trabalho que foi feito. Nesses casos, o trabalho deve ser dado de volta para o funcionário adequado a ser feito novamente. Esse cenário representa o mesmo modelo de simulação como o descrito acima para uma indústria processadora. Aqui, porém, você pode estar mais preocupado com a fila de clientes e com o tempo que passam esperando, pois o serviço lento pode ser muito caro para os negócios de uma copiadora.

Aqui temos outro exemplo de aplicação de um modelo de simulação, aplicado para a indústria de transporte. Caminhões de transporte comerciais que viajam sobre uma ponte do Canadá para os Estados Unidos devem passar por um recinto alfandegário, antes de serem autorizados a entrar no país. Cada motorista de caminhão deve primeiro obter a documentação necessária e, em seguida, passar por uma inspeção final do caminhão. Há três categorias gerais de caminhões. Cada categoria tem um tipo diferente de papelada para preencher e deve aplicar-se a um departamento diferente do recinto alfandegário. Uma vez que a papelada estiver ok, todas as categorias de veículos devem passar pelo mesmo processo de inspeção. Se eles falharem a inspeção, eles devem apresentar documentações adicionais, etc Mais uma vez, esta situação contém os elementos exatos de simulação parecidos com o exemplo da fábrica, apenas aplicados ao setor de transportes. Aqui, você pode estar interessado no tamanho de filas de caminhões ao longo e posterior a ponte. Se o tamanho for em quilômetros e, assim, estiver bloqueando o tráfego na cidade canadense vizinha, então você pode precisar de mudar a forma como o mecanismo funciona no posto alfandegário.

Construindo o Modelo

Se em algum momento você encontrar dificuldades durante a construção deste modelo, um tutorial totalmente funcional pode ser encontrada em <http://www.flexsim.com/tutorials>

Para verificar se o software Flexsim foi instalado corretamente, abra o aplicativo clicando duas vezes no ícone Flexsim em seu desktop. Uma vez carregado e aberto o software, você deve acessar o menu do Flexsim e barras de ferramentas, Biblioteca e a visualização ortográfica.

Etapas

- Etapa 1: Inicie FlexSim
- Etapa 2: Selecione as Unidades
- Etapa 3: Crie os Objetos
- Etapa 4: Conecte os Objetos
- Etapa 5: Defina o tempo entre chegadas (Inter-Arrival Time)
- Etapa 6: Atribua um tipo de item e cor (Itemtype and a Color)
- Etapa 7: Defina a capacidade máxima do primeiro buffer (Queue's Maximum Content)
- Etapa 8: Defina a lógica de envio do primeiro buffer (Queue1's Routing)
- Etapa 9: Defina os tempos de processamento
- Etapa 10: Defina a capacidade máxima do segundo buffer (Queue2's Maximum Content)
- Etapa 11: Defina o tempo de processamento para a estação de teste (Tester's Process Time)
- Etapa 12: Defina a lógica de envio da estação de teste.
- Etapa 13: Resete e rode o modelo.
- Realize experimentos com o modelo
- Aleatoriedade
- Crie os Dashboards
- Resultados

Etapa 1: Inicie o FlexSim

- Abra o FlexSim com um duplo clique no ícone do FlexSim, que deve estar presente em seu desktop. A tela inicial Guiaixo irá aparecer por padrão. Selecione "New Model".



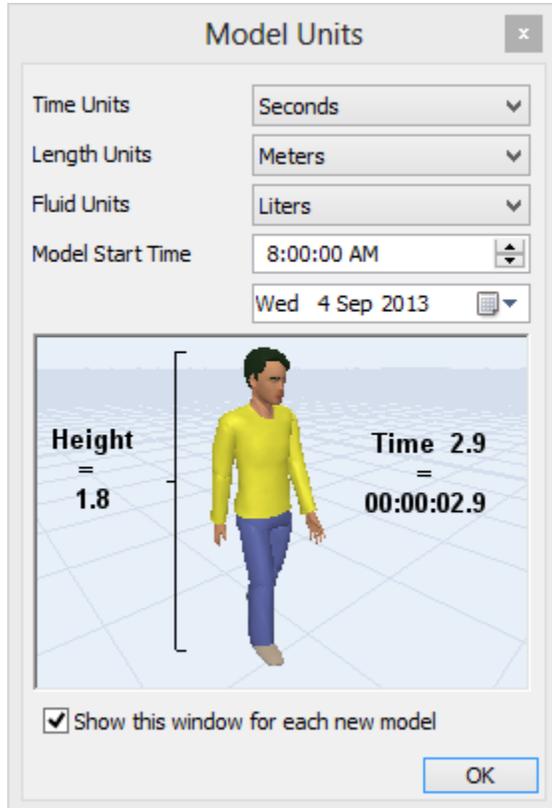
Passo 2: Selecione as unidades

O Flexsim permite ao usuário selecionar a unidade apropriada para o modelo a ser desenvolvido. Por padrão, a janela de unidades do modelo (**Model Units window**) irá aparecer toda vez que for construir um novo modelo. Você pode selecionar unidades para tempo, comprimento e fluidos. A unidade que você assumir será considerada em todo o modelo. O tempo de início do modelo (Model Start Time) pode ser

alterado após o modelo for criado, no entanto, as unidades de tempo, distância e de fluídos NÃO PODEM ser alterados.

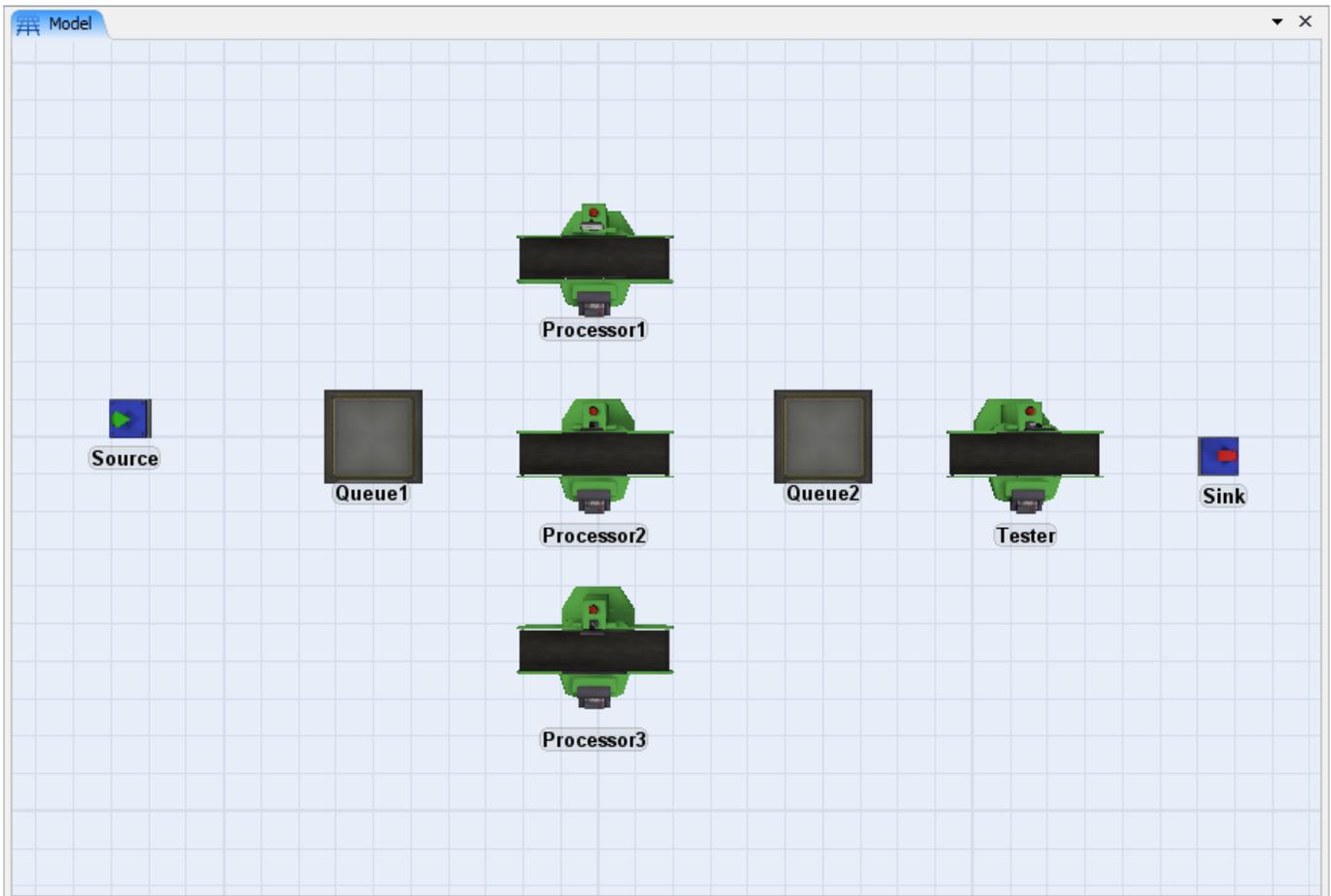
Para este modelo, iremos utilizar as seguintes unidades:

- **Unidade de tempo:** Segundos.
- **Unidade de comprimento:** Metros
- **Unidade de fluídos:** Litros.



Etapa 3: Crie os Objetos

- Crie um Source, dois Queues, quatro Processors, e um Sink no modelo. Nomeie e coloque conforme demonstrado abaixo. Os objetos podem ser criados simplesmente, arrastando da biblioteca de objetos para dentro do modo de visualização em 3D. Para nomear um objeto, dê um duplo clique sobre ele, e troque o nome na parte superior da janela de propriedades. Pressione Apply e OK.

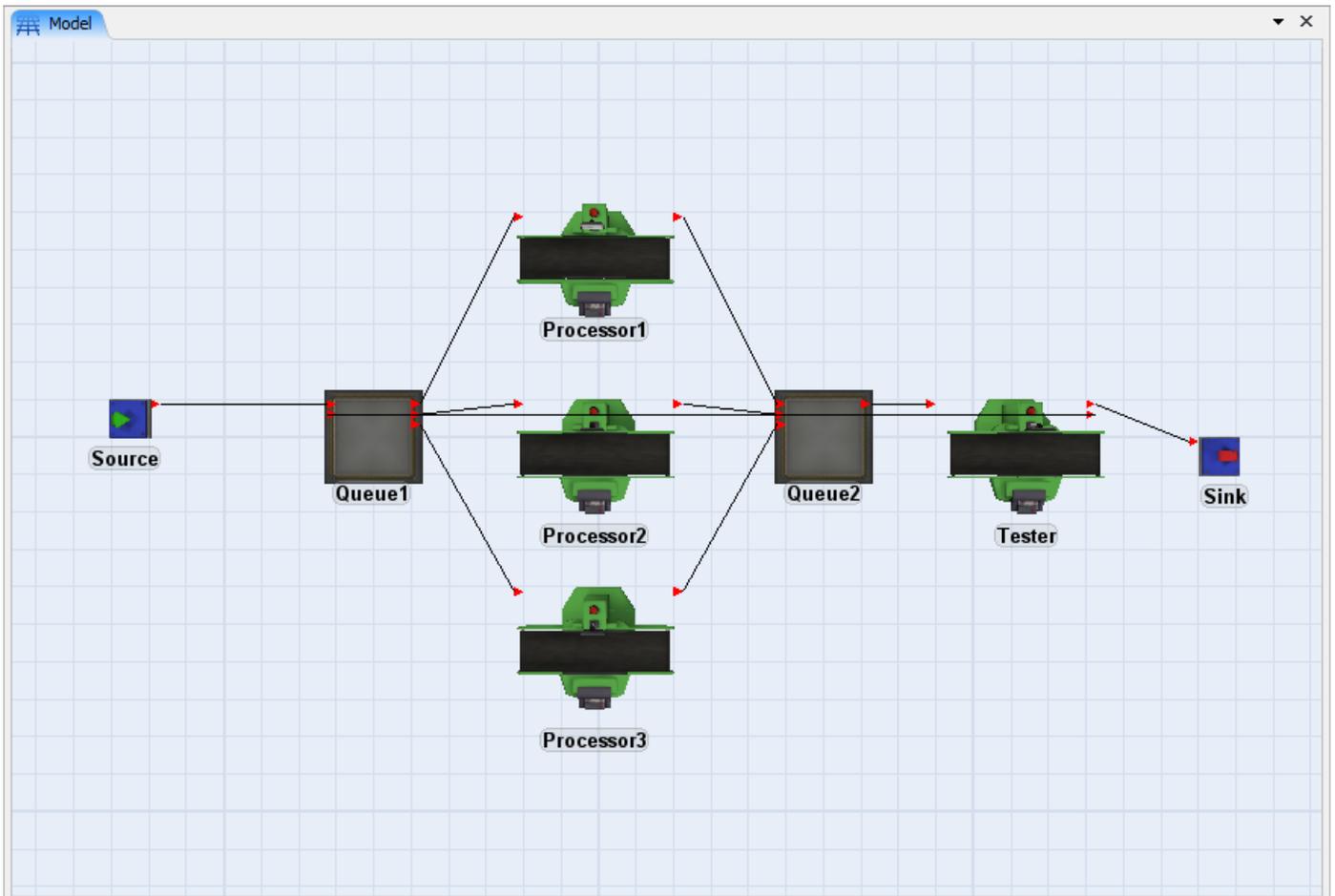


Etapa 4: Conecte os Objetos

Entre no modo de conexão simplesmente clicando em  ou simplesmente mantendo a tecla A de seu teclado pressionada. Uma vez no modo de conexão, existem dois meios de fazer a conexão entre os objetos. Você pode clicar em um objeto e depois em outro, ou você pode clicar e arrastar de um objeto para o outro. Em ambas as formas, mantenha em mente que a direção do fluxo é dependente da ordem no qual é realizado a conexão. O Fluxo vai do primeiro objeto para o segundo objeto na conexão. Além do mais, as conexões podem ser quebradas, clicando no botão  ou mantendo a tecla Q pressionada em seu teclado enquanto clica e arrasta de um objeto para o outro da mesma maneira quando você os conecta.

- Conecte *Source* com *Queue1*.
- Conecte *Queue1* com *Processor1*, *Processor2*, e *Processor3*.
- Conecte *Processor1*, *Processor2*, e *Processor3* com *Queue2*.
- Conecte *Queue2* com *Tester*.
- Conecte *Tester* com *Sink* e *Queue1*.

Observação importante: Os equipamentos nomeados como *Tester*, são na verdade os equipamentos ou estação de processamento (*Processors*) e todas as suas propriedades são alteradas dando um duplo clique sobre o objeto *Processor* ou destacando ele, onde você verá no painel do lado direito de sua tela, o *Quick Properties*, onde é possível alterar algumas propriedades básicas.



O próximo passo é alterar as propriedades dos diferentes objetos de forma como eles irão comportar-se como indicado na descrição do modelo. Vamos começar com o Source e trabalhar até chegarmos no Sink. Cada objeto tem sua própria janela de propriedades através do qual os dados e a lógica são adicionados ao modelo. Clicando duas vezes em um objeto acessa a janela de propriedades do objeto.

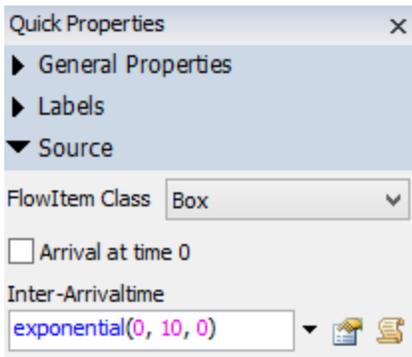
Para este modelo, queremos três tipos de produtos diferentes para entrar no sistema. Para fazer isso, para cada flowitem será atribuído um itemtype e um valor inteiro entre um e três usando uma distribuição uniforme (ver Flexsim Conceitos para obter mais informações sobre itemtypes). Isto será conseguido acessando a tela de propriedades do Source, na Guia trigger e clicando no sinal + em OnCreation.

Etapa 5: Defina o tempo entre chegadas (Inter-Arrival Time)

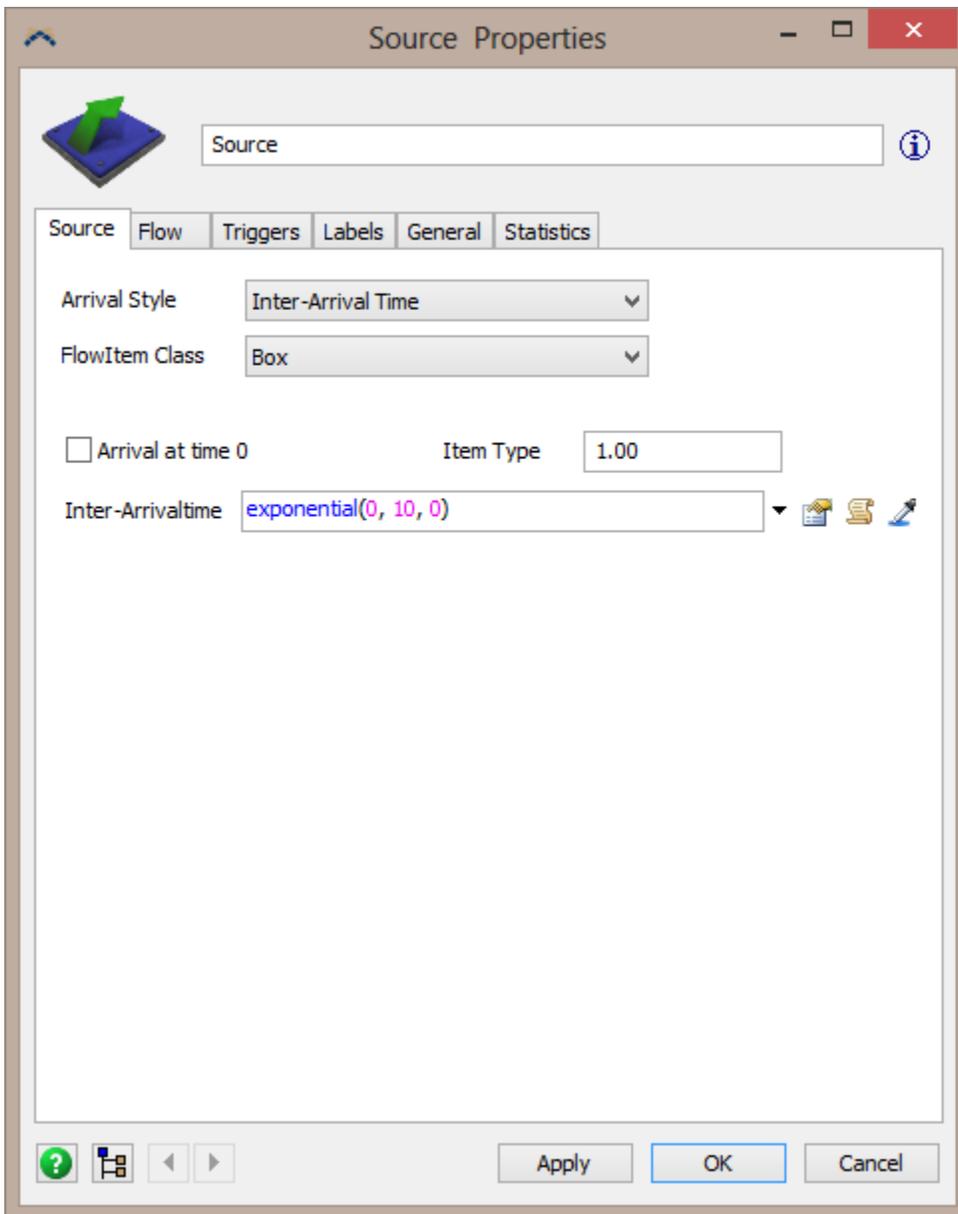
Produto chega a cada 5 segundos, exponencialmente distribuído. O Source, por padrão usa uma distribuição exponencial para o inter-arrival time, mas você irá alterar a média desta distribuição. **Distribuições estatísticas como a exponencial são usadas durante a simulação para modelar as variações que acontecem em sistemas da vida real.**

Você pode editar o tempo entre chegadas através de duas janelas diferentes.

- 1) Dê um clique sobre o Source para abrir a janela de propriedades do objeto na janela de propriedades rápidas (Quick Properties window). Esta janela estará do lado direito de sua tela.

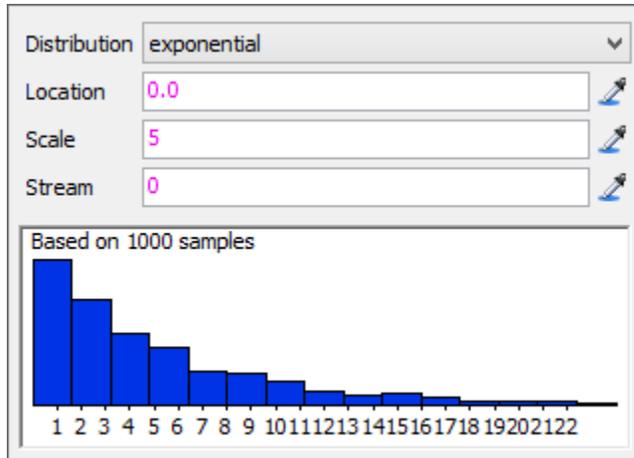


- 2) Dê um duplo clique sobre o source para abrir a janela de propriedades (**Properties**).



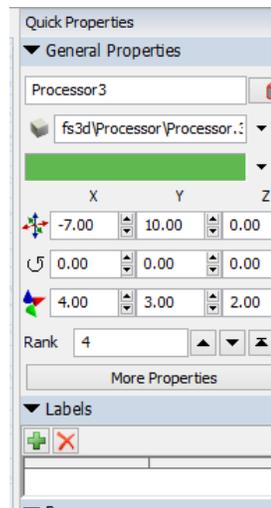
- Na guia **Source**, clique sobre o botão . Uma janela popup irá aparecer
 - Configura a distribuição (**Distribution**) para exponencial.
 - Digite para o parâmetro de localização (**Location**) o valor 0.

- Digite como parâmetro de escala (**Scale**) o valor 5.
- Digite em **Stream** o valor 0.



Clique em qualquer lugar fora da janela popup para salvar a configuração realizada. O botão  irá aparecer próximo ao campo drop-down **Inter-Arrivaltime**. Este botão abre a janela de opções das distribuições estatísticas. Lembre-se que as unidades foram definidas anteriormente. Coloque 5 para parâmetro de escala (**Scale**) que é a média da distribuição para 5 segundos. Se a unidade de tempo foi definido como horas, então a média será 5 horas.

- Se você editou o intervalo entre chegadas (Inter-Arrival time) através da janela de propriedades rápidas (Quick Properties window), você precisará abrir a janela de propriedades do Source para efetivar a etapa 6. Isto pode ser realizado ao clicar no botão **More Properties** Guiaixo da seção de propriedades gerais conforme imagem Guiaixo.



Distribuição Exponencial

Muitos problemas de simulação necessitam usar a distribuição exponencial. Isto é verdadeiro nos problemas que envolvem chegadas e partidas (filas), como a simulação de uma agência bancária, da saída (caixas) de um supermercado, de um aeroporto, etc...

A função de densidade probabilística da exponencial é igual a:

$$f(x) = \alpha e^{-\alpha x}$$

onde α é uma constante conhecida e a média (μ) é igual a ¹

A função de distribuição acumulada é dada por:

$$F(x) = 1 - e^{-\alpha x}$$

De modo a se fazer uso do método da transformação inversa devemos resolver para x . Assim temos:

$$\alpha \ln[1 - F(x)]$$

Etapa 6: Atribua um tipo de item e cor (Itemtype and color)

O próximo item que precisamos configurar é o número de itemtype para os flowitems conforme eles entram no sistema. Este valor é uniformemente distribuído entre 1 e 3, o que significa, ou seja, a chance de que o produto entrar é tipo 1 é tão provável quanto a do tipo 2, que é bem provável, para a do tipo 3. A melhor forma de fazer isso é ir no OnCreation trigger do Source.

- Clique na guia **Triggers**. Adicione a função (pressionando o botão ) em **OnCreation*** trigger. Selecione **Set Itemtype and Color**** da listagem. Uma tela popup contendo as funções pré-programadas irá aparecer conforme Guiaixo.

***Nota sobre o “OnCreation”:** Este código é executado quando um flowitem é criado em um modelo.

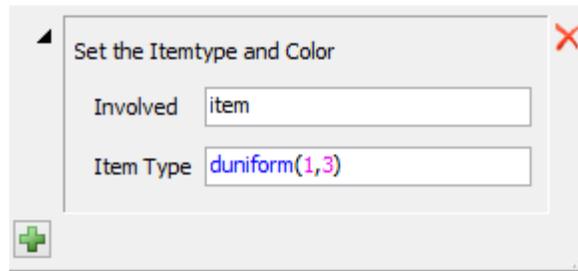
Variáveis de acesso de “OnCreation”:

current: o objeto atual

item: o objeto que acabou de ser criado

rownumber: o número da linha de chegada (se aplicável)

****Nota sobre “Set Itemtype and Color”:** Comando que encontra-se na listagem drop-down utilizado para criar tipos de itens (entidades) distintas e com cores diferentes.

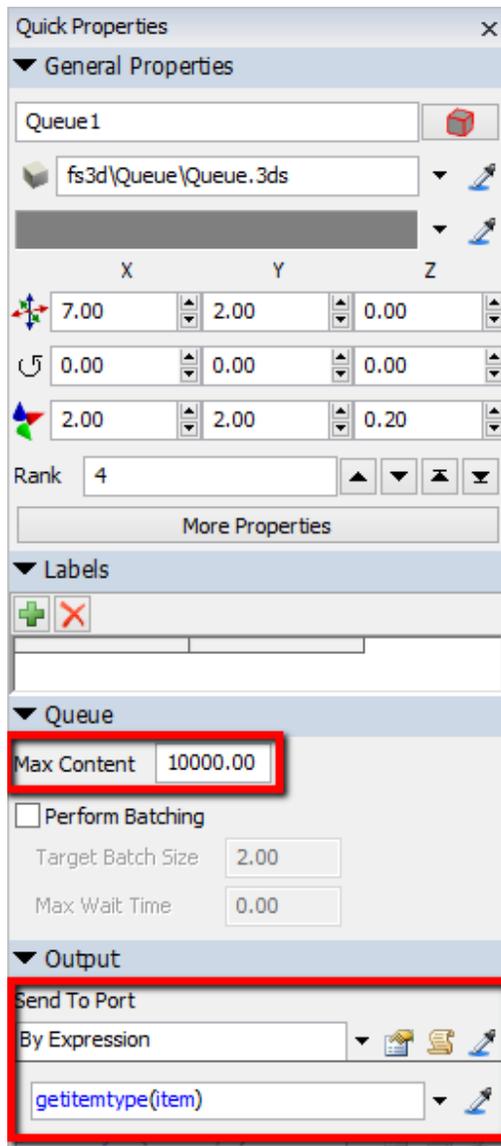


A distribuição *duniform* é semelhante a uma distribuição uniforme, exceto que em vez de devolver um número real só irá retornar números inteiros. Clique **OK** para aplicar as mudanças e fechar a janela.

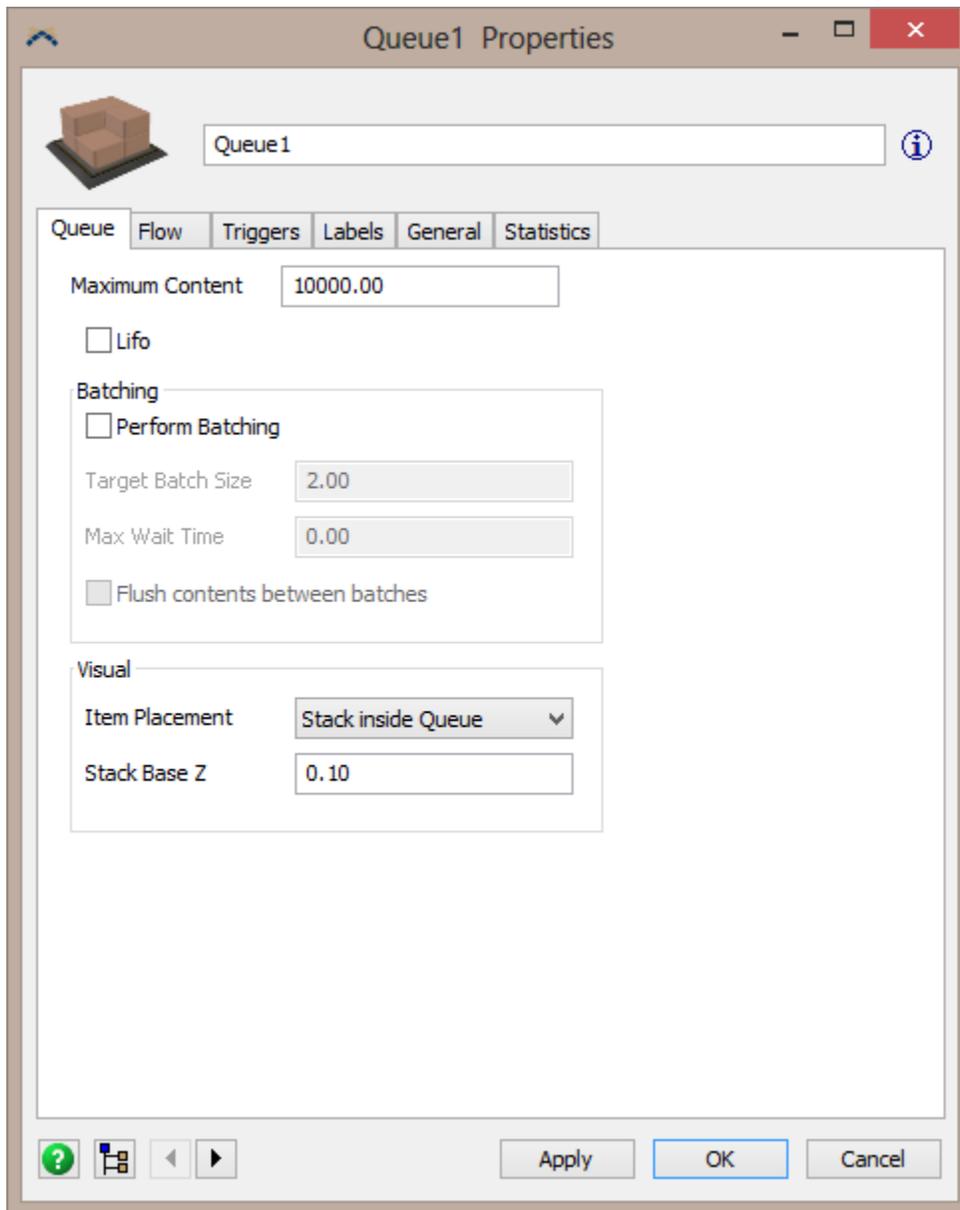
Etapa 7: Defina a Capacidade Máxima do Buffer (Queue1)

O próximo passo é editar o Queue. Existem duas coisas que nós precisamos configurar no *Queue1*. Primeiro, nós precisamos configurar a capacidade máxima do Queue. Segundo, nós precisamos que o Queue envie o itemtype 1 para o *Processor1*, itemtype 2 para o *Processor2*, e o itemtype 3 para o *Processor3*.

Esta etapa, juntamente com a etapa 8 pode ser feita através da janela Quick Properties conforme mostrado abaixo ou dando um duplo clique sobre o Queue1 para abrir a tela de propriedades, como descrito.

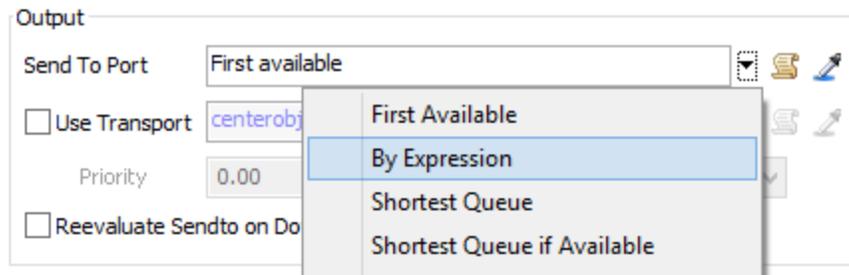


- Dê um duplo clique no *Queue1* para abrir a janela **Properties**.
- Na guia **Queue**, mude o **Maximum Content** para 10000.
- Clique em **Apply**, mas não feche a janela de propriedades.



Etapa 8: Defina a lógica de envio do primeiro Buffer

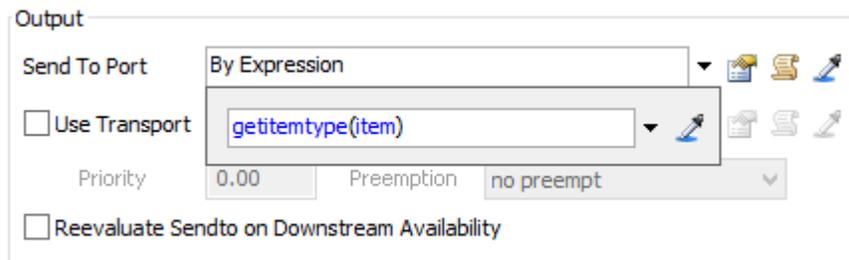
- Clique na guia **Flow**.
- Guiaixo de **Output**, selecione **By Expression** da listagem drop down em **Send To Port**.



- Uma janela com sugestões de expressões e funções pré-programadas irá aparecer. A expressão padrão para **By Expression*** é **getitemtype(item)****. Com esta expressão, o produto do tipo 1 será enviada para a porta 1, tipo 2 para a porta 2 e assim por diante. Clique em qualquer lugar fora da janela para fechar.

***Nota sobre “By Expression”:** Isto informa que o commando será executado de acordo com uma expressão lógica que enviará as entidades de acordo com os seus tipos para portas de saída específicas.

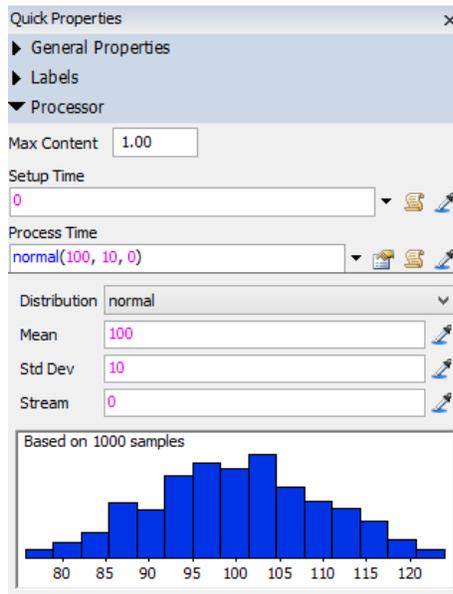
****Nota sobre “getitemtype(item)”:** Isso retorna o valor itemtype do item, ou seja, os itens serão diferenciados por tipo e serão enviados para seus respectivos destinos.



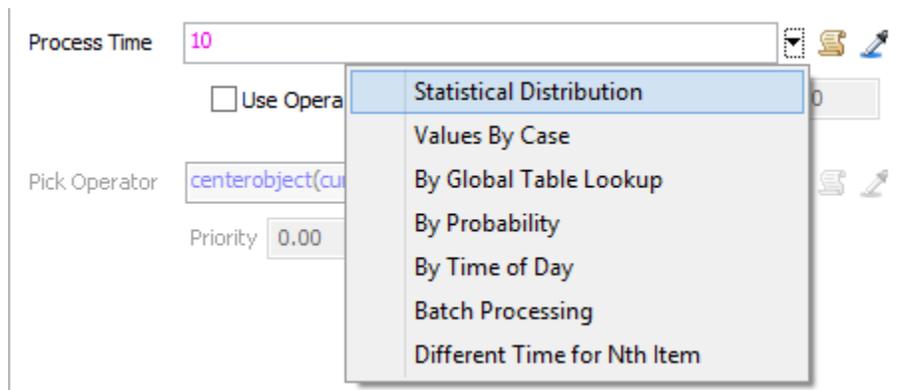
Etapa 9: Defina os tempos de processamento

O próximo passo será configurar o tempo de processamento para os três equipamentos (processors)

- Veja como é feito no Quick Properties.



- Dê um duplo clique sobre o *Processor1* para abrir sua janela **Properties**.
- Na guia **Processor**, no painel **Process**, selecione **Statistical Distribution** da lista **Process Time**.
- No popup **Statistical Distribution**, selecione a **Distribution** para **exponential**. Use os parâmetros padrão fornecidos para esta distribuição.
- Repita estes passos para o *Processor2* e *Processor3*.



Etapa 10: Defina a capacidade máxima do Buffer 2

Siga a etapa 7 para alterar a capacidade máxima (**Maximum Content**) do Queue para 10000.

Etapa 11: Defina o tempo de processamento da estação de teste

Como descrito na etapa 9, isto pode ser configurado através da tela Quick Properties também.

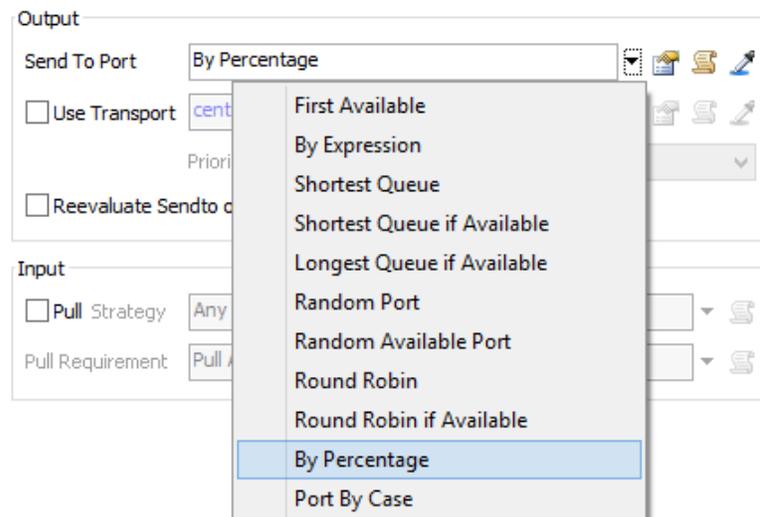
- Dê um duplo clique sobre o *Tester (processors)* para abrir a janela **Properties**.
- Na Guia **Processor**, no painel **Process**, selecione todo o texto presente no campo **Process Time**.
- Substitua o texto pelo número **4**. Isto configure o tempo de processamento para uma constante de 4 segundos.
- Clique em **Apply**, mas não feche a janela de propriedades.

Process Time  

Use Operator (s) for Process Number of Operators

Etapa 12: Defina a lógica de envio da estação de teste

Agora nós precisamos configurar a estação de teste (processors) para enviar produtos ruins de volta para o modelo e enviar os produtos fabricados sem defeito para o final do modelo. Quando você criou as conexões dos objetos, você deveria ter primeiramente conectado ao sink e então conectado um retorno para o primeiro queue. Conectando nesta ordem, irá fazer a primeira porta de saída da estação testing (processor) estar conectado ao sink e a segunda porta de saída da estação testing (processor), estar conectado ao *Queue1*. Você pode verificar se as portas estão conectadas corretamente, ao clicar em Output Ports no painel **Ports** na Guia **General**. Se as portas estiverem fora de ordem, você poderá arrumá-las utilizando os botões Rank ^ e Rank v para organizá-los. Agora, nós precisaremos definir a lógica de envio do tester, baseado na percentagem de produtos bons e ruins. Clique na guia **Flow**. Selecione **By Percentage** da listagem **Send To Port**



- Use o  para adicionar um novo campo.
- Preencha os campos de acordo com a imagem Guiaixo.

Specify Percentages (must sum to 100) and Values  

Percent 

Port 

Percent 

Port 

Use Random Stream

Isto significa que 80% dos produtos são bons (os produtos produzidos corretamente e dentro da especificação) e serão enviados para fora do modelo através do Sink, e 20% dos produtos não foram produzidos dentro da especificação (produtos produzidos incorretamente) e devem ser enviados através da porta de saída para o primeiro queue.

Mais uma coisa que talvez queremos visualizar seria os itens que estão saindo do modelo, os produtos bons, e os itens que estão sendo enviados para o primeiro queue, os produtos ruins.

- Clique na guia **Triggers**. Adicione a função (clique sobre o botão ) na trigger **OnExit*** e selecione a opção **Set Color****. Selecione colorblack(item) da listagem.

***Nota sobre Entry/Exit Trigger e OnEntry/OnExit Trigger**

Entry Trigger: Esta função é executada toda vez que um flowitem entra no objeto.

Exit Trigger: Esta função é executada cada vez que um flowitem sai do objeto.

Nota sobre a Trigger OnEntry/Exit: Geralmente, as trigger de entrada e saída são executadas no início do evento OnReceive / OnSend de um objeto, antes de qualquer outra lógica ser executada. Isso significa que você pode mudar as variáveis, labels do objeto, etc, e que essas alterações sejam aplicadas corretamente dentro da lógica do evento. No entanto, executar comandos que podem afetar outros eventos do objeto não deve ser executado na trigger on entry/exit, porque alguns eventos têm de ser criado na lógica de evento do objeto, e as funções que afetam os eventos do objeto devem esperar até que os eventos sejam criados. Nesse caso, você deve enviar ao objeto uma mensagem no tempo 0 (usando o comando senddelayedmessage ()), e em seguida, executar a funcionalidade da trigger message. Isso permite que o objeto termine o resto de sua lógica de evento antes de seus comandos serem executados. Os comandos que se aplicam a esta definição são stopobject (), requestoperators (), openoutput (), openinput (), resumeinput (), resumeoutput (), e, em alguns casos, criar e expedir as sequências de tarefas, dependerá dos tipos de tarefas que estão neles.

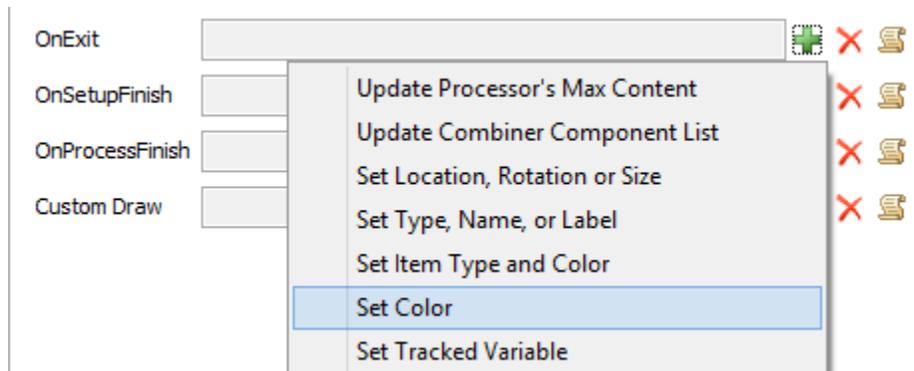
Variáveis de acesso:

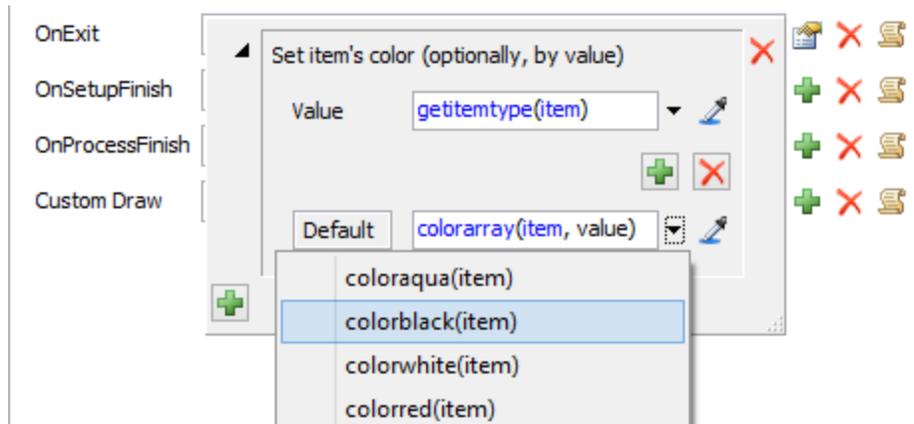
current: o objeto atual

item: o objeto envolvido, que só entrou / saiu

port: o número da porta que o objeto veio / através da porta da esquerda

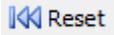
****Nota sobre Set Color: este comando é usado para configurar a cor de um item (entidade) quando ele passa pelo objeto . Neste caso usou-se o colorblack(item) para definir a cor preta do item.**

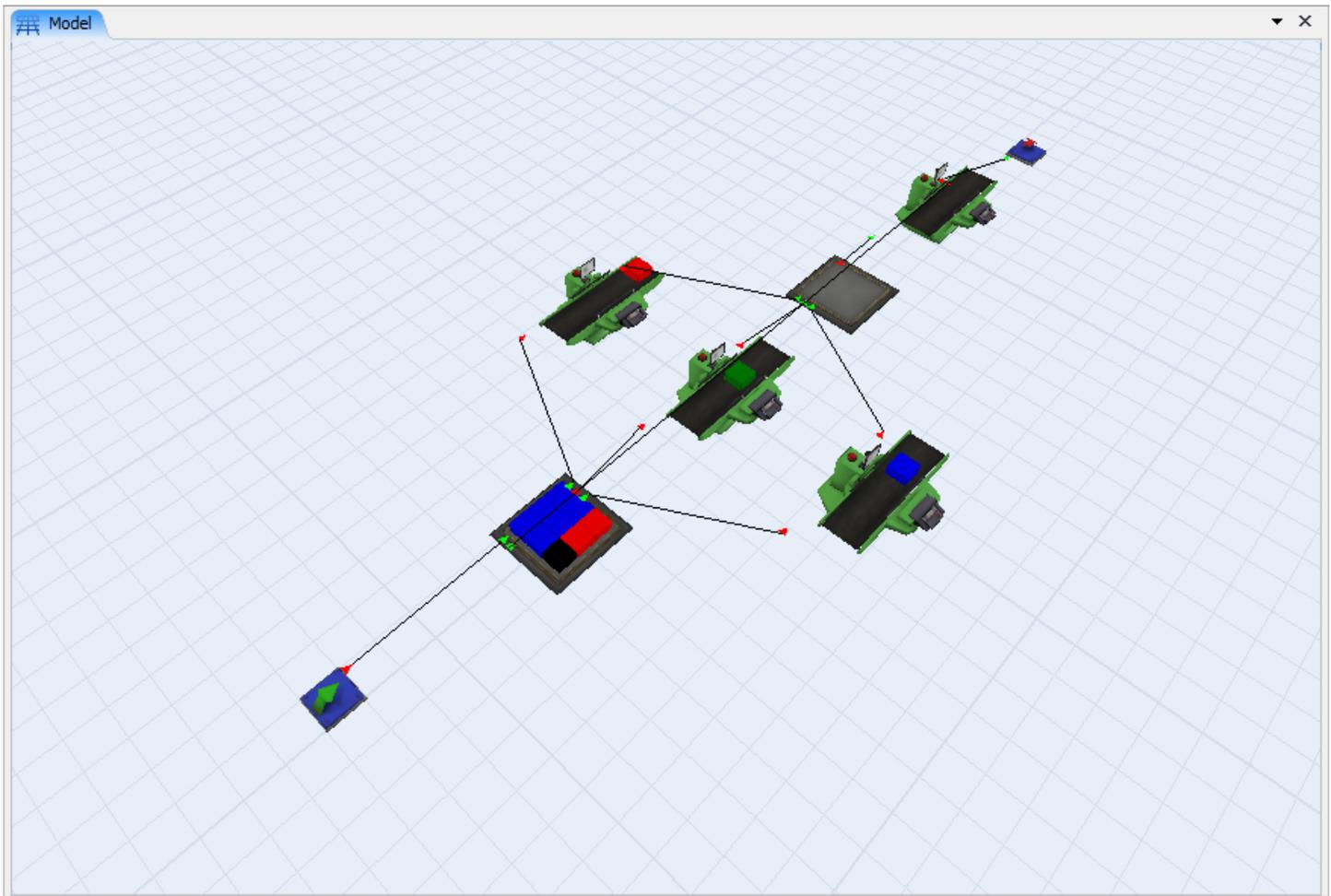




- Pressione **OK** para fechar a janela de propriedades.

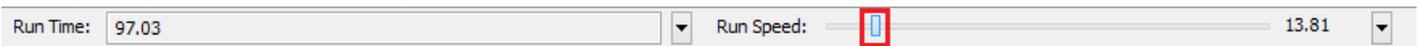
Etapa 13: Resete e Execute o Modelo

- Clique no botão  **Reset** localizado na parte esquerda superior da barra de ferramentas. Resetando o modelo define todas as variáveis do sistema para seus valores iniciais e limpa qualquer flowitems presentes no modelo. Resetar é necessário toda vez que nova conexão é realizada entre os objetos.
- Clique no botão  **Run** localizado logo a direita do botão de resetar.
- O modelo deve começar a rodar agora. Os produtos (Flowitems) devem se mover do primeiro queue, para um dos três processors, então, para o segundo queue, chamado de estação testing (processor), e de lá para o sink, com alguns produtos que não foram processados corretamente, sendo retornados para o primeiro queue. Os produtos que estão sendo retornados, estão retornando com a cor preta.



Para parar o modelo, pressione o botão  **Stop** a qualquer instante. Mais tarde, iremos mostrá-los como rodar um modelo para um tempo específico e para um número específico de iterações. Rodando o modelo mais de uma vez é importante quando uma distribuição estatística for utilizada para a definição inicial do modelo.

Para aumentar ou diminuir a velocidade da simulação, mova a barra que encontra-se ao lado de Run Speed para a direita ou esquerda.



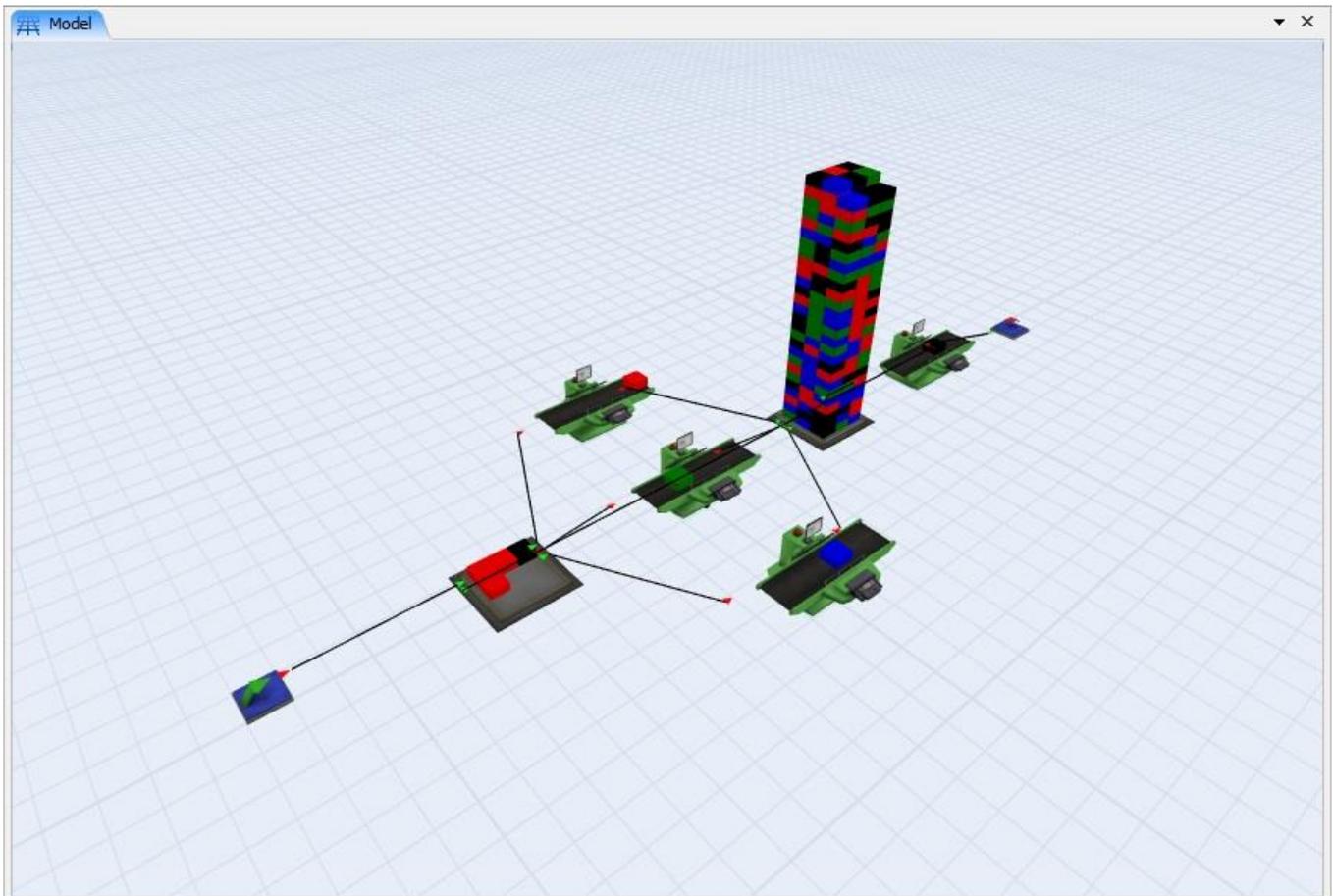
Movendo a barra, muda-se a velocidade com que o modelo é executado em relação ao tempo real. Isto não afetará os resultados do modelo.

Nós agora completamos as etapas para o desenvolvimento de um modelo. Vamos agora analisar algumas informações estatísticas que o modelo gera.

Realize experimentos com o modelo

Encontrando o gargalo

Na descrição do modelo, disse que queríamos saber onde estava o gargalo no sistema. Existem várias maneiras de determinar isso. Primeiro, você pode simplesmente examinar o tamanho visual de cada queue (fila). Se em uma fila (queue) no modelo contiver de forma consistente muitos produtos acumulados nele próprio, então isso é uma boa indicação de que as estações de processamento (s) que alimentam estão causando um gargalo no sistema. Na execução deste modelo, você vai notar que o segundo queue (fila), muitas vezes tem um monte de produtos à espera para ser processado, enquanto que o conteúdo do primeiro queue é geralmente de 20 ou menos.



Outra forma de encontrar a localização de um gargalo é examinando as estatísticas do estado de cada um dos processadores. Se os três processadores a montante estão sempre ocupado, enquanto que a estação de testing (processor) estiver inativa, então, é bem provável que o gargalo seja os três processadores iniciais. Por outro lado, se a estação de testing (processor) estiver sempre ocupado, e os processadores a montante estiverem frequentemente ociosos, o gargalo provavelmente estará na estação de testing (na imagem acima, o processador que encontra-se sozinho logo após o segundo queue).

Execute o modelo, pelo menos, para 50 mil unidades de tempo. Então pare o modelo e abra a janela de propriedades do primeiro dos três processadores iniciais.

Clique sobre o objeto processor e no painel Quick Properties a direita de sua tela você consegue visualizar os dados estatísticos do processor.

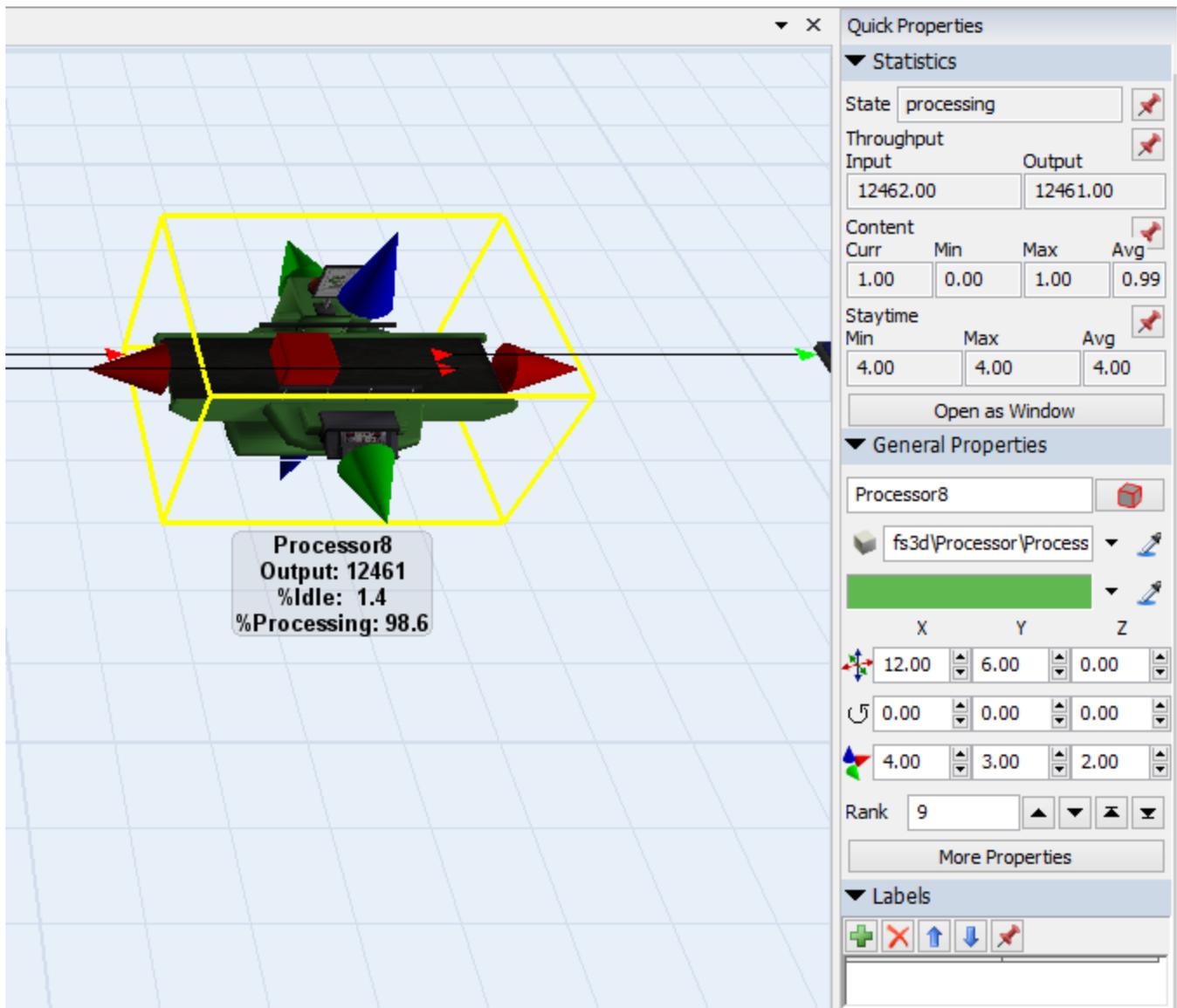
The screenshot displays a 3D simulation environment with a grid floor. A production line is visible, including a source on the left, three processors (Processor3, Processor4, Processor5), and a queue (Queue6) on the right. Processor3 is highlighted with a yellow box and has a data tooltip showing: Output: 4265, %Idle: 14.8, and %Processing: 85.2. The 'Quick Properties' panel on the right is open for Processor3, showing the following statistics:

Statistics			
State	processing		
Throughput			
Input	4266.00	Output	4265.00
Content			
Curr	Min	Max	Avg
1.00	0.00	1.00	0.85
Staytime			
Min	Max	Avg	
0.00	88.46	10.09	

Below the statistics is an 'Open as Window' button. The 'General Properties' section shows the object name 'Processor3' and its path 'fs3d\Processor\Process'. It also includes 3D coordinate fields for X, Y, and Z, with values -4.00, 13.00, and 0.00 respectively. A 'Rank' field is set to 4. The 'Labels' section at the bottom contains icons for adding, deleting, and moving labels.

As informações mostram que a estação de processamento estava ocioso em 15% do tempo de simulação, e que estava ocupado em 85 % do tempo, durante a simulação. Feche esta janela e, em seguida, clique duas vezes em cada um dos outros dois processadores para ver suas estatísticas. Eles devem ter um resultado muito próximo do apresentado acima.

Dê um duplo clique na estação de Teste para abrir a janela **Properties**. Analise as informações estatísticas conforme fizemos para o processador acima. Veja que o tester é o nosso gargalo no sistema, pois está em aproximadamente 98,6% processando as entidades (flowitems) e com tempo médio de permanência das entidades no equipamento de 0.99 unidades de tempo.

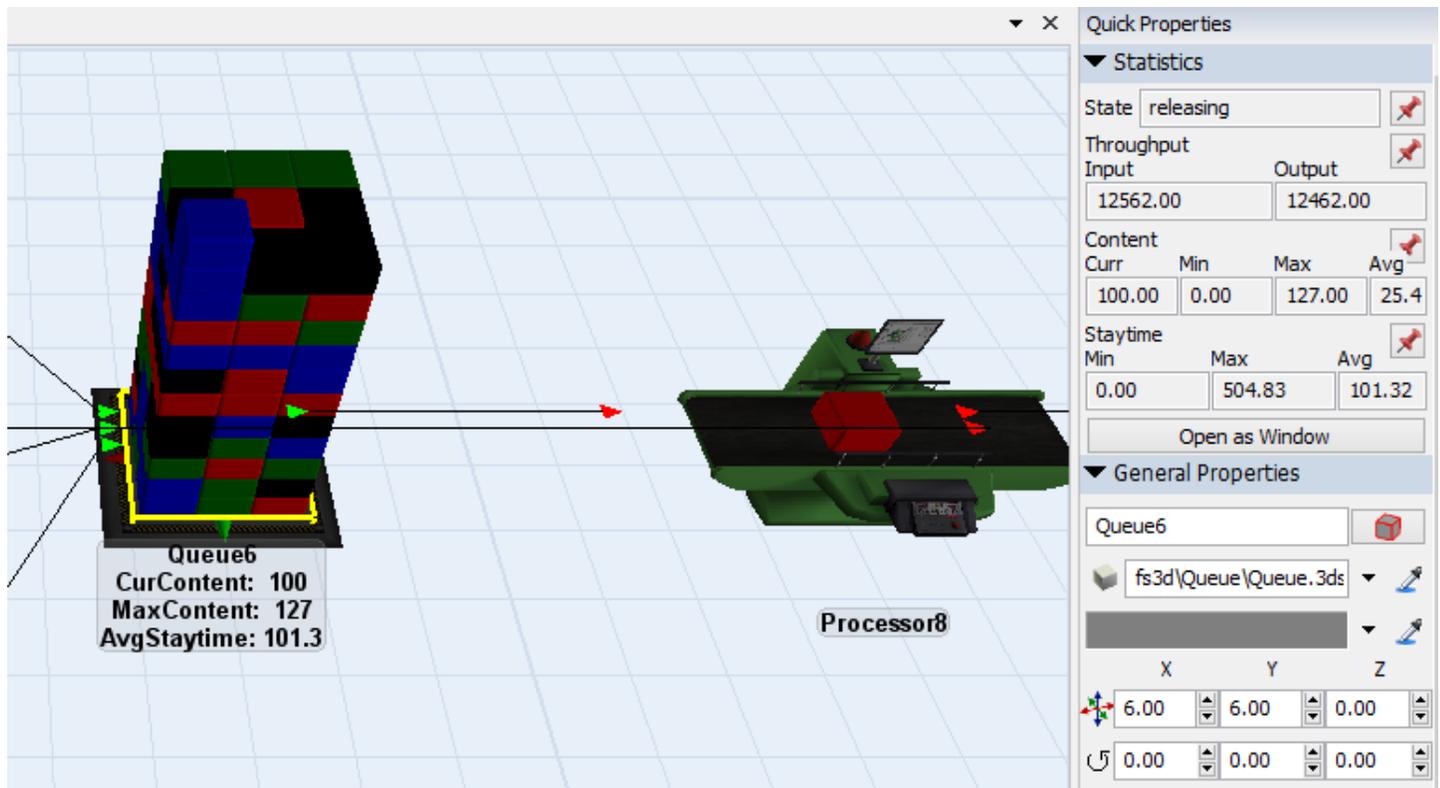


Agora que descobrimos onde o gargalo está presente, a questão é, o que devemos fazer a respeito? Isto depende de vários fatores de custos versus ganhos, bem como sobre os objetivos futuros da empresa. No futuro, a instalação atual será capaz de empurrar mais produtos através de um ritmo de produção mais rápido? Em nosso modelo, o source produz um produto a cada cinco segundos, em média, enquanto a estação testing envia um produto para o sink uma vez a cada 5 segundos, em média. Este valor médio de 5 segundos para a estação testing (processor) pode ser calculada utilizando o tempo de ciclo de 4 segundos e sua estratégia de envio 80/20. Assim, ao longo do tempo, o total de níveis de capacidade do nosso modelo ficará off. Se a fábrica começar a empurrar mais produtos por esta parte da instalação, isso equivale a uma taxa de chegada superior (um menor tempo de Inter chegada ou inter-arrival time) para o source. Se, não fizemos alterações para a estação testing (processor), o nosso modelo poderia acumular continuamente mais e mais partes, e o conteúdo das filas continuaria a aumentar até que não houvesse mais espaço disponível para acomodar os produtos. Para corrigir isso, vamos ter de adicionar uma segunda estação testing (processor), tendo em vista que encontramos o gargalo.

Outra situação em que pode-se querer adicionar uma outra estação de testing (processor) é se o tamanho da fila da fila do testador (processor) for muito importante para nós. Se é caro para permitir que o tamanho da fila do testador (processor) acumule quantidades elevadas de produtos, então seria inteligente adicionar uma segunda estação de testing (processor) para certificar-se de que o tamanho

da fila, bem como o tempo de espera de cada produto na fila, não fique elevado. Vamos olhar para as estatísticas da fila.

Clique em *Queue2* para abrir o painel **Quick Properties** e veja as informações estatísticas do Queue



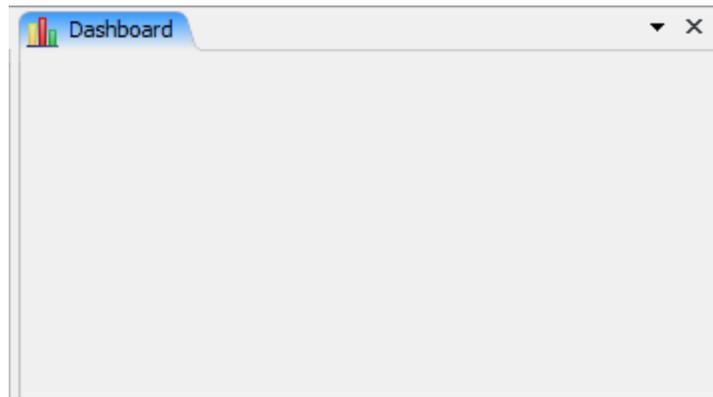
Continue a executar o modelo, e você verá que estes valores mudam conforme a simulação é executada. Olhe para os valores de conteúdo ou capacidade média e os valores médios do staytime. Staytime refere-se ao tempo que os flowitems (produtos) permaneceram no queue, ou tempo de permanência na fila. Logo no início da simulação, a capacidade ou conteúdo máximo do queue é geralmente baixo, mas, conforme a simulação prossegue, pode atingir valores elevados como 150 ou 200 (Staytime Max). Se um tamanho médio da fila de 150 ou 200 é inaceitável, então pode ser necessário adicionar uma segunda estação testing (processor).

Crie os dashboards para análise anterior

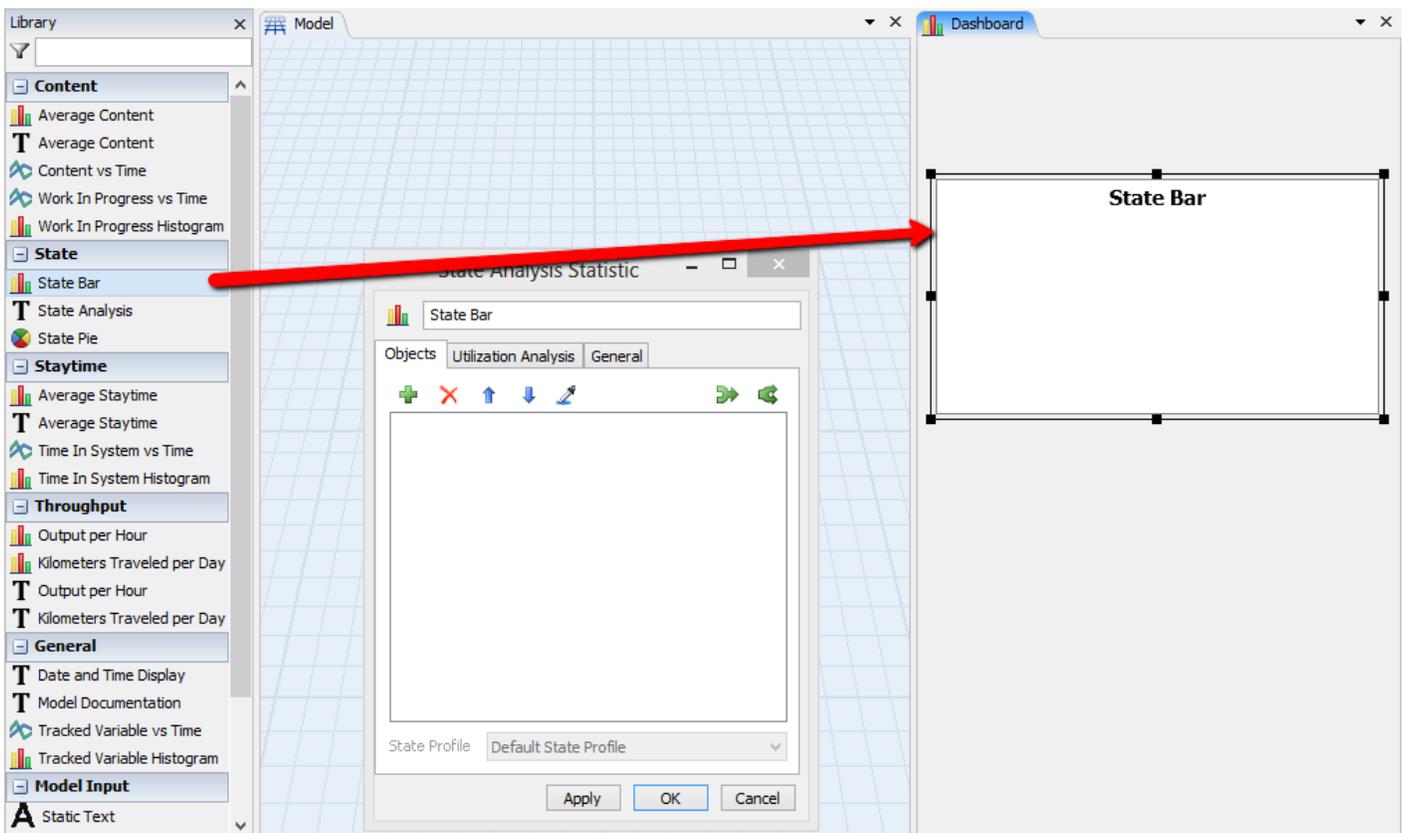
Avaliando o exercício anterior

Rode o modelo por pelo menos 50.000 segundos. Verifique primeiro que o *Queue2* está agora quase sempre vazio, enquanto o Queue para os 3 processors está atuando com elevada frequência. Vamos usar o dashboard para comparar as duas estações de testing (processor) lado a lado.

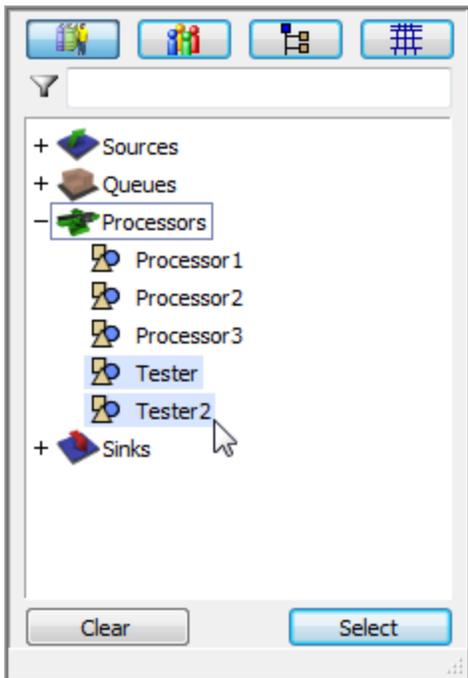
- Acesse o menu **Statistics**, selecione **Dashboard**, o qual encontra-se na última opção. O **Dashboard** irá aparecer.



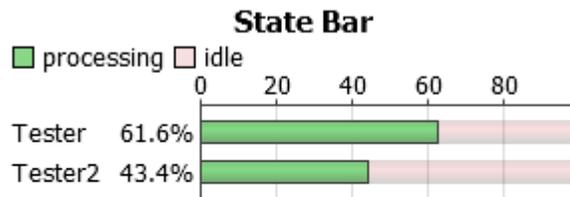
- Arraste o ícone  para dentro da janela de Dashboard. Isto deve trazer uma janela de seleção conforme Guiaixo.



- Na guia **Objects**, clique no . Isto abrirá outra janela.
- No popup clique em , expanda os **Processors**, e selecione *Tester* e *Tester2*.



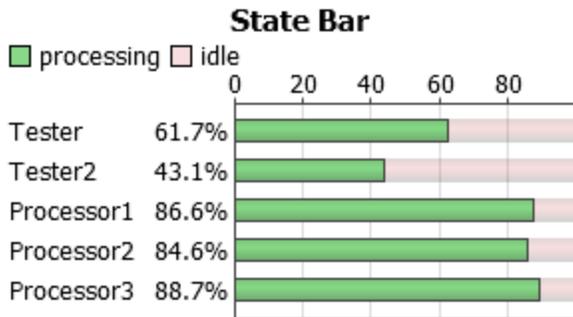
- Clique no botão **Select** para finalizar sua seleção. Então, clique em **OK**. Um gráfico em branco deve aparecer dentro do dashboard.
- Resete e execute o modelo novamente. O gráfico no Dashboard será dinamicamente e automaticamente atualizado.



A razão pelo qual o percentual está diferente para as duas estações testing (processors) é porque o testing do queue envia para a primeira estação testing (processor) disponível. Sempre que ambos os testers (processors) estiverem disponíveis, um produto será enviado para a primeira estação testing (processor) do modelo, tendo em vista que é o primeiro disponível ou *the first available*. Produtos somente irão para a segunda estação testing (processor) se a estação original já estiver ocupada. Além do mais, a estação testing (processor) original possui uma maior utilização que a segunda estação de testes (processor).

Agora adicione os outros três processadores no gráfico **State Bar**.

- Dê um duplo clique no gráfico, e a mesma tela de seleção dos objetos irá se abrir.
- Selecione *Processor1*, *Processor2*, e *Processor3* da lista de seleção. Seja qual objeto for selecionado será adicionado ao conteúdo anterior do gráfico.
- Resete e rode o modelo novamente. Agora, todos os 05 processadores podem ser comparados lado a lado.



Nós eficientemente mudamos o gargalo do sistema da estação testing (processors) para os três processors. Também, ao aumentarmos a produtividade em 15% e conseqüentemente adicionar outra estação testing (processor), nós diminuimos significativamente a utilização de cada estação testing (processor). Se isso é uma boa decisão, dependerá muito do custo para adicionar uma segunda estação testing (processor). Agora, visto que o gargalo está nos 03 processors, e com o objetivo de aumentar a produtividade, e além do mais, aumentar a utilização de cada estação testing (processor), teríamos que adicionar mais processors. Novamente, existe uma análise de custo/benefício nesta decisão. Você pode tentar alterar qualquer parâmetro (como o tempo de processamento dos processors) e verificar o efeito no modelo. Mesmo as pequenas mudanças podem causar grandes impactos no modelo.

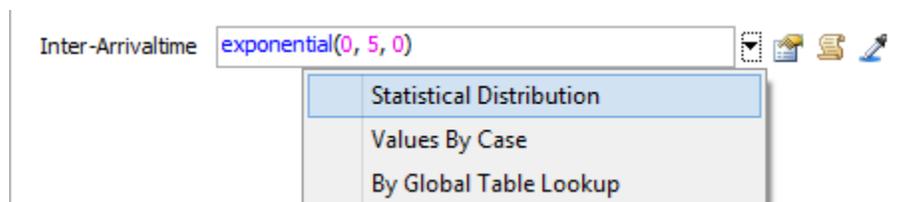
Aleatoriedade

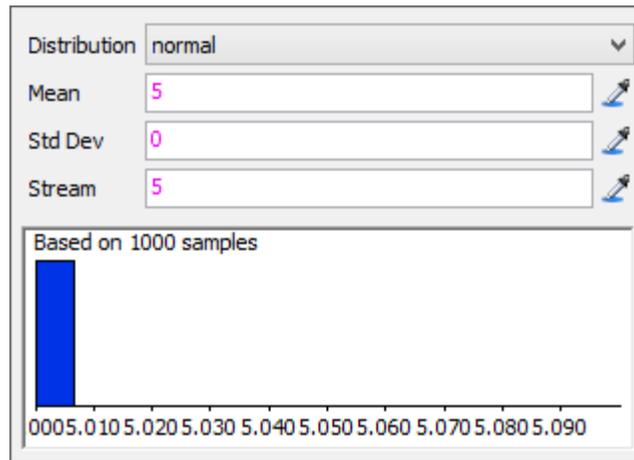
Vamos fazer alguns testes adicionais antes de nós decidirmos por adicionar outra estação testing (processor). Tendo em vista que em média, um produto chega do source a cada 05 segundos, e em média, um produto vai para o sink a cada 5 segundos, porque o queue deveria estar acumulando todos produtos? Os produtos estão deixando o modelo tão rápido quanto eles estão entrando, então, não deveríamos ter acúmulo de produto no sistema.

A razão pelo qual o queue está acumulando é por causa da aleatoriedade do sistema. Sim, em média, o produto chega ou é criado a cada 5 segundos, mas esta taxa de chega obedece uma distribuição exponencial. Para uma distribuição exponencial com um média igual a 5, na maioria do tempo, os produtos irão chegar de fato, a uma taxa superior a 5 segundos. Mas de vez em quando, haverá um longo período de seca, onde não há produtos chegando. No final, se equilibrará com uma média de 5 segundos, e os produtos podem chegar mais rapidamente, e, assim, acumular-se no queue anterior a estação de teste, uma vez que a estação testing é o gargalo.

E se, em sua fábrica, os produtos chegarem a uma taxa mais previsível, ao invés de chegarem por uma taxa que representa uma distribuição exponencial imprevisível? Vamos testar.

- Dê um duplo clique *Source* para abrir a janela **Parameters**. Em **Inter-Arrival** selecione **Statistical Distribution**. O pop up de Statistical Distribution irá aparecer.





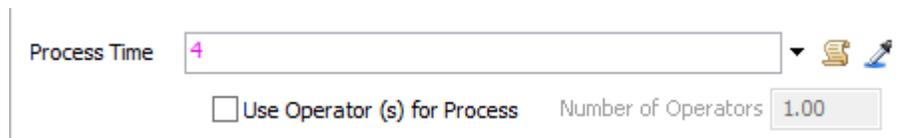
- Uma vez configurado resete (**Reset**) e rode (**Run**) o modelo novamente.

Se você ainda não tiver a janela de propriedades do *Queue2*'s disponível, abra-a novamente através de um duplo clique sobre o *Queue2*. Continue rodando o modelo. Você irá notar aqui que o conteúdo de produtos no queue não aumentará muito. Na verdade, não irá passar de 50 ou 60 enquanto antes, eles às vezes poderia chegar até a 150 ou 200. Isto é uma diferença significativa causada por simplesmente pela aleatoriedade presente no modelo.

Maior Produtividade

Agora, imagine que a instalação (fábrica), precise aumentar significativamente a taxa de produtividade do sistema em 15%. Isto significaria, aumentar o tempo médio de inter-arrival do source de 5 para 4.25 segundos. Tendo em vista que a estação testing (processor) já encontra-se em 100% de sua utilização, nós teremos que obviamente, adicionar uma segunda estação testing (processor) no sistema. Vamos fazer esta mudança.

- Dê um duplo clique em *Source* novamente para abrir a janela **Properties**. Clique no botão  para o **Inter-Arrivaltime**. Mude a média da distribuição normal de 5 para 4.25. Clique em **OK** na janela Properties do source.
- Agora, iremos criar a segunda estação de teste (processor). Crie outro objeto **Processor** no modelo e coloque-o Guiaixo de *Tester*. Nomeie como *Tester2*.
- Conecte o *Queue2* com o *Tester2*.
- Conecte o *Tester2* com *Sink* e com o *Queue1*.
- Dê um duplo clique em *Tester2* para abrir a janela **Properties**.
- Na guia **Processor**, no painel **Process**, selecione o texto do campo **Process Time**. substitua o texto por **4**



- Clique em **Apply**, mas não feche a janela Properties.
- Clique na guia **Flow**. Guiaixo de **Sent To Port** selecione a opção **By Percentage**. Entre com os mesmos parâmetros que você utilizou para o *Tester1*.
- Adicione a trigger (OnExit) para alterar a cor para black, exatamente como no outro Tester (processor).
- Ao terminar de fazer as mudanças, **Reset** e **Run** o modelo novamente.

Resultados

Através da criação de um modelo que simula o nosso sistema, temos claramente determinado os efeitos que certas decisões terão sobre o sistema. Podemos agora usar as informações que recolhemos a partir da simulação para tomar decisões melhor fundamentadas para o futuro da empresa

Com este modelo simples, muitas das mesmas conclusões poderiam ter sido feitas através de modelos matemáticos e fórmulas. No entanto, os sistemas reais são frequentemente muito mais complexo do que o modelo que acabamos de construir, e estão fora do escopo da modelagem matemática. Usando simulação com o Flexsim, podemos modelar essas complexidades da vida real, e examinar os resultados, assim como fizemos neste modelo.

Flexsim também dá as suas simulações um forte apelo visual. É muito mais fácil convencer uma equipe de gestão e planejamento, se a equipe pode ver os efeitos dessa decisão em uma realidade em 3D. Este ambiente é criado automaticamente quando você constrói seus modelos com o FlexSim.

Qual o próximo passo?

Agora que você tornou-se familiar com o FlexSim e o uso em simulação, nós sugerimos seguir o passo a passo do tutorial para resolução de alguns exemplos. O tutorial encontra-se no menu Help do software.

Ativação das licenças

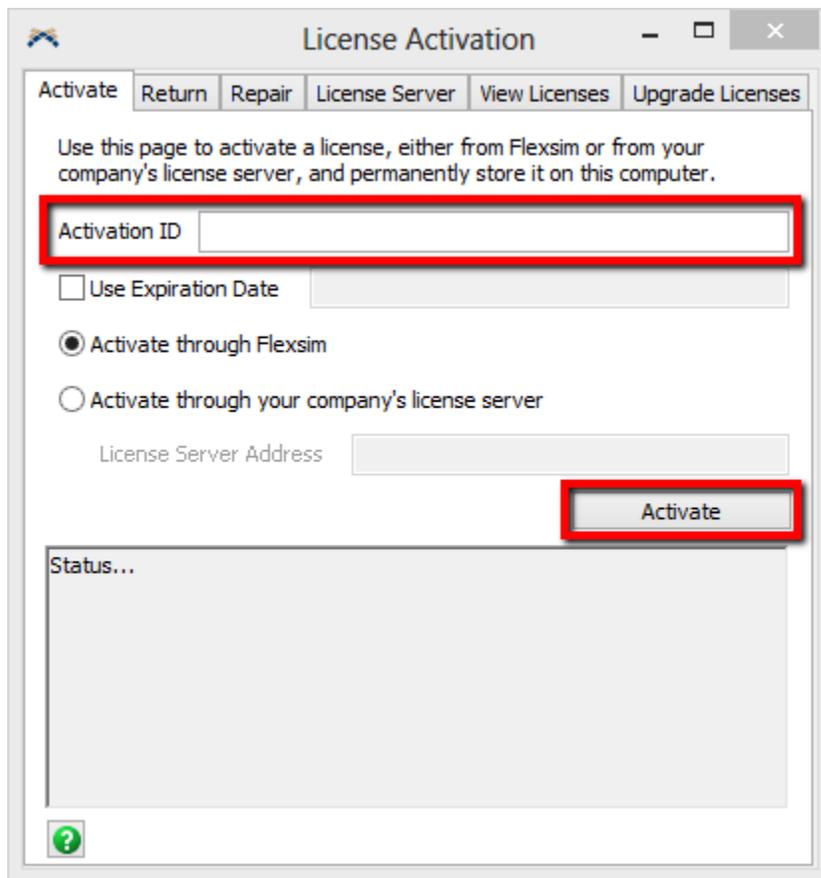
1. Conceitos
2. Exemplo
3. Referência

Conceitos de Ativação das licenças

A janela License Activation é encontrada no menu principal em **Help > License Activation**. Desta janela, você pode ativar licenças standalone, retornar licenças, reparar licenças, configurar para acessar como concurrent license server, visualizar os direitos do licenciamento mantidos no Flexnet trusted storage, e upgrade (atualizar) licenças standalone.

Tópicos

- **Licenciamento**
- **Licenças de Servidores**
- **Reparando Licenças**
- **Visualizando Licenças**
- **Realizando Upgrade das licenças**



The screenshot shows the 'License Activation' window with the 'Activate' tab selected. The window title is 'License Activation'. Below the title bar, there are tabs for 'Activate', 'Return', 'Repair', 'License Server', 'View Licenses', and 'Upgrade Licenses'. The main content area contains the following elements:

- A text box for 'Activation ID' (highlighted with a red box).
- A checkbox for 'Use Expiration Date' with an empty text box next to it.
- Two radio buttons: 'Activate through Flexsim' (selected) and 'Activate through your company's license server'.
- A text box for 'License Server Address'.
- An 'Activate' button (highlighted with a red box).
- A 'Status...' label above a large empty text area.
- A help icon (question mark in a green circle) in the bottom left corner.

Licenciamento

Flexsim não precisa de uma licença para sua versão demo e gratuita. A versão gratuita permite você criar no máximo 20 objetos em seu modelo e rodar o modelo com vários cenários.

Se você quiser comprar a licença do software você pode contatar nosso departamento de vendas pelo telefone (19) 3308-7833 ou vendas@flexsimbrasil.com.br. Uma vez que você ativou a licença com sucesso através do servidor de sua empresa ou da FlexSim, muito das opções de menu que estão desabilitadas na

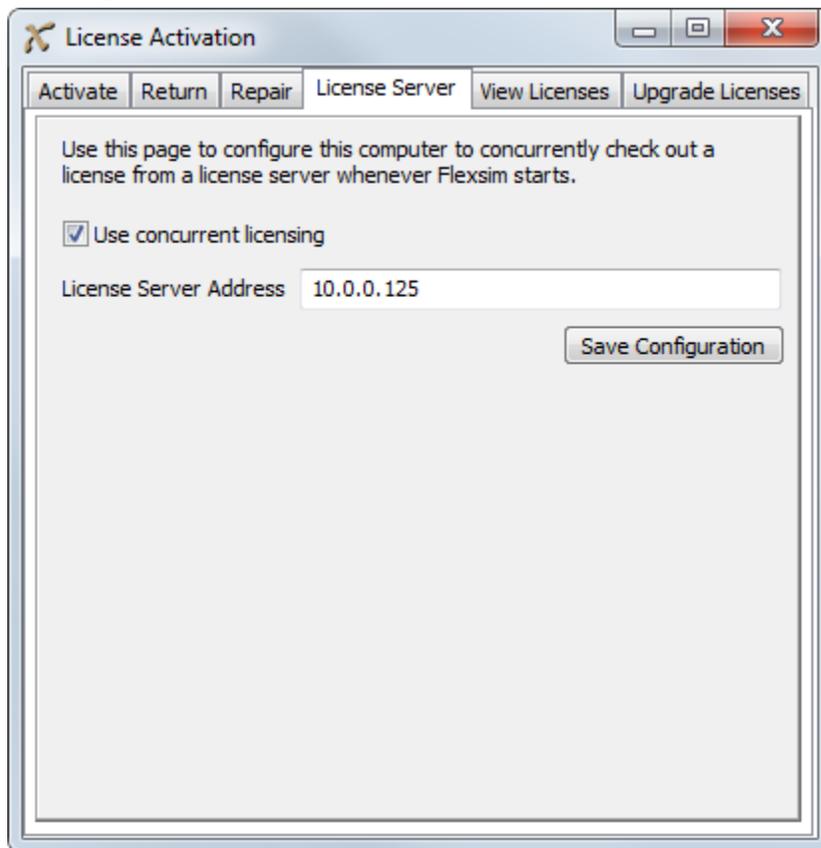
versão demo, irão se tornar disponíveis. Você também pode visualizar o tipo de licença em **Help > About FlexSim**.

Para uma explanação sobre os erros apresentados enquanto estiver ativando/retornando as licenças, visite a página de referência.

Repair (Reparar a licença)

Na guia repair, você vai ver todas as licenças ativadas em seu computador. Licenças são mantidas no FLEXnet trusted storage em um computador. Algumas informações sobre as licenças são armazenados em vários locais em seu disco rígido e registro. Adulteração destes locais podem quebrar as licenses trust flags, que desativa a licença. Isso pode acontecer com alguns limpadores de registro ou pontos de restauração do Windows. Se a sua licença foi desativada, então você pode usar a guia de reparação para gerar um arquivo xml de reparo. Envie este arquivo por e-mail para o suporte da FlexSim através do site em (www.flexsim.com/support) e então você vai receber de volta um arquivo .xml para processar ou reparar. Processando o arquivo de reparação com apoio da Flexsim irá reparar as trust flags em suas licenças desabilitas e permitirá que ela seja usado novamente.

Licenças de Servidores



Se você estiver usando uma concurrent license server, você pode configurar os clientes para se conectar ao servidor usando este guia. Marque a caixa e digite o endereço IP do servidor de licença. Pressione o botão Save Configuration, e em seguida, feche e reabra o Flexsim, a fim de verificar a licença do servidor.

Se você obteve com sucesso a licença do servidor, então muitas das opções do menu que estavam acinzentado na opção demo, agora estarão disponíveis. Você também vai ver o tipo de licença em Help> Sobre a janela Flexsim em vez de License: None.

Antes de tentar conectar os clientes para um servidor concorrente, você deve ativar suas licenças no servidor e iniciar o License Server Manager programa no servidor. Instruções e arquivos necessários de configuração do servidor estão disponíveis nas ferramentas de Licenciamento LAN, disponível para

download na seção de conta (account) no site Flexsim se você tiver um servidor de licenças simultâneas (concurrent server license) em sua conta Flexsim.

A entrada de registro no Windows para o FLEXSIM, nosso gestores de licenças, armazena qualquer localidade que o FlexSim tem assegurado uma liberação para licença em rede, e pode armazená-la em múltiplas localidades. Sempre que você entrar com um novo valor em Concurrent License Server Address e reiniciar o FlexSim, ele adiciona mais entradas aos registros. Se você realizou o check out de uma licença do servidor, FlexSim pode realizar com sucesso o check out da licença daquele servidor quando você iniciar o FlexSim, mesmo se a caixa "Use concurrent licensing" não tiver selecionada.

Visualizando Licenças

Na guia view license, você pode ver o conteúdo do Flexnet trust storage do computador. Licenças não são de modo algum vinculado a qualquer instalação do Flexsim. Seu computador está licenciado. A versão atual de instalação do Flexsim não importa. Basta usar o FlexSim para ativar / voltar / ver licenças no computador para simplificar. As licenças não estão vinculados a instalação do programa Flexsim.

Por exemplo, você poderia instalar o FlexSim, ativar uma licença, desinstalar o FlexSim e instalar novamente, e sua licença ainda estará lá. Os programas Flexsim estão totalmente separados onde as licenças são armazenadas no seu computador em Flexnet trust storage.

Não importa qual a versão do Flexsim você usar para ativar ou devolver licenças; Eles estão chamando exatamente o mesmo código.

Você pode ter apenas uma licença de 5.1 no computador e a licença irá trabalhar com ambos 5.0.4 e 5.1.0 sem problemas. Você não precisa de várias licenças para licenciar vários programas Flexsim. A licença não está ligada à instalação. A licença está ligada ao computador. Lembre-se, um computador tem uma licença 5.0 que não será capaz de executar o software 5.1. A licença do computador deve ser \geq número da versão do software, a fim de executar.

Se você desinstalar Flexsim sem antes devolver sua licença, você terá que reinstalar Flexnet a fim de devolver a licença para que você possa movê-lo para outro computador. Como as licenças estão ligadas ao computador, e não a instalação Flexsim, você não precisa retornar a sua licença se você está simplesmente desinstalando e reinstalando o Flexsim.

Realizando upgrades das licenças

Na guia upgrade, você pode solicitar upgrades para suas licenças. Isso é necessário para executar as versões mais recentes do software. Por exemplo, se você tivesse usando a versão Flexsim 5.0 anteriormente instalado e atualizado para o 5.1, você precisa atualizar também a sua licença para 5.1.

O processo para atualizar uma licença para uma versão mais recente tem três etapas:

1. O Flexsim License Server precisa receber a ordem para atualizar a licença, que cria um novo activation ID para o upgrade.
2. A licença antiga do Flexsim necessita ser retornada para o servidor de licenças.
3. A nova licença precisa ser ativada na máquina do cliente.

Este processo pode ser feito manualmente, retornando as licenças antigas, pressionando o botão Upgrade Licenses em "my licenses" na seção de sua conta no site da Flexsim, e, em seguida, ativando os novos ids de ativação que aparecem na página (você pode precisar atualizar a página par ver o novo Activation ID).

O botão Request Upgrades na Guia upgrade licenses automatiza esse processo em um único clique de botão, enviando solicitações HTTP para o servidor FlexSim. Se este processo automático não funcionar, então você pode fazê-lo manualmente.

Para atualizar licenças de servidor, você deve retornar manualmente as licenças e ativar as novas licenças.

Retornando uma licença de servidor:

No FlexSim ServerActUtil, vá em Tools > View License Rights.

Copie o Fulfillment ID selecionando-o e pressionando Ctrl-C.

Vá em Connect > Return.

Cole Fulfillment ID dentro do campo usando Ctrl-V.

Pressione o botão Return.

Exemplos de Ativação de Licenças

Tópicos

- **Ativando uma Licença Padrão**
- **Ativando através de uma licença de servidor**
- **Retornando a licença**
- **Reparando a licença**
- **Ativando uma licença concorrente**

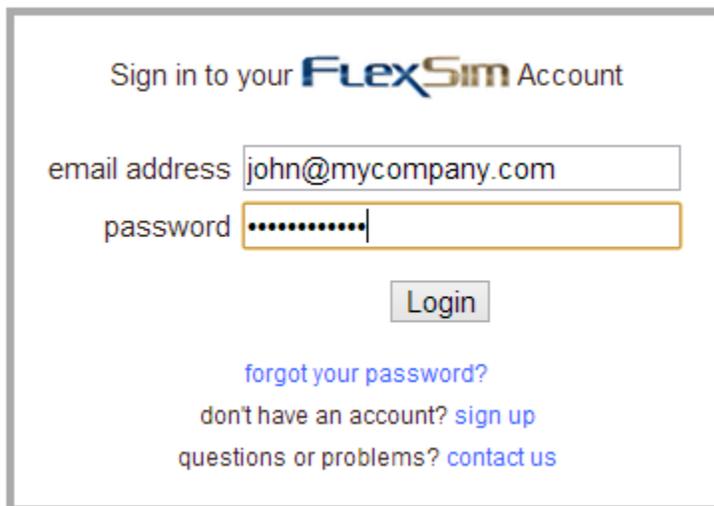
Ativando uma licença padrão

Isto irá levar você através das etapas necessárias para ativar uma licença padrão do FlexSim através dos servidores de licença da Flexsim. Se a sua empresa tem os próprios servidores, veja como ativar através dos servidores. Se sua empresa está usando uma licença de servidor concurrent, veja como ativar as licenças concurrent.

Faça o login em sua conta no site da FlexSim

Primeiramente você precisará a licença disponível dentro de sua conta da FlexSim, após a compra.

- Vá para <http://www.flexsim.com/account/> e realize o login.



Sign in to your **FlexSim** Account

email address

password

[forgot your password?](#)

don't have an account? [sign up](#)

questions or problems? [contact us](#)

Encontre a licença disponível

- Uma vez realizado o log in, vá para a página 'my licenses'.
- Expanda sua pasta padrão do FlexSim para mostrar as licenças disponíveis.

Navigation menu: welcome, download files, **my licenses**, my profile, tools

Buttons: View My Licenses, Add HASP Licenses, Share My Licenses

Activation IDs: Refresh Table, Upgrade Licenses

software/activation ID <small>[hover:Transferable/NodeLocked]</small>	v	owner	seats available / total <small>[hover:install type]</small>	expires M=maintenance T=timed
Flexsim Standard			14 / 17	

Você pode ver o número de assentos (licenças) disponíveis no lado direito, assim como o dia que expira a licença e o contrato de manutenção.

Ativando a licença

- Copie o Activation ID para uma licença disponível. O Activation ID iniciará com flexsim.com.
- Cole o Activation ID dentro do campo Activation ID.
- Pressione o botão **Activate**.

License Activation

Activate | Return | Repair | License Server | View Licenses | Upgrade Licenses

Use this page to activate a license, either from Flexsim or from your company's license server, and permanently store it on this computer.

Activation ID

Use Expiration Date

Activate through Flexsim

Activate through your company's license server

License Server Address

Activate

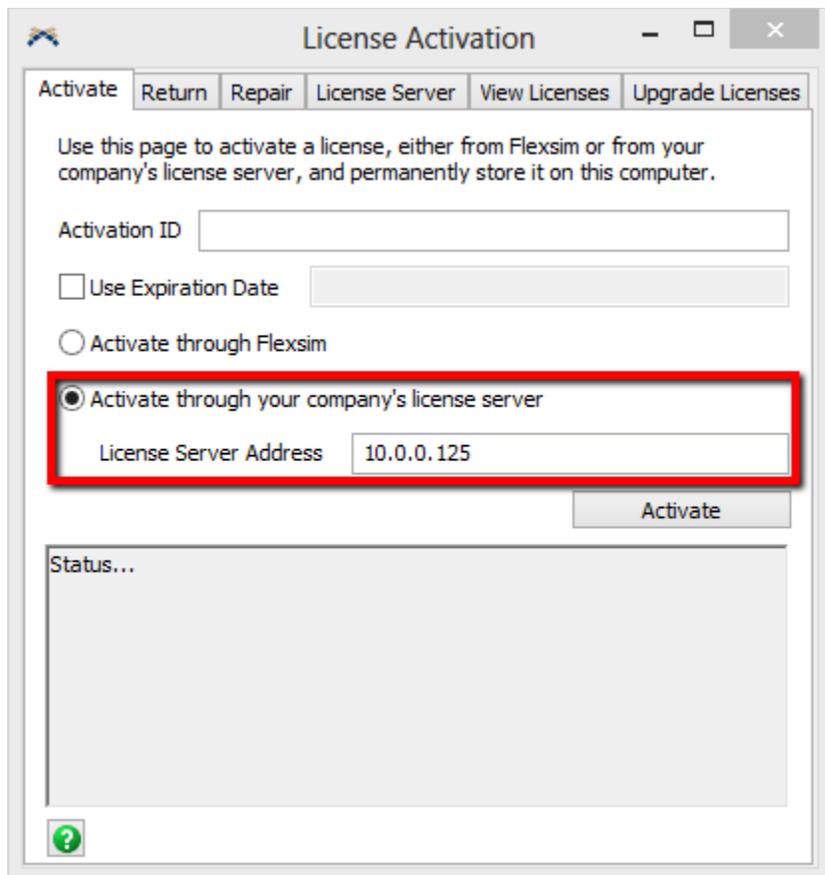
Status...

Nota: Se a sua licença é timed, então você precisará verificar a data que irá expirar e entrar com a data no formato 23-NOV-2009 antes de ativar.

Para uma explanação sobre os erros que aparecem enquanto ativa as licenças, visite a página de referência.

Ativando sobre o servidor de licenças das empresas

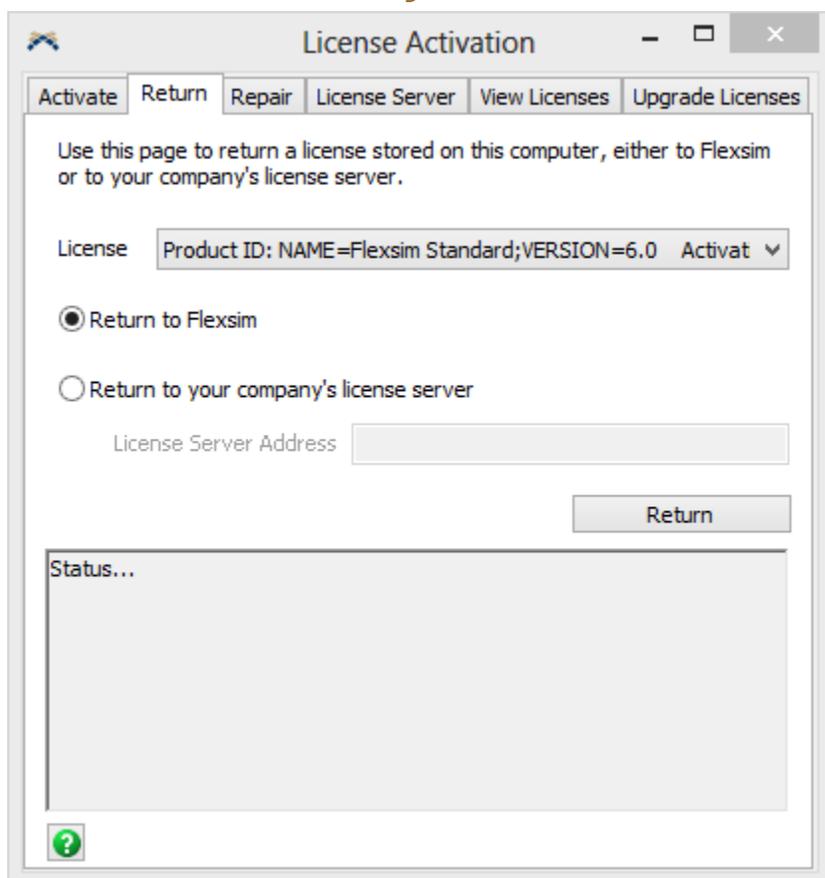
- Na guia Activate da janela License Activation, check o botão **Activate through your company's license server**.
- Entre com o endereço de ip do servidor de licença de sua empresa no campo **License Server Address**.
- Pressione o botão **Activate**.



The screenshot shows a window titled "License Activation" with several tabs: "Activate", "Return", "Repair", "License Server", "View Licenses", and "Upgrade Licenses". The "Activate" tab is selected. Below the tabs, there is a text box with the instruction: "Use this page to activate a license, either from Flexsim or from your company's license server, and permanently store it on this computer." There are three radio button options: "Use Expiration Date" (unchecked), "Activate through Flexsim" (unchecked), and "Activate through your company's license server" (checked). Below the selected option is a text field labeled "License Server Address" containing the value "10.0.0.125". An "Activate" button is located to the right of this field. At the bottom of the window, there is a "Status..." area and a help icon (question mark in a green circle).

Para uma explicação dos erros visualizados enquanto ativa as licenças, visite a página de referência.

Retornando as licenças



The screenshot shows a window titled "License Activation" with a tabbed interface. The "Return" tab is selected. The window contains the following elements:

- Navigation tabs: Activate, Return, Repair, License Server, View Licenses, Upgrade Licenses.
- Instructional text: "Use this page to return a license stored on this computer, either to Flexsim or to your company's license server."
- License selection: A dropdown menu labeled "License" with the text "Product ID: NAME=Flexsim Standard;VERSION=6.0" and a small "Activat" label with a downward arrow.
- Radio buttons: Two options are available: "Return to Flexsim" (which is selected) and "Return to your company's license server".
- Text input: A field labeled "License Server Address" is present below the second radio button.
- Action button: A "Return" button is located to the right of the "License Server Address" field.
- Status area: A large rectangular area labeled "Status..." is at the bottom of the main content area.
- Help icon: A green question mark icon is located in the bottom-left corner of the window.

Retornando uma licença padrão

- Selecione a licença que você quer que retorne do menu drop down **License**.
- Pressione o botão **Return**.

Retornando as licenças do servidor de sua empresa.

- Check o botão **Return to your company's license server**.
- Entre com o endereço de ip do servidor de licença de sua empresa no campo **License Server Address**.
- Pressione o botão **Return**.

Para uma explanação dos erros visualizados enquanto ativa as licenças, visite a página de referência.

Reparando uma Licença

The screenshot shows a window titled "License Activation" with a tabbed interface. The "Repair" tab is selected. The window contains the following elements:

- Navigation tabs: Activate, Return, Repair (selected), License Server, View Licenses, Upgrade Licenses.
- Instruction: "Use this page to repair a license that has been corrupted."
- License dropdown menu: "No Corrupted Licenses on this Computer".
- Radio button: "Repair through Flexsim" (selected).
 - Text: "To repair through Flexsim, you must generate an xml repair request file, email that file to support@flexsim.com, then process the xml response file that is returned."
 - Buttons: "Generate XML Request..." and "Process XML Response..."
- Radio button: "Repair through your company's license server".
 - Text: "License Server Address" followed by an empty input field.
 - Button: "Repair"
- Status area: A large text box labeled "Status..." which is currently empty.
- Help icon: A green question mark icon in the bottom left corner.

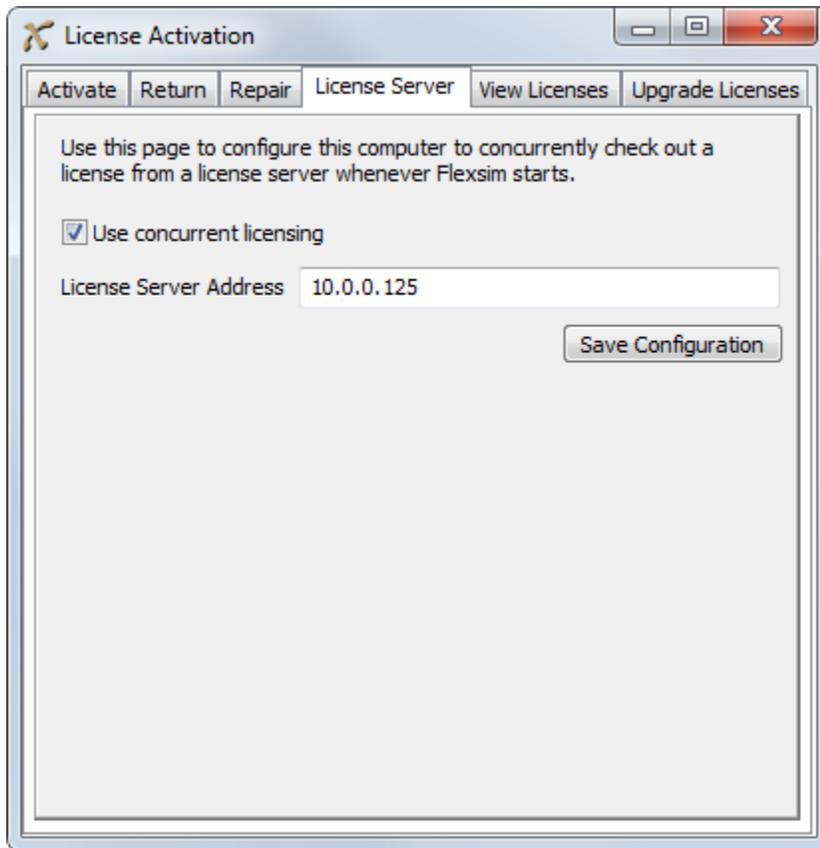
Reparando uma licença padrão

- Selecione a licença corrompida do menu drop down **License**
- Pressione o botão **Generate XML Request...**
- Envie o arquivo gerado para o e-mail support@flexsim.com
- Uma vez que o time de suporte da FlexSim enviar o arquivo de resposta XML, pressione o botão **Process XML Response...** e selecione o arquivo de resposta.
- Pressione o botão **Repair**.

Reparando as licenças do servidor de sua empresa.

- Check o botão **Repair through your company's license server**
- Entre com o endereço de ip do servidor de licença de sua empresa no campo **License Server Address**
- Pressione o botão **Repair**.

Ativando uma licença Concorrente



- Check o botão **Use concurrent licensing**.
- Entre com o endereço de ip do servidor de licença de sua empresa no campo **License Server Address**
- Pressione o botão **Save Configuration**.

Referência na Ativação das Licenças

Tópicos

- Erros de ativação comuns
- Erros de retorno comuns
- Solucionando problemas e erros das licenças de servidor concorrente.

Erros de ativação comuns

Operations Error 7288

```
Status...
Status: 4 Creating request
Status: 5 Request created
Status: 6 Context created
Status: 7 Connected to remote server
Status: 8 Request Sent
Status: 9 Polling for response
Status: 10 Waiting for response
Status: 9 Polling for response
Status: 11 Done
Operations error: 7288 The activation of the fulfillment is denied by the activation policy because fulfill count exceeded the available seat count.
```

Operations error 7288 significa que sua licença já foi ativada em algum computador. Para ativar a licença em sua máquina, você precisa encontrar o computador que está com a licença ativada e retorná-la. Você pode encontrar o computador que está usando a licença, ao clicar em checking the output of View Licenses em qualquer computador que você possa ter ativado a licença.

Error 50041

```
Status...
Status: 4 Creating request
Status: 5 Request created
Status: 6 Context created
Status: 0 Error
ERROR: fixActAppActivationSend - (50041,41143,34)
"Failed to connect to the license server or Operations server.
Recovery: check connection and that server is operational."
```

Error 50041 geralmente significa que seu computador não está conectado na internet ou na sua rede/firewall configurações não estão permitindo a comunicação com o servidor de licenças da FlexSim. O mecanismo de ativação das licenças usa o mecanismo de ativação através de solicitações enviadas a http porta 80. Tenha certeza que esta comunicação esteja disponível em sua rede.

Erros de Retorno Comuns

Operations Error 7466

```
Status...  
Status: 4 Creating request  
Status: 5 Request created  
Status: 6 Context created  
Status: 7 Connected to remote server  
Status: 8 Request Sent  
Status: 9 Polling for response  
Status: 10 Waiting for response  
Status: 9 Polling for response  
Status: 11 Done  
Operations error: 7466 The return of the fulfillment is denied by the return policy because max return exceeded
```

Operations error 7466 significa que a opção de retorno está desativada em sua licença. Sua licença está ativada para ser one-time activation dentro de um computador. Se você acredita que sua licença deve estar apta a ser transferida de um computador para outro, contate o seu distribuidor local ou escritório da FlexSim para discutir este ponto.

Troubleshooting Concurrent Server Errors

1. Verifique se o computador do cliente está conectado ao servidor em uma rede local. Tenha certeza que você pode efetivar o ping no servidor de endereço especificado.
2. Certifique-se de que o servidor tem licenças válidas usando o menu **Tools > View licenses rights** no programa flexsimserveractutil.exe do servidor. Este programa está contido nas ferramentas de Licenciamento LAN disponíveis para download no site da Flexsim.
3. Certifique-se que programa server's License Manager (Imadmin ou Imtools) está devidamente configurado e tem licenças disponíveis para o uso.
4. Consulte o Flexnet License Administration Guide para obter detalhes adicionais sobre como o servidor de licenças FlexNet trGuialha. Este manual pode ser encontrada em http://www.globes.com/support/fnp_utilities_download.htm.
5. Se você ainda continuar tendo problemas, contate o Suporte da Flexsim. Certifique-se de enviar-lhes a informação que é impressa quando você pressionar Tools > View License Rights no programa flexsimserveractutil.exe e também uma tela do Dashboard no Imadmin (ou Status Enquiry output da Imtools).(ou a saída do **Status Enquiry do Imtools**).

Conceitos do FlexSim

1. Overview
2. Flowitems
3. Portas
4. Itemtype
5. Labels
6. Item e Current
7. Return Values
8. Picklists
9. Template Code
10. Model Tree View

Overview dos conceitos do FlexSim

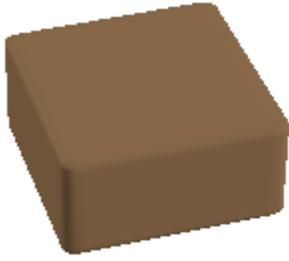
Este tópico descreve em detalhes, os conceitos que são essenciais para a compreensão de como desenvolver modelos com o Flexsim. Você deve ter passado pelo getting started e tutorial fornecido com este guia antes de ler este tópico. Neste tópico não vamos construir nenhum modelo específico. Isso nos permite focar exclusivamente sobre os conceitos que estão sendo discutidos, em vez de passar o tempo focado nas etapas de construção do modelo. Vamos, no entanto, citar um exemplo de modelo onde os conceitos podem ser aplicados, e sentir-se livre para construir o seu próprio modelo. Se você tem afinidade com este tutorial, você deve ter as habilidades necessárias para construir os exemplos do modelo citado. Se você decidir construir o seu modelo, no entanto, aconselhamos que você leia este tópico primeiramente, e então retorne para construir o modelo. Provavelmente, haverá alguns assuntos no final do tópico que você vai querer compreender antes de construir o modelo.

Introduzindo... FlexScript

Não deixe o nome assustar você – **FlexScript** é uma linguagem de comandos script poderosa e fácil de usar que irá ajudá-lo e aliviar sua vida mesmo nos modelos mais complexos. Unicamente da linguagem das soluções do software da FlexSim, Flexscript fornece uma abordagem direta para permitir aos usuários uma customização rápida das triggers e parâmetros dentro dos projetos de simulação.

Através das seções de conceitos, iremos incluir trechos de programação/codificação que ajudam a esclarecer o conceito do que está sendo discutido. A lógica que o exemplo implementa utilizando código pode ser feito de várias maneiras, usando as opções de escolha das funções pré-programadas drop-down ou usar exemplos de código diretos com o flexscript. Se você ainda não estiver familiarizado com flexscript, então você pode pular os exemplos com os códigos e seguir adiante, no entanto, tentaremos passar uma descrição concisa do que as amostras de código podem fazer, para que você venha a ser capaz de entender o que está acontecendo, mesmo se você estiver pouco familiarizado com o flexscript. Se você acabou de fazer os tutoriais e não aprendeu ainda como editar diretamente o código de um campo, então leia sobre o tópico. Para mais informações sobre como escrever código flexscript consulte o tópico no Flexsim

Flowitems



Flowitems são as entidades (produtos por exemplo) que se movem através de seu modelo. Flowitems pode representar peças, paletes, conjuntos, papel, embalagens, pessoas, telefonemas, pedidos ou qualquer coisa que se move através do processo que você está simulando. Flowitems podem ter processos realizados sobre eles e podem conduzidos através do modelo por recursos de manuseio de materiais. No Flexsim, flowitems são gerados pelo Source. Uma vez flowitems passaram pelo modelo, eles são enviados para um objeto de saída do modelo, o Sink.

Portas

Cada objeto do Flexsim tem um número ilimitado de portas através das quais eles se comunicam com outros objetos. Existem três tipos de portas: de entrada, saída e central.

Portas de entrada e de saída são usados no encaminhamento de flowitems. Por exemplo, um selecionador de pacotes de correio coloca o pacote sobre uma das várias esteiras transportadoras de acordo com o destino do pacote. Para simular isso no Flexsim, você iria ligar a porta de saída de um processador (neste caso, o selecionador de pacotes) com a porta de entrada várias esteiras transportadoras, ou seja, uma vez que o processador (selecionador de pacotes) tenha concluído o processamento do flowitem (ou pacote), ela envia para uma esteira transportador.

Portas centrais são usados para criar referências de um objeto para outro. Um uso comum para as portas centrais é fazer referência a objetos móveis, tais como operadores, empilhadeiras, guindastes e de recursos fixos, como máquinas, filas ou transportadores.

Criando portas

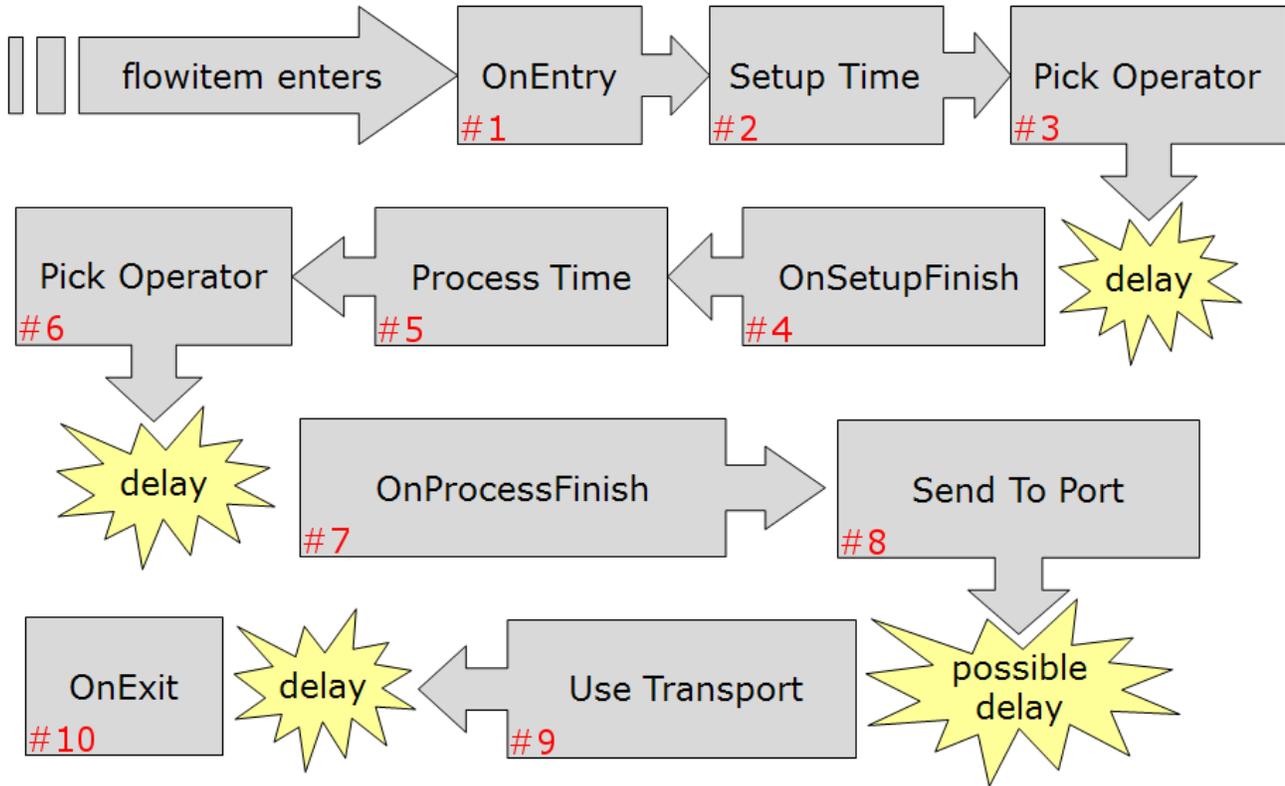
Portas são criadas e conectados por uma das duas opções Guiaixo:

1) Ao clicar em um objeto e arrastar na direção de um segundo objeto, mantendo pressionada diferentes letras no teclado. Se a letra 'A' é pressionada enquanto clica-e-arrasta, uma porta de saída será criado no primeiro objeto e uma porta de entrada será criada no segundo objeto. Estas duas novas portas serão, então, conectados automaticamente. Mantendo pressionada a tecla 'S' vai criar uma porta central, em ambos os objetos e ligar as duas novas portas centrais. As ligações são quebradas e portas eliminadas mantendo a tecla 'Q' pressionado para as portas de entrada e saída e a tecla 'W' para as portas centrais. A tabela logo Guiaixo mostra as letras do teclado usadas para fazer e quebrar os dois tipos de conexões de porta. O exercício 1 do tutorial, mostra como fazer estas conexões.

	Output – Input (porta de saída – porta de entrada)	Central
Conectar	A	S
Desconectar	Q	W

2) Entrando no modo de conexão ao clicar no botão . Uma vez no modo de conexão, há um par de maneiras de fazer uma conexão entre dois objetos. Você também pode clicar em um objeto, em seguida, clique em outro objeto, ou você pode clicar e arrastar de um objeto para o próximo como no método anterior. De qualquer forma, tenha em mente que a direção do fluxo de uma conexão é dependente da ordem em que você faz a conexão. Fluxo vai do primeiro objeto para o segundo objeto. Aliás, as ligações podem ser quebradas clicando no botão  e em seguida, clicando ou arrastando a partir de um objeto para outro, da mesma maneira como quando conectando os objetos.

Ordem de Eventos



Conforme as entidades (flowitems) se movem através do modelo, eles entram e saem dos objetos que chamamos de Fixed Resource. Cada entidade (flowitem) irá seguir o caminho similar ao mostrado acima. Este é caminho é para um objeto Processor.

Exemplo

Considere uma entidade (flowitem) entrando em um processador. As seguintes imagens mostram onde cada trigger e picklist podem ser acessadas através da janela de propriedades do processador. Algumas destas informações podem ser acessadas através do painel Quick Properties ao lado direito de sua tela.

Processor	Breakdowns	Flow	Triggers	Labels	General
OnReset					+ X [icon]
OnMessage					+ X [icon]
OnEntry			# 1		+ X [icon]
OnExit			# 10		+ X [icon]
OnSetupFinish			# 4		+ X [icon]
OnProcessFinish			# 7		+ X [icon]
Custom Draw					+ X [icon]

Processor Breakdowns Flow Triggers Labels General

Maximum Content Convey Items Across Processor Length

Setup Time Use Operator(s) for Setup Number of Operators
 Use Setup Operator(s) for both Setup and Process

Process Time Use Operator(s) for Process Number of Operators

Pick Operator Priority Preemption

Processor Breakdowns Flow Triggers Labels General

Output

Send To Port Use Transport Priority Preemption
 Reevaluate Sendto on Downstream Availability

Input

Pull Strategy Pull Requirement

Etapa 1: OnEntry

Quando a trigger Entry dispara, a entidade já se moveu para dentro do objeto processor, então o conteúdo de cada Processor aumentou em uma unidade. Código pode ser adicionado para a trigger OnEntry para customizar o que ocorre quando o flowitem entra.

Etapa 2: Setup Time

O próximo segmento de código a ser disparado é o tempo de setup ou Setup Time. Este campo com opções pré-programadas dá a você a opção de definir o número da unidade de tempo no modelo que leva a entidade (flowitem) para ser “preparado” no Processor.

Etapa 3: Pick Operator

Se a caixa “Use Operator(s) for Setup” for marcado, o código “Pick Operator” será disparado, chamando o objeto Task Executer. Existe um atraso associado com a chamada do operador, tendo em vista que o operador precisa de movimentar da localização atual para o Processor. O atraso pode aumentar e se estender se não existir operadores disponíveis quando o código Pick Operator é disparado.

Etapa 4: OnSetupFinish

Uma vez que o setup estiver completo, a Trigger Setup Finish irá disparar.

Etapa 5: Process Time

Similar ao tempo de setup, este campo de opções pré-definidas permite a você definir o tempo de processamento ou Process Time para a entidade.

Etapa 6: Pick Operator

Se o "Use Operator(s) for Process" for marcado, o Pick Operator irá disparar. Se os operadores forem usado para setup e processamento, e o "Use Setup Operator(s) for both Setup and Process" estiver desmarcado, então esta opção pré-programada irá disparar duas vezes, uma para chamar o operador para setup e novamente para chamar o operador para processamento. Caso contrário, esta opção pré-programada será chamada somente uma vez.

Etapa 7: OnProcessFinish

Uma vez que o setup estiver completo, a trigger Process Finish será disparada.

Etapa 8: Send to Port

Neste momento, a entidade está pronta para deixar o Processor, ou objeto Fixed Resource. O Send To Port irá disparar e tentar encontrar um destino para a entidade (flowitem). O atraso possível acontece se o objeto não for possível de encontrar um destino disponível para a entidade (flowitem). Isto tipicamente acontece quando os próximos objetos estiverem cheios e não tem espaço disponível para aceitar a entidade. Pode ser também porque o próximo FixedResource ou objeto está parado, ou suas portas de entrada estão fechadas.

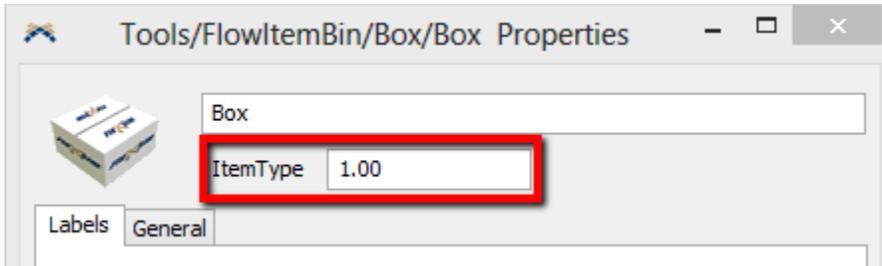
Etapa 9: Use Transport

Se o "Use Transport" está marcado, e uma vez que o destino tem sido a base usando o Send to Port, a opção pré-programada Use Transport será disparada para chamar um Task Executer para o Processor para transportar o item do Processor para seu destino. Novamente, irá resultar em um atraso enquanto um operador disponível é encontrado e se movimenta para o Processor.

Etapa 10: OnExit

Finalmente, a trigger Exit dispara. Quando a trigger dispara, a entidade ainda não saiu do Processor, então o conteúdo do processor permanece inalterável. Uma vez que esta trigger estiver completa, a entidade irá se movimentar para fora do Procesor (reduzindo seu conteúdo em 1 unidade) e ir para um Task Executer ou um próximo objeto (FixedResource). Se você necessitar de um código para ser executado quando um item deixa um FixedResource, mas ele precisar ser executado imediatamente após a saída do item do objeto, então chame o comando `senddelayedmessage()` com um tempo de atraso ou delay time de 0.

Itemtype



Itemtype é o primeiro conceito essencial que nós iremos aprender. É o valor que está estocado (inserido) sobre cada flowitem que viaja através do modelo do FlexSim. Este valor pode ser acessado ou alterado em qualquer ponto dentro de progresso do flowitem através do modelo. Cada flowitem simples tem seu próprio valor/característica de itemtype, isso significa que se você alterar o itemtype de um flowitem, irá alterar somente o valor para aquele específico flowitem, e o itemtype do outro flowitems não será alterado. O significado para o valor do itemtype é de competência do modelador. Em geral, significa que o valor descreve um produto, tipo ou categoria.

Exemplo

Por exemplo, uma loja dos correios, onde clientes vão para enviar suas correspondências ou tirar cópias (Xerox). Neste modelo, os flowitems, ou clientes, estão separados em duas categorias, aqueles que querem enviar correspondências e aqueles que querem obter cópias de documentos ou contratos. O valor do itemtype pode ser usado para fazer esta distinção. Por exemplo, clientes ou flowitems, que precisam enviar as correspondências, o valor do itemtype pode ser 1, e os clientes que precisam de cópias, o valor do itemtype pode ser 2. No FlexSim, nós configuramos o valor do itemtype no objeto source, quando o flowitem é criado. Em nosso exemplo, vamos dizer que 60% dos clientes que chegam no posto de atendimento dos correios são aqueles que querem enviar correspondências, e os 40% restantes são aqueles que querem tirar cópias. Para fazer isso no FlexSim, nós temos que ir até a Guia trigger e em Exit, adicionar o seguinte commando:

```
setitemtype(item, bernoulli(60, 1, 2));
```

```
Source1 - OnExit*
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4
5 setitemtype(item, bernoulli(60, 1, 2));
```

Alternativamente, existe uma opção de picklist disponível para configurar o itemtype ao pressionar o  próximo da trigger OnExist e selecionando Set Type, Name ou Label, ou *Set Item Type and Color*.

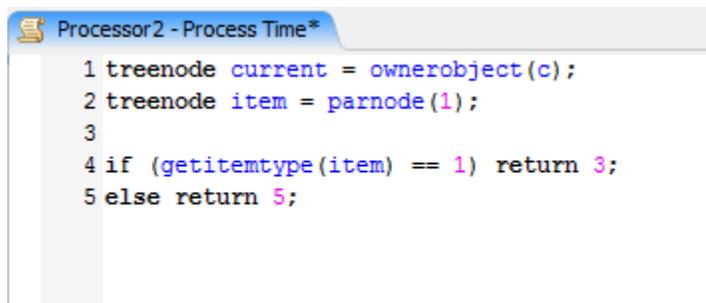
Este comando irá aleatoriamente configurar o valor do itemtype do flowitem que está saindo para 1 em 60% do tempo e 2 em 40% do tempo. O comando **setitemtype** configura o valor do itemtype de um objeto. Ele leva dois parâmetros. O primeiro parâmetros faz referência ao flowitem que você quer configurar o valor do itemtype, e o segundo parâmetros é o valor a ser definido para ele. Neste exemplo, o primeiro parâmetro é o item, ou o flowitem que está saindo do source, visto que estamos trabalhando no Source na trigger OnExit (a referência "item" será discutida em maiores detalhes mais tarde). Perceba que para o segundo parâmetro setitemtype, o comando **bernoulli** é usado. Este commando leva três parâmetros e retorna um dos dois possíveis valores. O primeiro parâmetro é um valor de porcentagem entre 0 e 100. O segundo e o terceiro parâmetros são "sucesso" e "falha", e representa os dois possíveis valores a serem retornados. Neste caso, 60% do tempo o commando Bernoulli irá retornar a 1 (parâmetro 2) e o restante do tempo ele irá retornar a

2 (parâmetro 3). Visto que o comando Bernoulli é o segundo parâmetro do comando setitemtype, o valor do itemtype será definido como tudo que é retornado pelo comando bernoulli, ou seja, o valor de 1 ou 2. A trigger OnExit do Source é executado toda hora que um flowitem é criado e sai do source. Isto significa que o comando setitemtype será executado várias vezes durante a realização da simulação, e cada execução estará associada com exatamente um flowitem. Items saindo do Source irão, assim, ser divididos em 60/40 para os itemtypes 1 e 2, respectivamente. Desde que o comando Bernoulli é estocástico, ou aleatório, não vai sempre manter uma relação perfeita 60/40. Você pode ter várias chegadas de clientes consecutivos cujo valor do itemtype é definido como 1, e vice-versa. Mas com o tempo a relação vai equalizar para 60/40.

Uma vez inicializado o valor do itemtype, a lógica em seu modelo pode então ser implementado baseado no valor do itemtype para cada cliente passando pelo modelo. No exemplo, o cliente do "envio de correspondência" pode levar 3 minutos para ser atendido, enquanto o cliente que quer fazer um cópia, pode levar 5 minutos para ser atendido. No FlexSim, você irá implementar esta diferença escrevendo um código flexscript no campo "process time" do objeto **processor**. O código deve parecer-se algo como abaixo:

```
if(getitemtype(item)==1) return 3;
```

```
eles return 5;
```



```
1 treenode current = ownerobject(c);  
2 treenode item = parnode(1);  
3  
4 if (getitemtype(item) == 1) return 3;  
5 else return 5;
```

Este código basicamente diz: se o valor do itemtype do flowitem que está atualmente sendo atendido (getitemtype(item)) é igual a 1 (é o cliente "de envio de correspondência"), então retorne 3 como tempo de processamento (process time). Caso contrário (é o cliente "cópia") retorne 5 como process time.

Mais uma vez, este exemplo pode ser feito sem escrever qualquer código, utilizando as funções pré-programadas dos pick-lists do Flexsim para definir a sua lógica. No entanto, o conceito-chave para entender aqui é que cada flowitem pode ter um valor itemtype cujo significado é definido por você, e que você pode usar o valor do itemtype para tomar decisões no seu modelo.

Nota sobre os objetos no FlexSim: Todo flowitem em um modelo tem um itemtype. No entanto, objetos no FlexSim como Sources, Queues e Processors não possuem um itemtype

Nota sobre o valor do itemtype: O itemtype é um número de ponto flutuante de precisão dupla. Isto significa que o itemtype não pode manter somente valores inteiros como, 1,2,3, etc., pode também manter valores como 1.5 ou 99.9. No entanto, o itemtype não pode manter valores string como "package".

Nota sobre a aparência do flowitem: O valor do itemtype não irá definir a aparência visual do flowitem. Isto pode ser configurado ao escolher a flowitem class na janela do Source's Parameters, como

caixa	ou	pallet.
-------	----	---------

Labels

Os Labels são também um conceito-chave para compreender a construção de modelos no Flexsim. Eles são muito semelhante ao valor do itemtype, que armazenam dados de objetos que podem ser utilizados durante a tomada de decisão do modelo. No entanto, existem algumas diferenças fundamentais, listadas abaixo:

- Cada label tem um nome que é definido por você modelador.
- Ao contrário do itemtype, que é específico para flowitems, Labels podem ser definidas para objetos, assim como para flowitems, por exemplo, em uma Source, Queue ou processor (equipamento).
- Um objeto pode ter tantas Labels quanto você quiser atribuir a ele.
- As Labels podem ter número ou strings values, enquanto um itemtype só pode ter um valor numérico. As Labels podem até mesmo carregar listas ou tabelas de valores.
- Com as Labels, você deve adicionar explicitamente a label para o objeto através de sua janela de propriedades, ao contrário do valor itemtype, que está automaticamente incluído em cada flowitem.
- Ao adicionar uma Label a um flowitem no Flowitem Bin, a Label é específica para essa classe de flowitem. Isto significa que, se você adicionar uma Label para a classe flowitem Pallet, apenas flowitems que são criados a partir dessa classe Pallet terá essa label sobre eles.

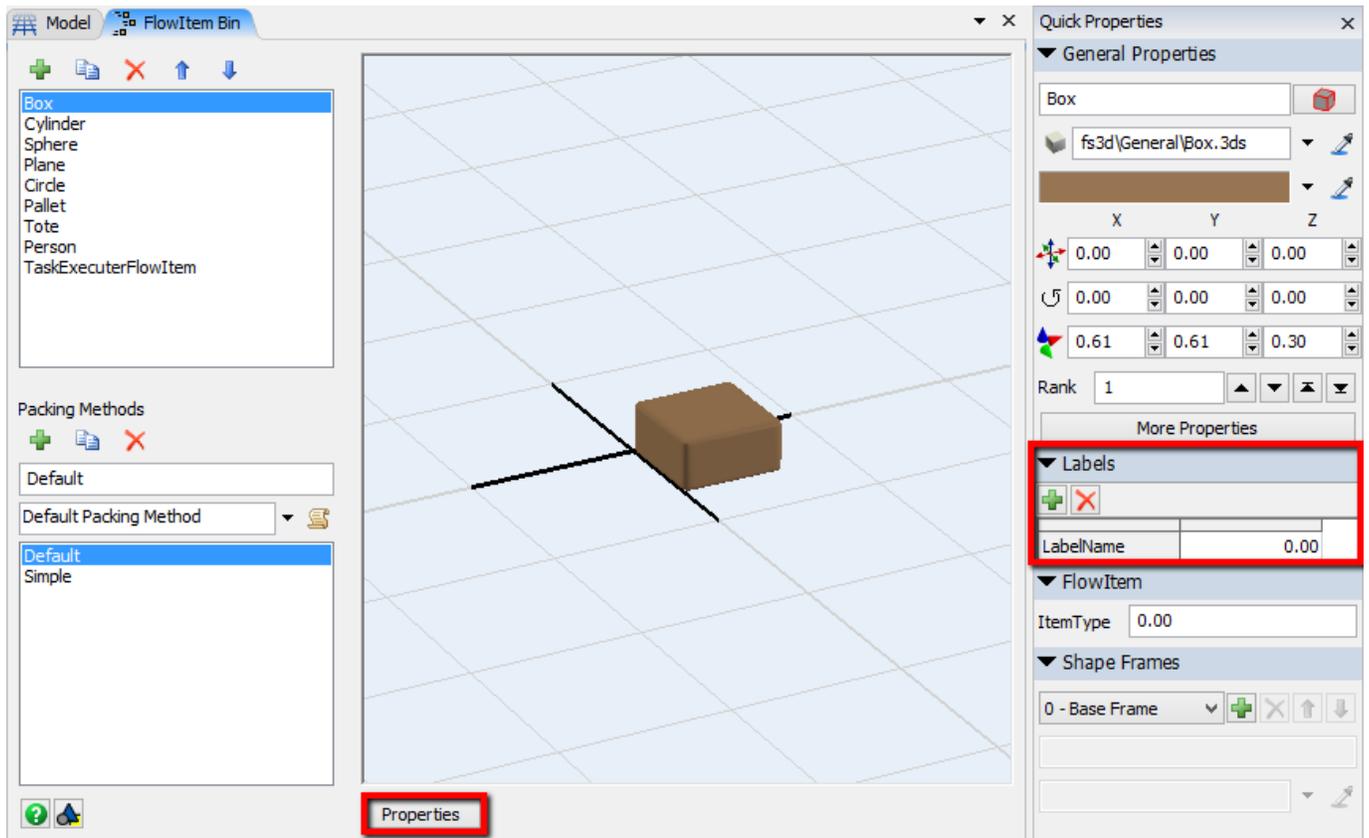
Values

Para flowitems, o valor especificado na janela Properties será o valor da Label padrão para todos os flowitems que são criadas, mas você pode alterar esse valor em cada flowitem à medida que progride através de seu modelo. Para as labels dos objetos do FlexSim, os valores das labels permanecem as mesmas a menos que você tenha uma lógica dentro do objeto que altere o valor da label.

Para Labels dos objetos do FlexSim, o valor da Label permanecerá o mesmo, a menos que você check o botão **Automatically Reset Labels** na guia label. Alternativamente, você pode adicionar código de programação ou opção de picklist na guia Trigger em OnReset para resetar o valor da label. Isso permite que você defina a Label de volta para um valor inicial quando você pressiona o botão Reset para reiniciar a execução do modelo.

Exemplo

Para adicionar Labels para flowitems, vá para o FlowItem Bin, selecione a classe flowitem que está sendo criado pelo seu Source (caixa neste exemplo), e pressione o botão Properties. Vá até a guia Labels e adicione *strings* ou *number labels* usando os botões na parte inferior do painel. Para adicionar Labels aos objetos no Flexsim, clique com o botão direito do mouse sobre o objeto e na janela ortho, selecione a opção *Properties*. Especifique o nome de cada Label na coluna row da linha à esquerda, e seu valor a direita de seu nome.



O processo funciona de forma similar para outros objetos do FlexSim. Você pode editar suas labels através da janela Quick Properties, ou dar um duplo clique sobre o objeto e editar na guia labels. Especificamente, cada nome de label é definido nas linhas na coluna da esquerda e o seu valor ao lado direito de seu nome. Vamos estender o modelo de exemplo mencionado anteriormente para fazer uma explanação sobre Labels. Digamos, por exemplo, que cada "cliente que deseja tirar cópia" tem um certo número de cópias que precisam ser feitas, e que, o tempo de atendimento para aquele cliente é dependente do número de cópias necessárias para o atendimento. Um cliente que precisa de 1.000 cópias levará mais tempo para ser atendido do que um cliente que precisa de uma cópia. Como antes, o valor de itemtype para cada flowitem, ou cliente, refletia a categoria do cliente, ou "que deseja enviar um pacote ou correspondência" ou "que deseja tirar cópia", mas agora para "clientes que desejam tirar cópia", precisamos adicionar uma Label que nos diz quantas cópias o cliente necessita. Mais uma vez, para adicionar uma label a um flowitem, vá para o Flowitem Bin, e em seguida, selecione a classe flowitem, vá para Propriedades e, na Guia Labels. Então, deve-se adicionar uma Label number e dar-lhe um nome como "**nrofcopies**". Como valor padrão podemos deixá-lo em 0 e configurar o valor em **Exit Trigger** que encontra-se nas propriedades de Source, especificamente na Guia Trigger.

Uma vez que a Label for adicionada na **Flowitem Bin**, podemos configurar o valor da labels de cada flowitem sendo gerado no Source. No exemplo, os clientes que precisam de cópias necessitam de um número aleatório de cópias entre 1 e 1000. Para implementar isto, nós teremos que realizar modificações na Trigger OnExist do Source, conforme demonstrado abaixo:

```
setitemtype(item, bernoulli(60,1,2));
```

```
if(getitemtype(item)==2)
```

```
    setlabelnum(item, "nrofcopies", duniform(1,1000));
```

Como descrito anteriormente, o comando SetItemType define o itemtype do item para uma divisão 60/40 entre 1 e 2. Agora vamos adicionar um commando if. Esta declaração basicamente diz: se o itemtype do flowitem de saída for 2 (que é cliente que deseja realizar cópias), então configure o valor da label do flowitem

chamado "nrofcopies" para um número aleatório entre 1 e 1000. O comando **setlabelnum** define um valor da label e leva três parâmetros. O primeiro parâmetro é o objeto cuja label pretendemos definir (item, ou o flowitem que está saindo). O segundo parâmetro é o nome da label ("cópias"). Este parâmetro precisa estar entre aspas, pois é um parâmetro string. O terceiro parâmetro é o valor que precisa configurar a label para (duniform (1,1000)). O comando duniform retorna um valor a partir de uma distribuição uniforme discreta. Leva dois parâmetros, ou seja, o valor mínimo e máximo, e retorna um número aleatório entre esses dois valores, uniformemente distribuída, ou seja, para cada valor entre o mínimo e o máximo a probabilidade é a mesma de ser retornado qualquer outro valor entre o mínimo e o máximo. A parte "discreta" significa que o comando só vai retornar 1,2,3, etc, em oposição ao comando uniforme (), que pode retornar valores como 1.5 ou 2.5. Desde que, nunca existirá um cliente que precise de 1,5 cópias, usaremos o comando duniform ().

Note que ao adicionar a label "nrofcopies" no Flowitem Bin, cada flowitem que é criado a partir da classe flowitem terá a label "nrofcopies" nele. Mesmo os clientes que desejam enviar correspondência terão esta label, mas nossa lógica simplesmente não vai olhar para a label para verificar se é um cliente que deseja enviar correspondências.

Agora que nós configuramos a label e também determinamos seu valor inicial, podemos definir a lógica para tomar decisões com base no valor da label em nosso modelo. Para um cliente que deseja tirar cópia, por exemplo, podemos mudar o tempo de atendimento com base no número de cópias que o cliente precisa. Para cada cliente que deseja tirar cópia, o tempo de atendimento pode ser a base de cinco minutos, como antes, mais um adicional de cinco segundos para cada cópia que precisa ser feita. Para fazer essa alteração, você precisa acessar o campo "Process time" em propriedades do Processors e mudar para a seguinte forma:

```
if (getitemtype (item) == 1)
```

```
    return 3;
```

```
    eles return 5 + (getlabelnum(item, "nrofcopies")*(5.0/60.0));
```

Como antes, usamos uma instrução if para dar o itemtype 1 (clientes que deseja enviar correspondência) um tempo de atendimento de 3 minutos. Na porção restantes (clientes que desejam tirar cópia), no entanto, voltamos a expressão: $5 + (\text{getlabelnum}(\text{item}, \text{"nrofcopies"}) * (5.0/60.0))$. Esta é a base de tempo de atendimento de 5 minutos, mais o número de cópias que o cliente necessita ($\text{getlabelnum}(\text{item}, \text{"nrofcopies"})$) vezes cinco segundos (5.0/60.0). Lembre-se que informações que definimos para o nosso modelo estão em minutos, por isso, se uma unidade de tempo do Flexsim é igual a um minuto, em seguida, cinco segundos é igual a 5/60 minutos ou unidades de tempo.

Nota sobre o operador de divisão: No exemplo acima usou-se a expressão 5.0/60.0 em vez de 5/60. É importante fazer essa distinção porque C++ vê as duas expressões de divisão de forma diferente. Com 5/60, C++ vê este como o inteiro 5, dividido pelo número inteiro 60. Além dos mais, um número inteiro dividido por um número inteiro, também deve ser um número inteiro, ou 0. Com 5.0/60.0, no entanto, C++ vê-la como a divisão de dois números de ponto flutuante, e além do mais, o resultado é uma fração entre 0 e 1. Por outro lado, o flexscript, que não é fortemente semelhante ao C++, na verdade, interpreta a expressão 5/60 como a divisão de dois números de ponto flutuante, o que significa que seria ótimo usar 5/60 em flexscript. No entanto, nas poucas situações como esta, onde implementação do flexscript desvia da implementação C++, nós recomendamos você a escrever código que é cross-compatível com flexscript e C++, e além do mais a expressão correta seria 5.0/60.0. Para mais informações sobre inteiro vs divisão de ponto flutuante, consulte o tópico sobre como escrever lógica no Flexsim.

Então, novamente, assim como o valor itemtype, podemos usar labels para armazenar dados em flowitems (ou objetos), e então podemos acessar os dados para tomar decisões no modelo.

Item e Current

```
Source1 - OnExit
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
```

Os termos itens e current que são duas variáveis de acesso referem-se a objetos no Flexsim. Quando você editar o código de um determinado campo, você sempre vai ver no topo do código uma ou mais declarações "header". Estas declarações configuram suas variáveis de acesso, e geralmente será algo parecido com o seguinte:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);
```

Para mais informações sobre a porta, veja a seção correspondente.

Exemplo

Neste exemplo, a primeira declaração é o que chamamos de uma variável de declaração. Ela declara uma variável chamada current. O tipo de variável current é um treenode. Esta é uma variável do tipo que mantém uma referência a um objeto na estrutura da árvore (Tree) do Flexsim. Eu não quero entrar em muitos detalhes sobre isso, então, em poucas palavras, todos os dados no Flexsim, incluindo objetos e flowitems, é armazenado como nós (nodes) em uma estrutura de árvore (tree), e o tipo de variável treenode é simplesmente uma referência a um nó (ou objeto), na estrutura de árvore (Tree). Para mais informações sobre a estrutura da árvore, consulte o tópico sobre a estrutura da árvore de Flexsim (FlexSim's tree). A primeira declaração também define o valor desta variável current nomeada como: **ownerobject (c)**. Agora, eu também não quero entrar em muitos detalhes sobre o significado de ownerobject (c) é porque isso pode ser uma explicação complexa de entender neste momento. O essencial aqui é que você tem uma variável treenode (ou objeto de referência) chamada current, e você vai ter que confiar em mim quando disser que current irá sempre apontar para o objeto "atual" que você está editando o campo. Se você entrar na janela de parâmetros de um Source e editar na Trigger OnExit, então, nesse campo, current é uma referência ao objeto Source. Se, por outro lado, você vai para a janela de parâmetros de um objeto como o Processor, e abrir o código para editar o campo Process Time do processor, em seguida, dentro desse campo, current é uma referência a esse objeto Processor.

O código do exemplo também tem uma segunda declaração. A afirmação é outra declaração de uma variável treenode chamada item desta vez, que é dado o valor: **parnode (1)**. Mais uma vez, a fim de não desviarmos a atenção, não iremos explicar o comando parnode, mas apenas dizer que este item sempre vai se referir ao flowitem que está associado com uma execução específica de um campo ou trigger. Por exemplo, se você estiver implementando uma Trigger em OnExit de um Source, então, cada vez que o exit trigger é executado, o item irá se referir ao flowitem que está saindo do Source naquele momento específico. Note-se que a referência item irá mudar a cada vez que a Trigger on Exit for executado porque um novo flowitem está saindo, considerando que a referência current será o mesmo todo o tempo porque o objeto Source não muda.

Então, essas declarações cabeçalho configuram as variáveis de acesso que podem ser usadas, dentro do código. É por isso que nos exemplos anteriores fomos capaz de usar o termo item, por escrito, nos comandos:

```
setitemtype(item, bernoulli(60,1,2));
```

ou:

```
if(getitemtype(item)==2)...
```

por que eu tenho a referência para um flowitem e a referência é chamada item.

Frequentemente as declarações do cabeçalho mencionado acima irão variar de acordo com o tipo de campo que você está escrevendo o código. Por exemplo, as declarações de cabeçalho de uma trigger exit deve ser semelhante a este:

```
treenode current = ownerobject(c);
```

```
treenode item = parnode(1);
```

```
int port = parval(2);
```

Aqui existe uma variável-declaração adicional chamada Port. Neste caso port é o número da porta de saída através da qual item irá sair. Declarações como reset trigger's, por outro lado, será parecido como esta:

```
treenode current = ownerobject(c);
```

Aqui existe a declaração de uma variável, chamada current. Não existe declaração de item. A razão para isso, é porque o reset trigger, o qual é executado quando você pressiona o botão reset, não tem um flowitem específico associado com sua execução.

Este guia documenta cada campo e suas variáveis de acesso nos tópicos e sub-tópicos da seção da lista de opções.

Revisão sobre labels e itemtype

Então, para rever, nos campos de código do Flexsim, você terá acesso as variáveis chamadas current e item. Current fará sempre referência ao objeto cujo código você está editando. Item será sempre referência para flowitem que está associado com a execução específica do campo, por exemplo, o item que estiver saindo de Source. As variáveis de acesso variam de acordo com o tipo de campo, mas você sempre pode encontrar quais variáveis estão disponíveis apenas olhando para as declarações de cabeçalho no topo do seu código, ou consultando o manual do usuário.

Nota sobre a especificação correta dos objetos quando acessar labels ou itemtype: É importante entender qual objeto está mantendo uma label ou itemtype. Por exemplo, no exemplo acima, usamos o comando getlabelnum (item, "nrofcopies"). Nós não usamos getlabelnum (current ", nrofcopies.) A razão pela qual nós usamos item e não current é porque a label é armazenado no próprio flowitem, e não no objeto do Flexsim. Se você adicionar uma label a um flowitem no flowitem bin, então, o item deve ser referenciado com o comando getlabelnum. Por outro lado, se você estiver usando uma label sobre o objeto (você adicionou a label para o objeto através de sua janela de propriedades), então current deve ser referenciado no comando getlabelnum.

Return Values

No Flexsim há uma estreita interação entre o comportamento "sob o capô" dos objetos em seu modelo e na lógica que você implementa nos objetos através de campos do código. Muitas vezes, a razão na qual um campo de código é executado deve-se ao motivo que o objeto do Flexsim (processor, source, etc.) está solicitando dados de você que é o modelador de como ele deve funcionar. Por exemplo, o campo Process Time, do processor é executado porque ele precisa saber de você o tempo de processamento ou atendimento para um dado flowitem. A maneira que você passa a informação correta de volta para o processador é através do valor de retorno desse campo, ou em outras palavras, através da execução de uma instrução de retorno em seu código.

Example

Vamos voltar para o exemplo dos correios mencionado anteriormente. Nesse exemplo, implementamos o código para o process time como:

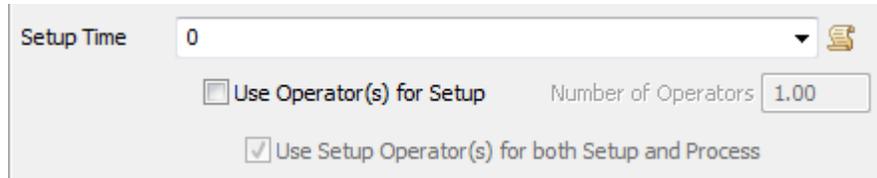
```
if(getitemtype(item)==1) return 3;
```

```
eles return 5 + (getlabelnum(item, "nrofcopies")*(5.0/60.0));
```

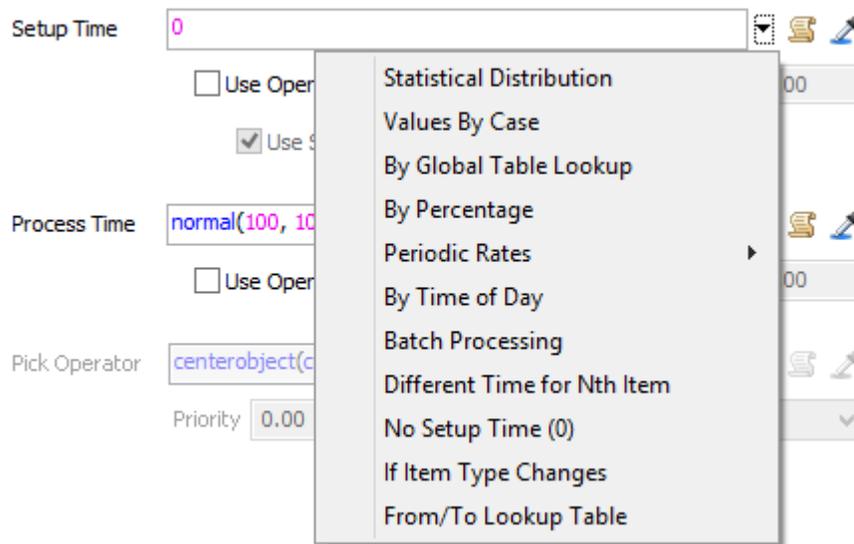
O processador irá executar este campo cada vez que recebe a flowitem, pouco antes de começar o tempo de processamento (process time). Ao executar este campo process time, é essencialmente solicitado o tempo de processamento (ou atendimento) que este item deve ter. Neste código, estamos usando a instrução de retorno para passar de volta para o processador o tempo adequado para o flowitem ser processado. Além do mais, a declaração de retorno é usado para dar ao objeto os dados adequados que ele necessita para operar conforme queremos.

Muitos campos não precisam de um valor de retorno. Por exemplo, no modelo dos correios a trigger exit do Source não inclui uma instrução de retorno. A razão para isso é porque com uma exit trigger o objeto Source não está tentando obter informações de retorno sua, e está simplesmente proporcionando-lhe um lugar onde você pode executar a funcionalidade quando um flowitem sai do Source.

Picklists

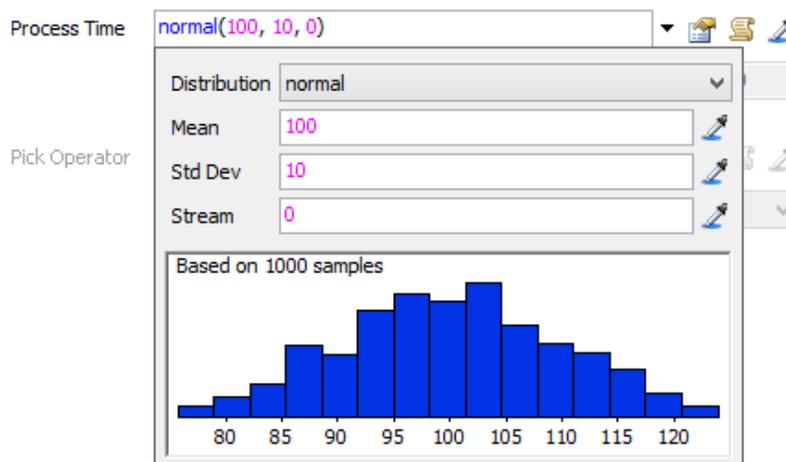


Você encontrará uma infinidade de janelas picklists dentro do FlexSim. Estas janelas oferecem ao usuário uma interface fácil para implementar funcionalidades no FlexSim. Por trás das cortinas, cada uma destas janelas referem-se a um pedaço de codificação. O mais legal sobre estas interfaces picklists é que eles permitem você escrever algumas funcionalidades sem a necessidade de escrever códigos através de programação. Eles fornecem uma lista de funcionalidades usadas comumente quando você clica sobre a caixa drop-down. Você pode também selecionar o texto e substituí-lo por uma constante ou expressão.



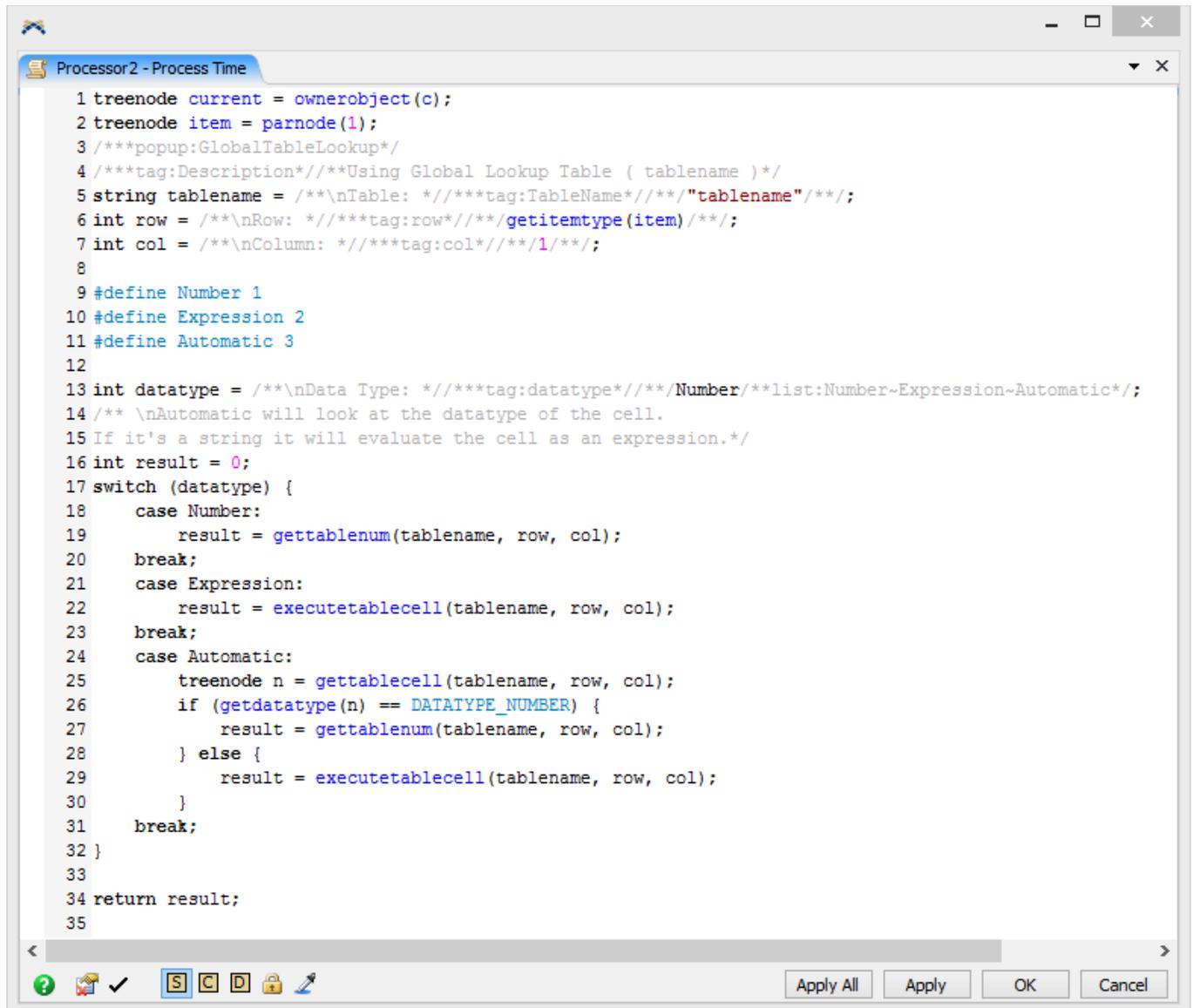
Popups

Cada picklist tem um popup associado para interface com usuário. Os Popups permitem você facilmente editar as opções de parâmetros. Você pode editar estas opções a qualquer tempo ao clicar no botão . Uma vez que você entrou com os valores que você quer no popup, clique em qualquer lugar fora do popup para fechá-la.



Code Edit (Edição de Código)

Modeladores experientes tem também a habilidade de escrever códigos explicitamente quando necessário. Ao clicar no botão , você pode abrir a janela de edição de códigos, no qual você pode ver todo o código implementado neste campo. Note que muitos dos códigos que vocês verão é na verdade usado para formatar a janela do template de código. Você pode diferenciar o código real do código de formatação do template pela cor. Código de formatação do template tem comentários em cinza (veja a página Guiaixo).



```
Processor2 - Process Time
1 treenode current = ownerobject(c);
2 treenode item = parnode(1);
3 /**popup:GlobalTableLookup*/
4 /**tag:Description/**Using Global Lookup Table ( tablename )*/
5 string tablename = /**\nTable: ****tag:TableName****/"tablename"*/;
6 int row = /**\nRow: ****tag:row****/getitemtype(item)/**;
7 int col = /**\nColumn: ****tag:col****/1/**;
8
9 #define Number 1
10 #define Expression 2
11 #define Automatic 3
12
13 int datatype = /**\nData Type: ****tag:datatype****/Number/**list:Number-Expression-Automatic*/;
14 /** \nAutomatic will look at the datatype of the cell.
15 If it's a string it will evaluate the cell as an expression.*/
16 int result = 0;
17 switch (datatype) {
18     case Number:
19         result = gettablenum(tablename, row, col);
20         break;
21     case Expression:
22         result = executetablecell(tablename, row, col);
23         break;
24     case Automatic:
25         treenode n = gettablecell(tablename, row, col);
26         if (getdatatype(n) == DATATYPE_NUMBER) {
27             result = gettablenum(tablename, row, col);
28         } else {
29             result = executetablecell(tablename, row, col);
30         }
31         break;
32 }
33
34 return result;
35
```

Para aprender mais sobre edição de códigos, veja a seção Code Editor page.

Para mais informações sobre como escrever códigos nos FlexSim, veja a parte que fala sobre escrevendo lógica no FlexSim.

Template Code

No campo de edição de códigos você poderá encontrar um texto cinza aparentemente espalhados por todo o código. Por exemplo, você poderá encontrar o seguinte código:

```
/**By Expression*/
/** \nExpression: */
double value = /**/10/**/;
return value;

/** \n\nNote: The expression may be a constant
value or the result of a command (getitemtype(),
getlabelnum(), etc).*/
```

O texto em cinza é o que chamamos template code. Isso permite que você especifique o que irá ser mostrado como texto fixo e/ou editável quando o usuário pressiona o botão . Isto faz o usuário de Flexscript/C++'s comentar a syntax, e então, dar algumas informações sobre o comentando.

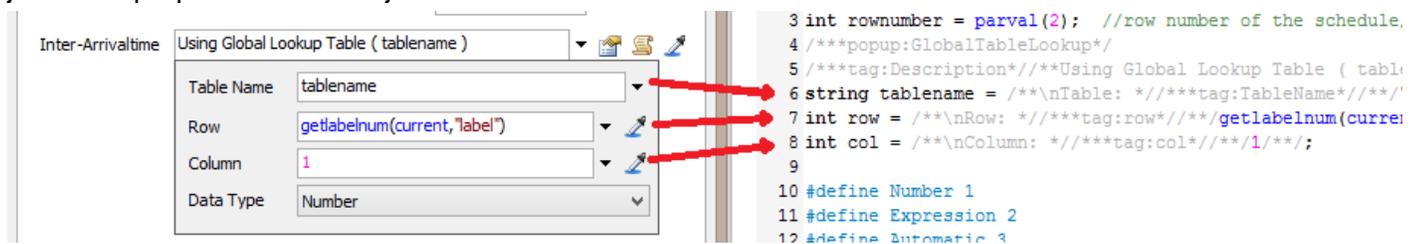
```
/**popup:ValuesByCase:hasitem=1:valustr=Port:doreturn=1*/
```

Este template de código é uma instrução para abrir um popup.

```
int case_val = /**tag:ValueFunc**/**/getitemtype(item)**/**/;
```

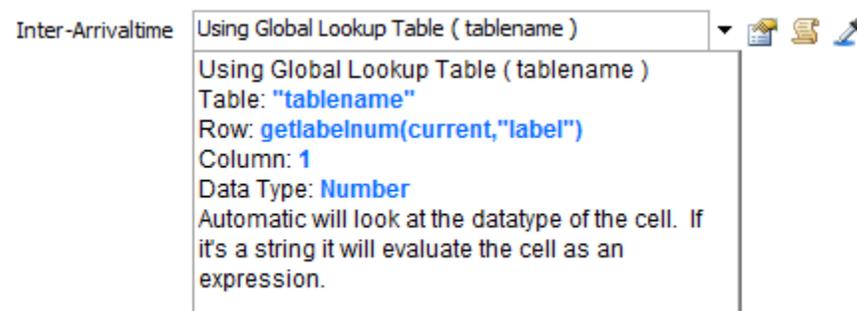
Quando o popup abre, ele analisa o roteiro de instâncias de `/**tag:TagName**/**/Value/**/` e coloca os `Value` no campo apropriado no popup. Quando o popup fecha, os valores de cada campo são colocados em seus respectivos `Value`. Desta forma, o mecanismo cria uma tag `/**tag:` de duplo caminho que linka o script e o popup.

Guiaixo uma tela mostrando o uso da Global Lookup Table do menu drop-down de **Inter-Arrivaltime** na janela de propriedades do objeto **Source**.



The screenshot shows the 'Inter-Arrivaltime' property window with a 'Using Global Lookup Table (tablename)' dropdown menu. The configuration fields are: Table Name: tablename, Row: getlabelnum(current, 'label'), Column: 1, and Data Type: Number. Red arrows point from these fields to the corresponding code in the adjacent window. The code includes: `3 int rownumber = parval(2); //row number of the schedule,` `4 /**popup:GlobalTableLookup*/` `5 /**tag:Description**/**/Using Global Lookup Table (tabl` `6 string tablename = /**\nTable: **/**tag:TableName**/**/'` `7 int row = /**\nRow: **/**tag:row**/**/getlabelnum(curre` `8 int col = /**\nColumn: **/**tag:col**/**/1/**/;` `9` `10 #define Number 1` `11 #define Expression 2` `12 #define Automatic 3`

Template de Popup de Texto



The screenshot shows the 'Inter-Arrivaltime' property window with a 'Using Global Lookup Table (tablename)' dropdown menu. The popup text is: `Using Global Lookup Table (tablename)`
`Table: "tablename"`
`Row: getlabelnum(current, "label")`
`Column: 1`
`Data Type: Number`
Automatic will look at the datatype of the cell. If it's a string it will evaluate the cell as an expression.

O mecanismo de código do template no Flexsim usa comentários para especificar o texto do template do modelo. Lembre-se que o texto do modelo (o texto que pode ser editado no template do tipo drop-down) é dividido em texto fixo em preto e texto azul editável. Em seu código, para especificar uma seção de texto em preto fixo, use um comentário multi-linha, mas adicione um asterisco adicional para marcar o início: `/**`. Ao acrescentar esta asterisco adicional, sinaliza ao interpretador do código do Flexsim que esta é uma seção de texto em preto fixo que deve aparecer quando o usuário olha para o texto do template. No exemplo acima, o texto: `/** By Expression */` torna então o texto: By Expression vai aparecer em texto na cor preta no template drop-down. Agora, para especificar o texto editável em azul. Com texto em azul editável, você quer que a entrada do usuário seja parte do código atual. Então, para começar um texto azul, você entra em um comentário multi-linha, e então imediatamente saia do comentário: `/**`. Você usa a mesma tag para sair de uma seção de texto azul. Além do mais, no código acima, o valor 10 está disponível para mudar o texto quando o texto do template é mostrado:

```
double value = /**/10/**/;
```

Por causa destes comentários, o flexscript somente vê: `double value = 10;` mas a vantagem é que agora você (ou outro modelador) podem rapidamente mudar o valor 10 para qualquer outro valor simplesmente pressionando o botão  e editando o texto do template em azul.

Você também pode perceber que os comentários ocasionalmente incluem a tag `\n`. Esta tag especifica uma nova linha a ser feita no texto em preto do template. Você também pode especificar uma nova linha apenas colocando uma nova linha no comentário, mas muitas vezes você vai querer o seu código de template para ocupar o menor espaço possível, para que o próprio código possa ser visto com mais facilidade.

Comentários

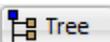
No flexscript, você pode "comentar" uma seção de texto flexscript para que o analisador não considere esse texto como parte do código. Isso permite que você adicione o texto descritivo que explica o que o código faz. Há duas maneiras de fazer comentários. O primeiro é o comentário de uma linha, e é feito com duas barras: `//`. O exemplo a seguir mostra um comentário de uma linha

```
// This is my one-line comment, it ends at the  
end of this line
```

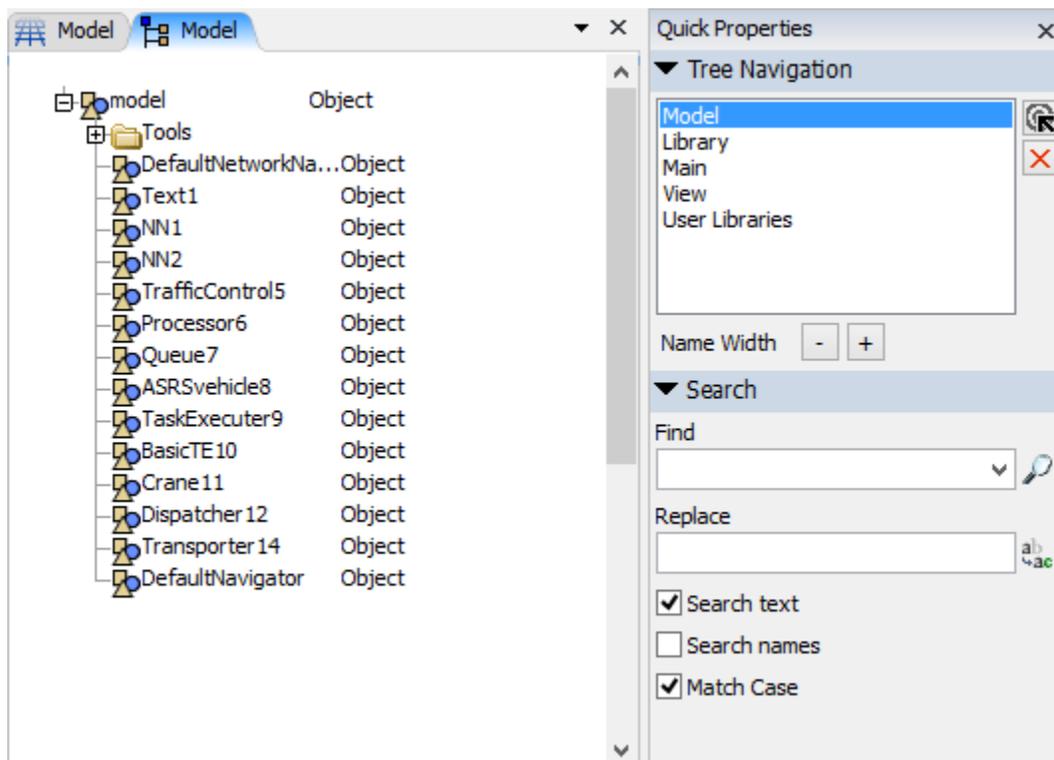
Um comentário de uma linha estende-se até o final da linha do texto. O outro comentário é um comentário de várias linhas. Aqui você pode sinalizar o início do comentário com o texto: `/*` e sinalizar o fim do comando multi-linha com o texto: `*/` como mostrado Guiaixo:/*

```
this is my multi-line comment  
it can span as many lines  
as I want it to  
*/
```

Model Tree View

O “**Model Tree View**” é utilizado no FlexSim para explorar a estrutura do modelo e detalhar a composição dos objetos. Para acessar o “Model Tree View”, clique no botão  na barra de ferramentas. As janelas Tree Edit e Tree Navigation aparecerão na tela (Figure 3-11).

Nota: Caso você esteja usando a versão de avaliação do FlexSim, a opção Model Tree View não estará disponível para utilização.



A janela “Model Tree View” é um local que proporciona diversos recursos únicos. Nesta janela é possível:

- Customizar os objetos do FlexSim utilizando as linguagens C++ ou Flexscript
- Visualizar os dados de todos os objetos
- Acessar a janela de propriedades (janela Properties)
- Editar o modelo, deletar objetos e modificar todos os dados de qualquer objeto

Caso você siga todas as regras de navegação do “tree view”, a tendência é achar esta janela uma das mais versáteis para fazer modificações no FlexSim. A estrutura de dados para qualquer ação feita no FlexSim está no “Tree”. A maioria das janelas editáveis encontradas no FlexSim são apenas Interfaces Gráficas do Usuário (GUI - Graphical User Interfaces) que mostram alguns dados contidos no tree. Partindo da idéia que todas as vistas e todos os objetos no FlexSim trabalham da mesma maneira, uma vez que você entender bem como funciona o tree, você estará apto a navegar e entender como qualquer outro meio de visualização do FlexSim funciona para acessar e alterar/utilizar as variáveis.

Base do Tree View

O FlexSim foi construído de modo que armazena todas as informações em uma estrutura de árvore (tree structure). Esta estrutura no tree é o núcleo da estrutura de dados para qualquer objeto orientado ao

FlexSim. Para aqueles que são familiarizados com a linguagem de programação orientada a C++, quando visualizarem o FlexSim imediatamente o reconhecerá o Tree View como o padrão C++ para objetos orientados ao gerenciamento de dados.

Existem diversos símbolos usados no FlexSim usados para facilitar o entendimento durante a navegação do tree.

Todo o node MAIN no tree view é uma referência para o projeto. A biblioteca e o modelo fazem parte do PROJECT. O node VIEW do tree contém todas as definições relacionadas a visualização e os GUIs dos objetos assim como todas as janelas que estão abertas. Quando uma sessão é salva todo o node MAIN e VIEW também são salvos juntos.

 O ícone de uma pasta representa que este node não possui nenhum objeto em seu interior, mas pode conter outros nodes ou objetos.

 O ícone do objeto é utilizado para representar um objeto na tree do FlexSim.

 O ícone do node é usado para especificar um node que possui algum dado dentro do objeto. Nodes de dados podem possuir outros nodes de dados em sua composição. Caso um node tenha o símbolo “+” a sua esquerda, significa que existem outros nodes de dados em seu interior. Os nodes de dados podem possuir valores numéricos ou alfanuméricos.

Alguns nodes de dados são utilizados para armazenar códigos executáveis. Existem quatro tipos de dados de códigos executáveis no FlexSim:

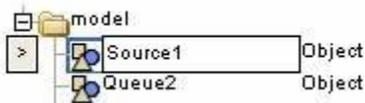
 C++. O código em C++ precisa ser compilado antes de rodar a simulação.

 Flexscript. Este código será auto-compilado durante a simulação. Leia mais sobre isso no campo “Escrevendo a Lógica no FlexSim”.

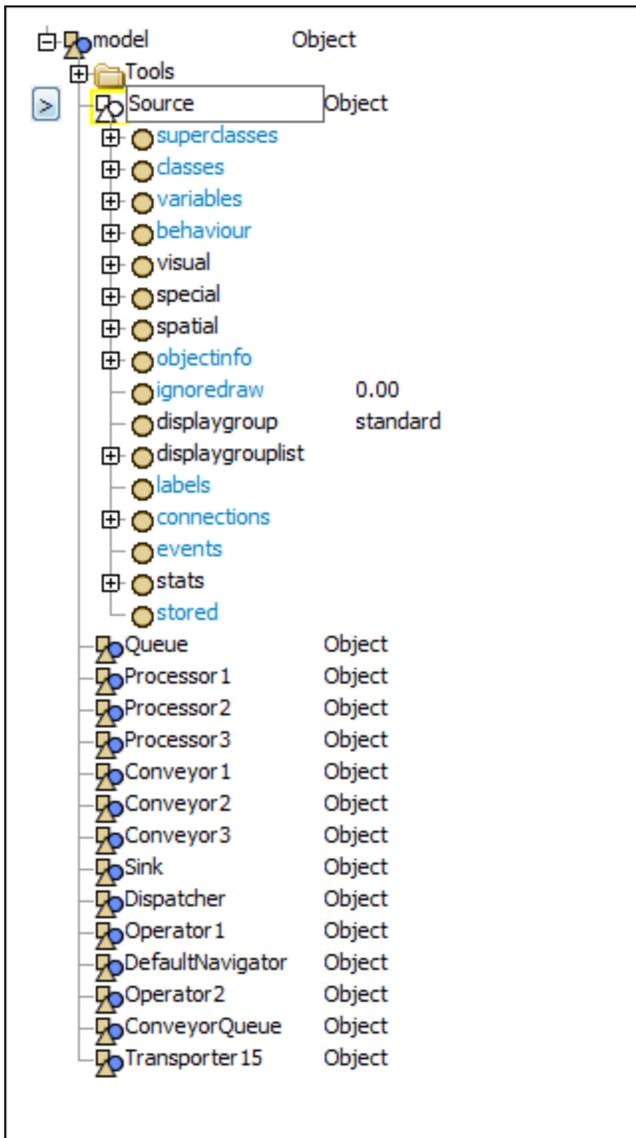
 Dll. Este node se refere a uma dll pré-compilada e compatível com o FlexSim. Para criar esta dll é necessário utilizar um projeto especial em Visual C++. Este projeto está disponível na comunidade do FlexSim.

 Global C++. Este é o código C++ em seu escopo global. Neste modo o código precisa conter funções completas. Estas funções podem ser acessadas de qualquer node presente no interior deste node. Normalmente este tipo de node se encontra como o primeiro node Guiaixo da pasta MAIN.

Quando algum objeto da tree é selecionado através de um clique no seu ícone na árvore, o objeto será mostrado da seguinte forma:



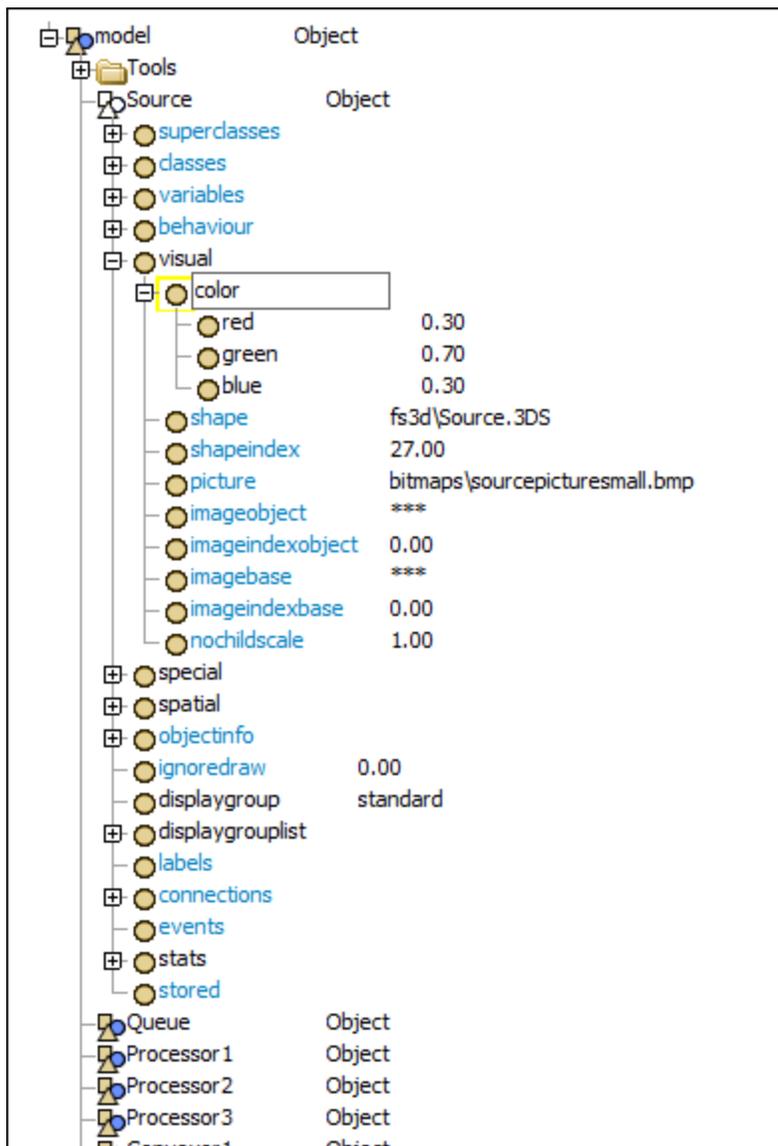
Uma caixa destacando o ícone do objeto aparecerá na tela juntamente com o símbolo de que aquele objeto pode ser expandido  na lateral esquerda do ícone do objeto. Caso você clique neste símbolo de expansão, os nodes de dados daquele objeto serão mostrados, conforme figura Guiaixo.



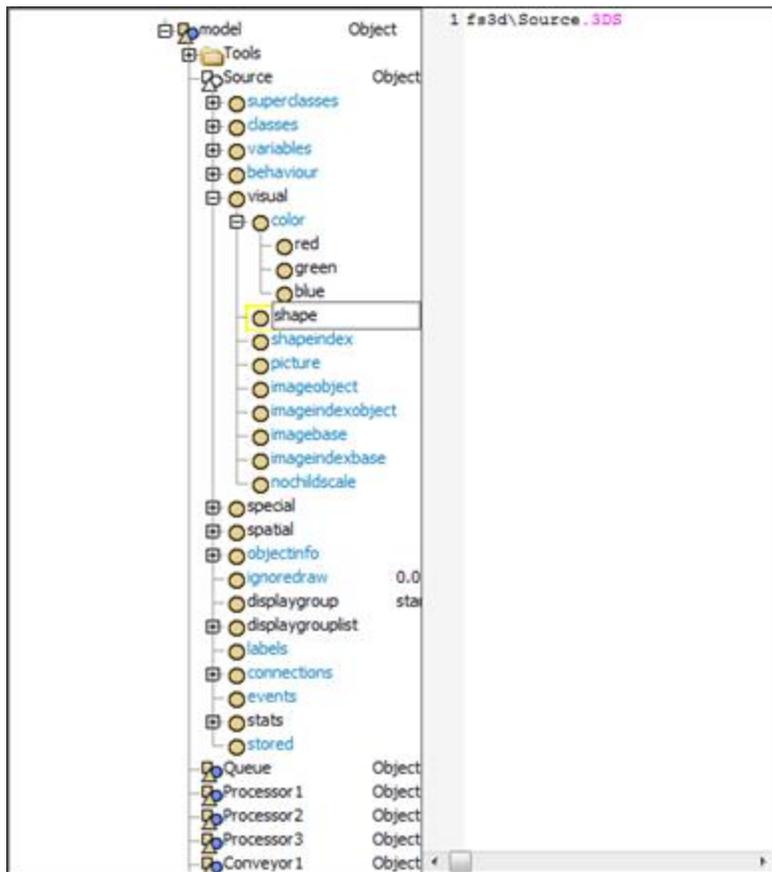
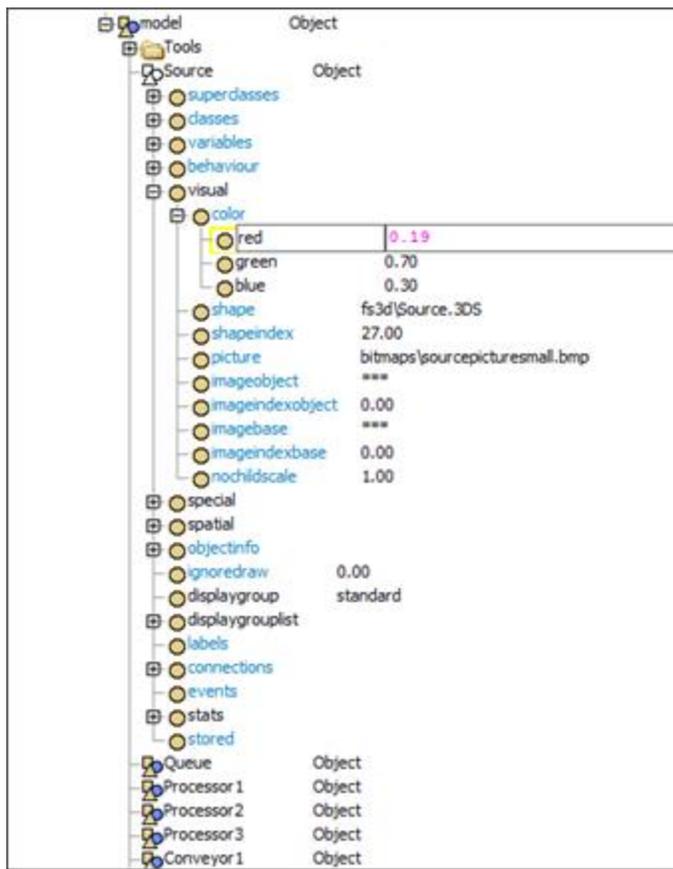
Como os objetos e os nodes de dados podem ser expandidos, a janela da tree pode crescer rapidamente, saindo dos limites do campo de visão da janela. O FlexSim lhe permite mover o tree para o lugar que achar melhor através de um clique na parte branca da janela e arrastando todo o tree para o local mais conveniente, facilitando a visualização do tópico de interesse.

Os nodes de dados podem ser expandidos através de um clique no “+” na lateral esquerda do ícone do node. Sabendo que os nodes podem possuir valores numéricos ou alfanuméricos, você verá na lateral direita do node o conteúdo que o mesmo está guardando.

Caso você selecione um objeto ou node provavelmente você não conseguirá mover a tree. Clique em uma área em branco e então mova o mouse para movimentar a tree para cima ou para baixo. Também é possível usar o scroll do mouse para movimentar para cima ou para baixo.



Os dados podem ser editados diretamente no tree através da seleção do node desejado. Caso este seja um node numérico, você será capaz de alterar o valor diretamente no campo em frente do node. Caso seja uma variável em forma de texto, o lado direito da janela se tornará uma janela específica para alterar aquele node.



Como você pode ver, a tree “Guiastece” todo o modelo com os dados necessários. A janela de propriedades é usada para facilitar o acesso do usuário aos nodes que estão na tree e fazem parte da composição do objeto. É totalmente possível manipular seu modelo através da tree, porém, é aconselhável que se faça todas alterações através da janela de propriedades para evitar a exclusão de dados necessários para o funcionamento correto do objeto. A janela de propriedades de qualquer objeto pode ser visualizada através de um clique duplo em cima do objeto na janela 3D do modelo, através de um clique duplo no ícone  do objeto na tree, ou então através da seleção Properties após um clique com o botão direito do mouse sobre um dos meios acima.

Tutoriais

1. Introdução
2. Exemplo 1
3. Exemplo 2
4. Exemplo 2 Extra Mile
5. Exemplo 3
6. Labels
7. Global Modeling Tools (Ferramentas Globais para Modelagem)
8. User Events (Eventos do usuário)
9. Time Tables (Tabelas de Tempos)
10. Kinematics
11. Task Sequence Tutorial (Tutorial Sequência de Tarefas)
 - Tutorial 1
 - Tutorial 2
 - Tutorial 3
12. SQL Tutorial (Tutorial do SQL)
13. Fluid Objects (Objetos de Fluido)
14. Experimenter / Optimizer Tutorial (Tutorial do Experimenter)

Introdução dos Tutoriais

Este tutorial básico irá guiá-lo através das etapas de criação de um fluxo de processo, na construção de um modelo, na introdução de dados, na visualização de uma animação, e na análise dos resultados. Cada exercício terá uma dependência com o exercício anterior, em termos de enunciado e sequência. Por isso, é importante entender completamente cada exercício antes de passar para o próximo. Você deve reservar pelo menos 45 minutos para completar cada exercício. O exercício 2 também irá incluir uma seção "Extra Mile" para agregar valor adicional ao seu modelo.

Cada exercício terá o seguinte formato:

1. Introdução
2. O que você irá aprender
3. Tempo aproximado de conclusão
4. Descrição do modelo
5. Dados do modelo
6. FlexSim software aprendido
7. Construção de modelos passo a passo

Os seguintes exemplos estão contidos neste tutorial:

Exercício 1:

Construa um modelo simples que irá processar 3 tipos diferentes de flowitem. Cada itemtype terá um encaminhamento específico. Objetos usados neste modelo serão o Source, Queue, Processor, Conveyor, e o Sink. As estatísticas básicas de desempenho do modelo serão apresentados, e as opções de parâmetros para cada objeto será informado.

Exercício 2:

Usando o modelo do exercício 1, iremos adicionar Operators e Transporters no processo. As propriedades dos objetos serão introduzidas, e análise estatística adicional com conceitos, será discutido.

Exercício 2 Extra Mile:

Após completar o Exercício 2, nós iremos mostrar como adicionar gráficos e 3D charts no modelo, usando o Dashboard. 3D visual text será adicionado usando o objeto VisualTool para dar informação em tempo instantâneo ao modelo.

Exercício 3:

Usando o modelo do exercício 2, você irá adicionar racks e network paths. Iremos adicionar alguma lógica e análise estatística ao modelo, assim como global tables usadas para ler e escrever dados.

Labels

Esta lição irá introduzir a utilização de labels. Você também conhecerá como o FlexSim trGuialha com Pull Requirements.

Propriedades Globais:

Esta lição irá apresentar os conceitos básicos de global properties que podem ser usados durante todo os modelos. Você também será apresentado aos objetos Combiner e Separator.

User Events (Eventos do Usuário)

Este exercício vai mostrar como utilizar o User Events. Qual pode ser utilizado para executar códigos específicos em tempos específicos durante a execução do modelo.

Time Tables (Tabelas de Tempo)

Este exercício irá introduzir ao uso de Time Tables no FlexSim.

Kinematics

Este exercício irá introduzi-lo ao uso do kinematics para realizar movimentos simultâneos com um objeto simples.

Sequência de Tarefas (Task Sequences)

Este exercício irá mostrar como usar Task Sequences e como customizá-las.

SQL

Este exercício irá mostrar como executar a leitura de um banco de dados SQL para/do FlexSim.

Não hesite em contatar o nosso grupo de suporte técnico se você tiver dúvidas, enquanto trabalhar com estes tutoriais. O suporte técnico com Flexsim pode ser alcançado através do telefone (19) 3308-7833, ou você pode enviar um e-mail suas perguntas através de um formulário web em <http://www.flexsim.com/support/>. Esperamos que goste de usar o Flexsim para ajudá-lo a otimizar seus processos.

Nota: As estatísticas dos modelos que serão construídos pode não ser exatamente o mesmo que aqueles encontrados no tutorial. Pequenas diferenças em itens simples, como a colocação de objetos pode alterar os resultados de longo prazo. Para efeitos de aprendizagem, essas mudanças não são importantes.

Tutorial Exercício 1

1. Introdução
2. Construção do modelo passo-a-passo

Introdução

O Exercício 1 irá introduzir os conceitos básicos de diagramação e construção de um modelo simples. Construir um diagrama do processo é uma ótima maneira de começar a construir modelos no Flexsim. Se você não pode construir um diagrama, fluxograma, ou pelo menos ver uma imagem em sua mente de como funciona o processo, você terá uma certa dificuldade em modelar em qualquer software de simulação.

Nota: Se você já visualizou o Getting Started Tutorial do software, muitos dos conceitos que você aprenderá neste material não será nova. No entanto, lições subsequentes irão ampliar seu conhecimento, por isso é uma boa idéia continuar adiante na leitura deste guia.

O que você irá aprender

- Como construir um simples layout
- Como conectar portas para dar a rota aos flowitems
- Como detalhar e entrar com os dados nos objetos do FlexSim
- Como navegar em animation views
- Como visualizar informações estatísticas simples nos objetos do FlexSim

Novos Objetos

Neste exercício, iremos apresentá-lo os objetos Source, Queue, Processor, Conveyor, e Sink.

Tempo Aproximado para completar este exercício

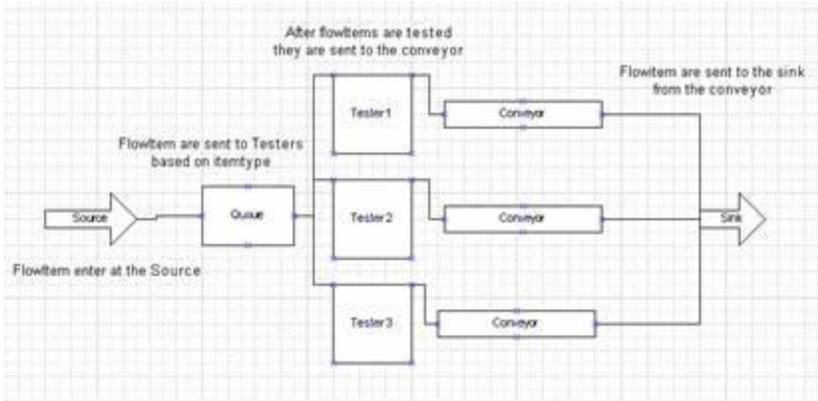
Este exercício deve levar de 30-45 minutos para ficar pronto.

Model Views

FlexSim usa um ambiente de modelagem tri-dimensional. O modelo de visualização padrão para construção de modelos é chamado de orthographic view. Você pode visualizar o modelo em uma perspectiva de visualização mais realística. É geralmente mais fácil de construir o layout do modelo em ortographic view, considerando que a perspectiva é mais para fins de apresentação. No entanto, você pode usar qualquer opção de visualização para construir ou executar o modelo. Você pode ter aberto tantas janelas quanto você desejar no Flexsim. Basta lembrar que, quantos mais janelas são abertas, maior a demanda sobre os recursos do computador.

Descrição do Exercício 1

Em nosso primeiro modelo, vamos olhar para o processo para testar três produtos saindo de uma linha de produção. Existem três flowitem itemtypes (três diferentes tipos de entidades ou flowitem) diferentes que chegarão com base numa distribuição normal. Serão distribuídos uniformemente entre itemtypes 1, 2 e 3. Os flowitems chegam e, como eles serão colocados em uma fila, esperam para ser testado. Três equipamentos (processors) estarão disponíveis para o teste. Um testador será utilizado para itemtype 1, outro para o itemtype 2, e o terceiro para itemtype 3. Uma vez que o flowitem é testado ele será colocado em um conveyor. No final do conveyor, o flowitem serão enviados para um sink de onde sairá do modelo. A Figura 1-1 mostra um diagrama do processo.



Dados do Exercício 1

Taxa de chegada no Source: normal(20,2) segundos

Capacidade máxima do Queue: 25 flowitems

Tempo de teste: exponencial(0,30) segundos

Velocidade do Conveyor: 1 metro/segundo

Rotas dos Flowitems: Itemtype 1 para Tester 1, Itemtype 2 para Tester 2, Itemtype 3 para Tester 3.

Construção do Modelo Passo a Passo

Construindo Modelo do Exercício 1

Abra o aplicativo através de um clique duplo no ícone do FlexSim na Área de trabalho. Uma vez que o software for carregado, você deverá visualizar o menu Flexsim e suas ferramentas, Biblioteca e a janela de Visualização Ortográfica do Modelo.

Se em qualquer momento da modelagem você se sentir em dificuldades, existe um tutorial completo do software para auxiliá-lo no endereço <http://www.flexsim.com/tutorials>.

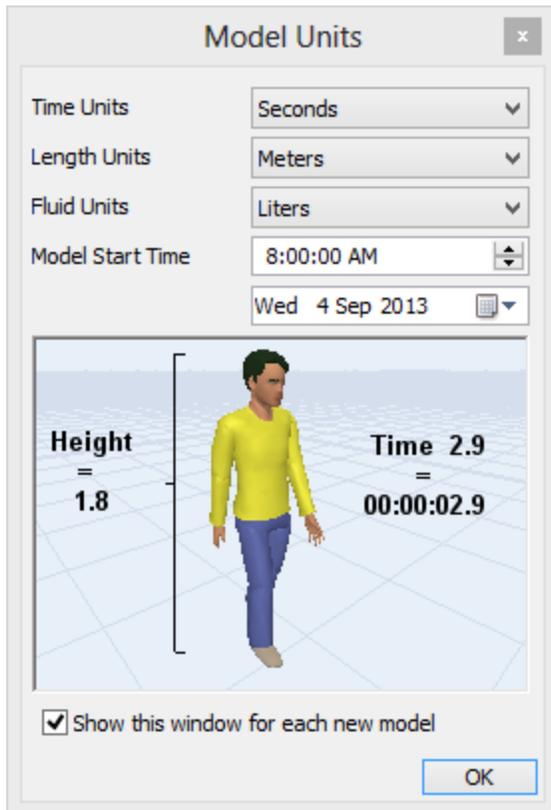
Iniciando um novo modelo

- Abra o FlexSim dando um duplo clique sobre o ícone do Flexsim em seu desktop. O guia de início aparece por padrão. Selecione "Build a New Model".



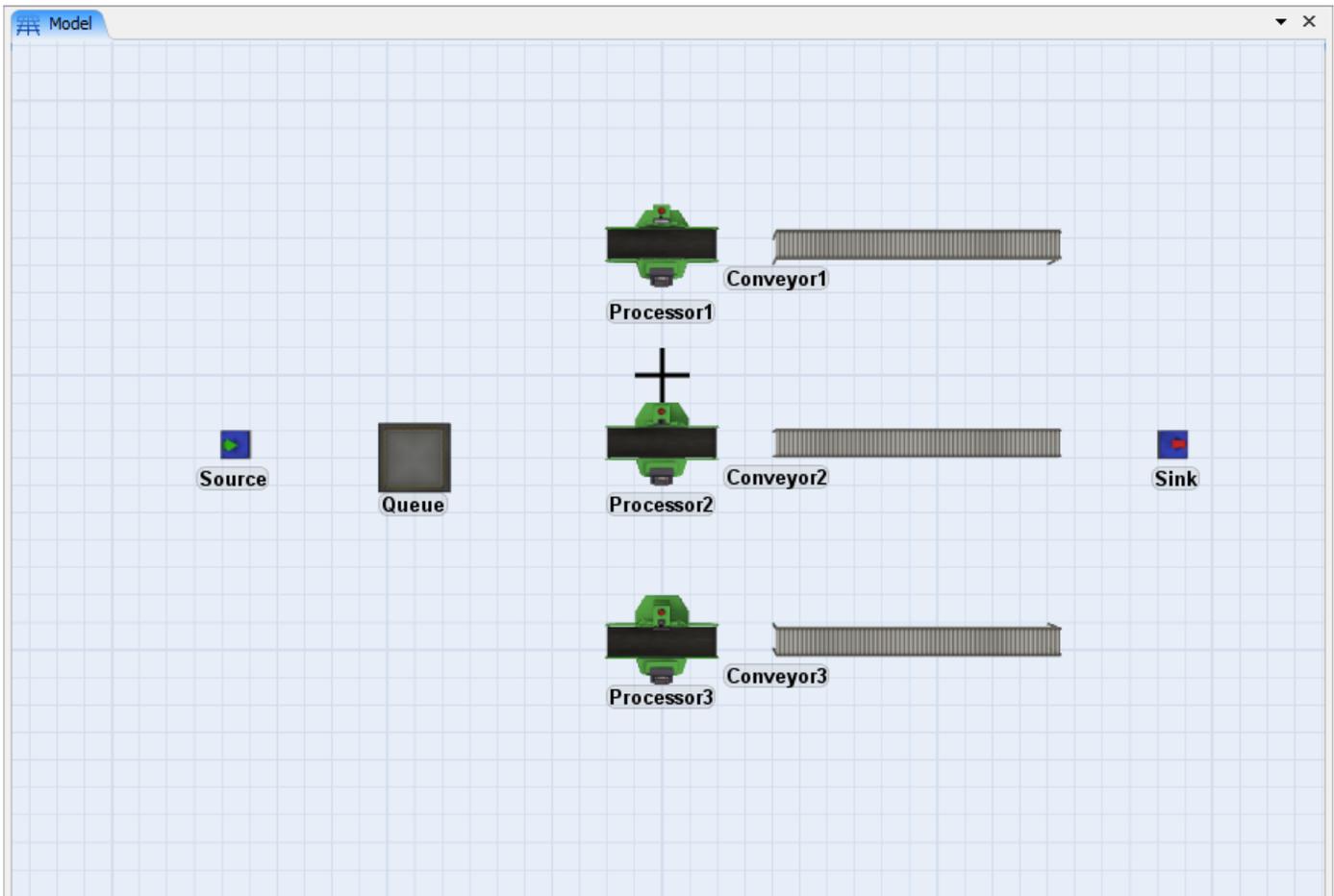
- Abra o FlexSim através de um clique duplo no ícone que está na área de trabalho. A janela "Startup Wizard" aparecerá na tela como padrão. Selecione então a opção "Build a New Model" para começar um modelo novo.

- **Time Units:** Segundos
- **Length Units:** Metros
- **Fluid Units:** Litros



Etapa 1: Crie os Objetos

- Crie um **Source** no modelo e o nomeie como *Source* (Para ver como fazer isso, clique aqui).
- Crie um **Queue**, 3 **Processors**, 3 **Conveyors**, e 1 **Sink** no modelo. Posicione-os e os nomeie de acordo com a imagem Guiaixo. Para nomear um objeto: clique duplo em cima do mesmo, altere o nome localizado no topo da janela de Propriedades (**Properties**) , e pressione **Apply** ou **OK**. Clique aqui para ver como isso é feito.

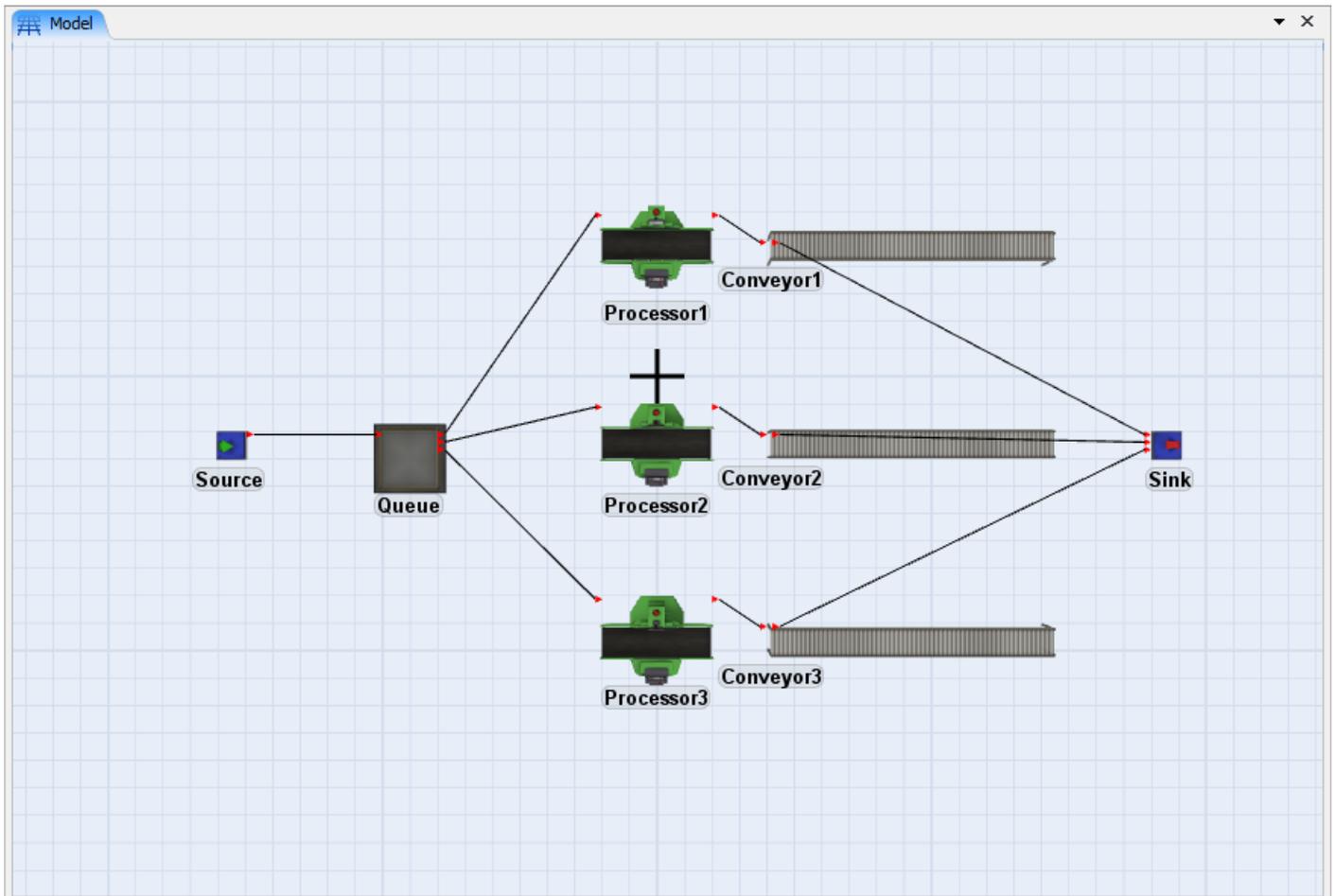


Etapa 2: Conecte as portas

Escolha o modo de conexão clicando no ícone  na barra de ferramentas ou mantenha pressionado a tecla A do teclado. Uma vez selecionado o modo de conexão, existem duas maneiras de fazer as conexões entre os objetos. Você pode clicar em um objeto e depois clicar no outro, ou clicar no primeiro, manter o botão pressionado, arrastar até o próximo objeto e soltar o botão do mouse. Outra coisa, nunca se esqueça que as conexões do modelo devem ser feitas de acordo com o fluxo do modelo, ou seja, de onde o item está (primeiro objeto conectado) para onde o item vai (segundo objeto conectado).

Para desfazer as conexões, basta escolher a opção  no menu de ferramentas ou manter pressionada a tecla Q e desfazer as conexões na mesma ordem que elas foram criadas.

- Conecte o *Source* ao *Queue*.
- Conecte o *Queue* ao *Processor1*, *Processor2*, e *Processor3*.
- Conecte o *Processor1* ao *Conveyor1*, *Processor2* ao *Conveyor2*, e *Processor3* ao *Conveyor3*.
- Conecte o *Conveyor1*, *Conveyor2*, e *Conveyor3* ao *Sink*.

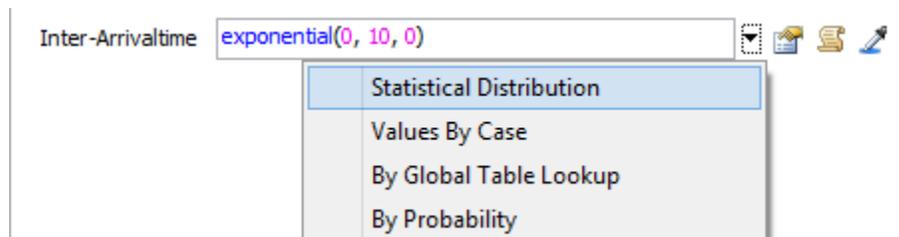


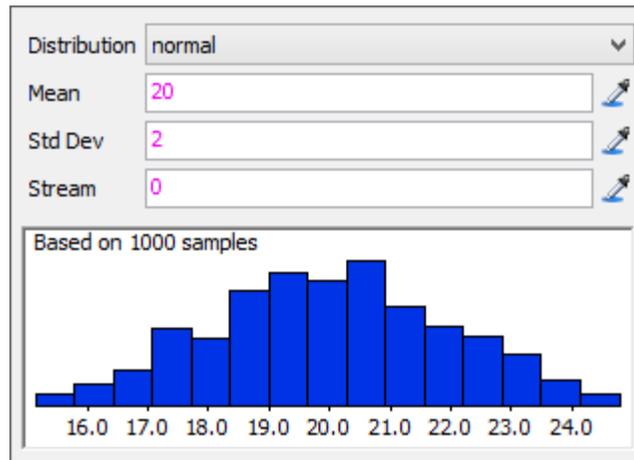
Etapa 3: Atribuição da taxa de chegada

Neste modelo será necessário alterar o campo Inter-Arrival time e o itemtype para serem gerados 3 itens diferentes.

- Clique duplo no *Source* to para abrir sua janela de **Properties**
- Na própria Guia *Source*, selecione a opção **Statistical Distribution** na picklist do campo **Inter-Arrivaltime**. Uma janela com as distribuições estatísticas aparecerão.
- Escolha a **Distribuição** normal.
- Defina o campo **Mean** para 20.
- Defina o campo **Std Dev** para 2.
- Defina o campo **Stream** para 0.

Estas configurações definem o tempo entre as chegadas para uma média de 20 segundos distribuídos de acordo com uma distribuição normal com Desvio Padrão de 2 segundos.





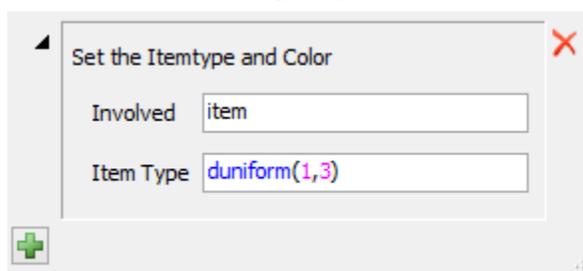
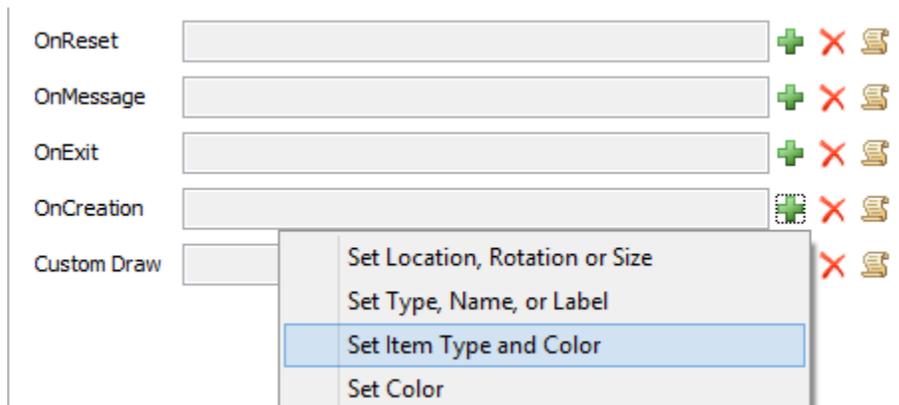
Feito isso, deve-se atribuir o número do itemtype para cada item de acordo com o modo que eles entram no sistema. Este número deve ser distribuído uniformemente entre 1 e 3. A melhor maneira de fazer isso é através da trigger OnCreation da Guia Triggers do Source, mas não feche a janela ainda.

Etapa 4: Configuração do tipo de item e cor

- Clique na Guia **Triggers**, e adicione uma função através do botão  na opção **OnCreation** e selecione a opção **Set Itemtype and Color**. Uma template de código irá se abrir.

A função duniform é similar a distribuição uniforme, porém, a única diferença é que a duniform gera apenas número inteiros entre o intervalo definido, e não números reais como a uniforme. Neste exemplo usaremos o texto padrão, pois o mesmo indica que serão gerados entre 1 e 3.

- Clique em **OK** para aplicar as alterações e fechar a janela de Propriedades.



A próxima etapa é a de configurar o Queue. Considerando que o Queue é um objeto que tem como principal função armazenar itens até que os mesmos sejam processados em algum centro de processamento,

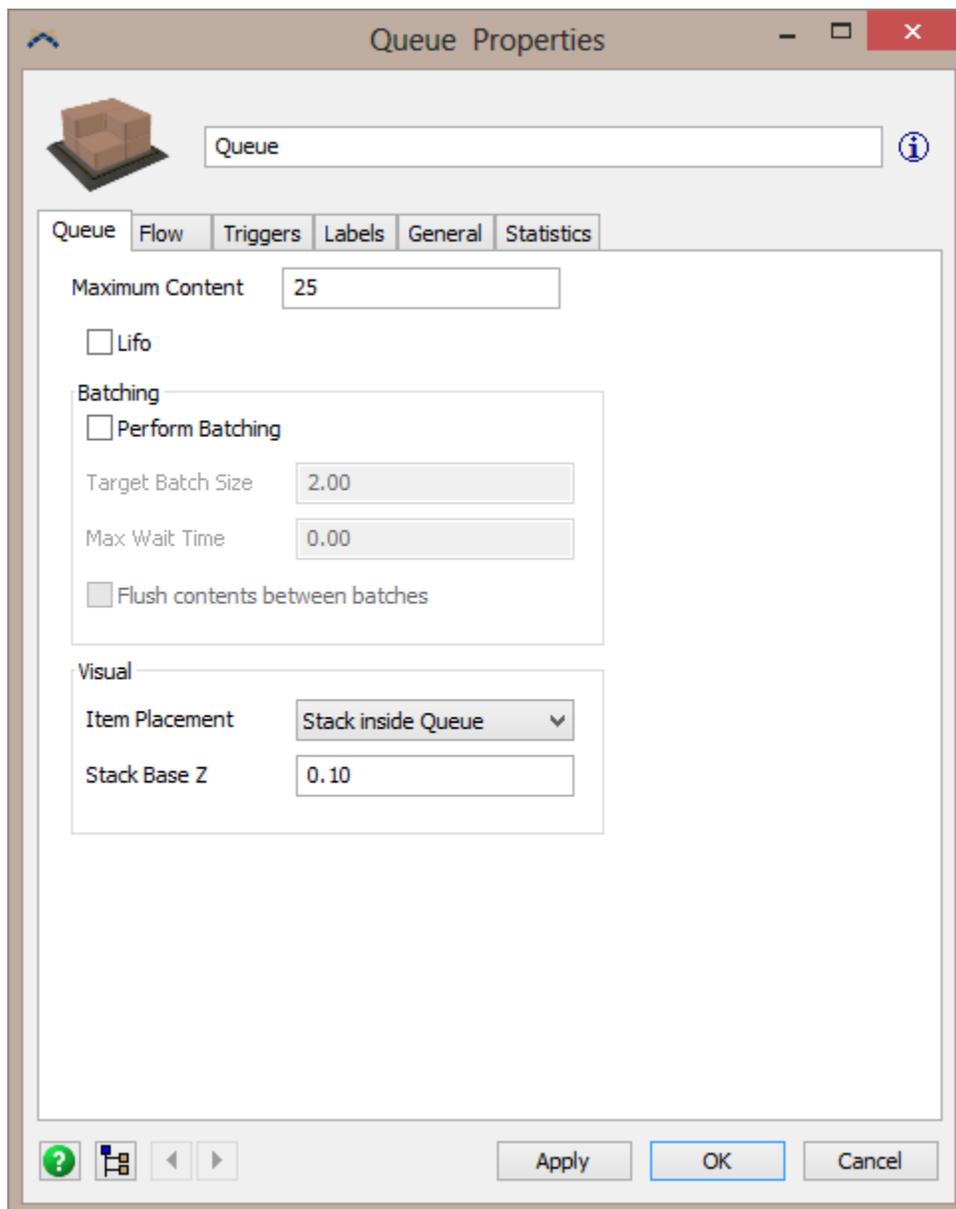
existem basicamente duas coisas que precisaremos fazer. Primeira, ajustar a capacidade do Queue para que seja armazenado no máximo 25 produtos. Segunda, definir a configuração para que o Queue envie itens itemtype igual a 1 para o *Processor*, itemtypes igual a 2 para o *Processor 2*, e itemtypes igual a 3 para o *Processor 3*.

Etapa 5: Configure a Capacidade Máxima do Queue

Você pode configurar a capacidade máxima do Queue ao clicar sobre o Queue no formato de visualização em 3D para visualizar suas propriedades na janela Quick Properties.

Alternativamente, você pode abrir a janela de propriedade dos objetos.

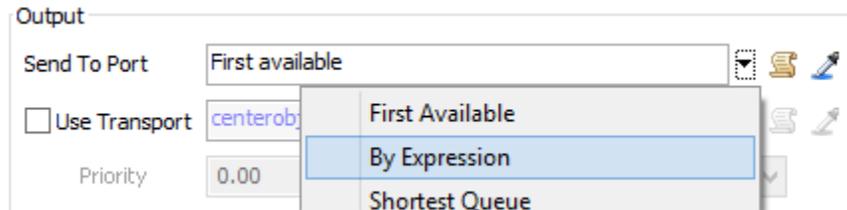
- Clique duplo no Queue para abrir a sua janela de **Propriedades**.
- Alterar a variável **Maximum Content** para 25. Isso é a capacidade máxima da fila.
- Não feche a janela ainda.



Etapa 6: Defina a lógica de fluxo para o Queue

Você pode definir a lógica de fluxo do Queue ao clicar sobre o Queue na visualização em 3D para mostrar suas propriedades na janela Quick Properties. Alternativamente, você pode abrir a janela de propriedades dos objetos:

- Clique na Guia **Flow** na propriedade do Queue para definir as opções do fluxo do Queue
- No campo **Send To Port**, selecione a opção **By Expression**.



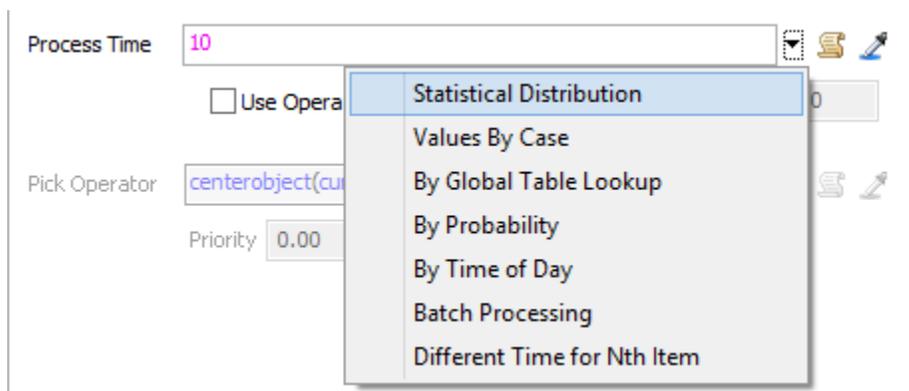
A partir do momento que definimos os tipos de itens no Source iguais a 1, 2 e 3, nós podemos agora utilizar estes tipos de itens para tomar uma decisão ao longo do modelo, neste caso, escolher por qual porta de saída o item seguirá dependendo do seu *itemtype*. Note que a função padrão existente na opção **By Expression** é a função `getitemtype(item)`. Deixe esta opção. O *Processor 1* deve estar conectado a porta de saída número 1, o *Processor 2* a porta número 2 e o *Processor 3* a porta número 3. Clique fora da caixa para adicionar as alterações.

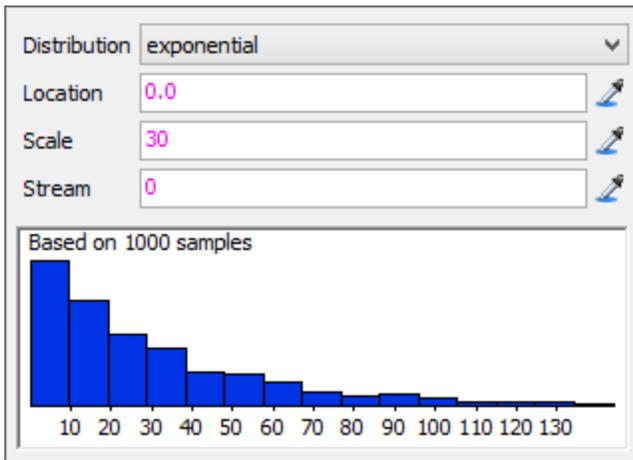
Clique no botão **OK** para adicionar as configurações e fechar a janela de propriedades do Queue.

A próxima etapa é a de configurar os tempos dos processos.

Etapa 7: Defina o tempo de processamento

- Clique duplo no *Processor1* para abrir a sua janela de **Propriedades**.
- Na Guia **Procesor**, na seção **Process Time**, selecione a opção **Statistical Distribution** na lista do campo **Time**. A janela de distribuições estatísticas aparecerá.
- Defina a **Distribuição** para “exponential”.
- Defina o parâmetro **Location** para 0
- Defina o parâmetro **Scale** para 30.
- Defina o parâmetro **Stream** para 0.
- Repita o mesmo procedimento para os outros 2 processors.





A velocidade padrão da esteira já está definida para 1 m/s, portanto não será feito qualquer alteração nas variáveis do conveyor por enquanto.

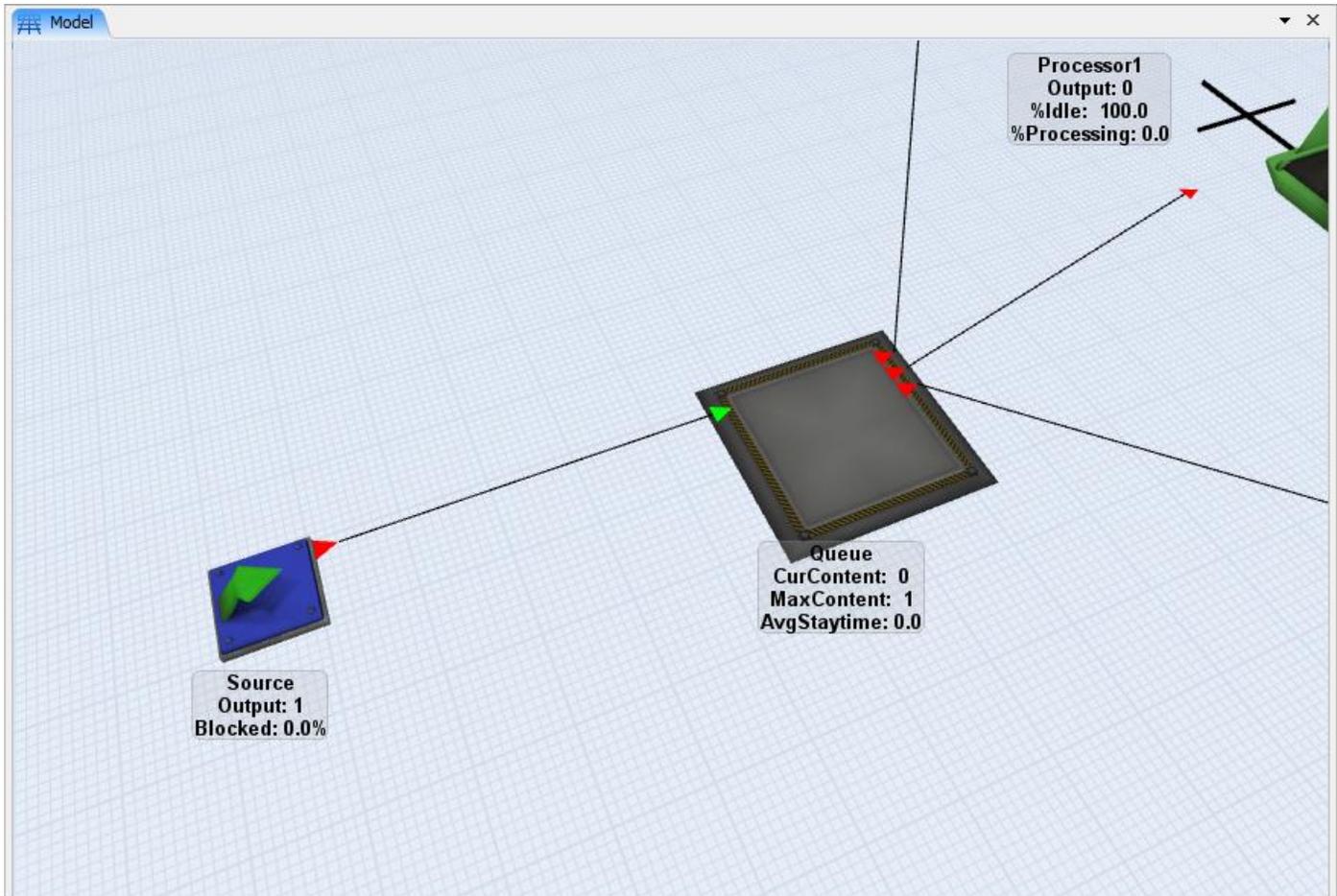
Estamos então prontos para rodar a simulação do modelo criado.

Etapa 8: Resete e Execute o modelo

- Sempre clique no botão  **Reset** para resetar a simulação e atualizar todos os parâmetros do sistema para sua posição inicial antes de rodar a simulação de qualquer modelo.
- Clique no botão  **Run** para iniciar a simulação.

Deverá aparecer na tela os itens sendo criados pelo *Source*, passando pelo *Queue*, sendo processados nos *Processor*, indo através dos *Conveyors* ao *Sink*. É possível alterar a velocidade da simulação através da barra *Run Speed* localizado no *Simulation Run Panel*.

Etapa 9: Visualize estatísticas simples



A imagem acima demonstra como visualizar estatísticas simples em cada objeto. Caso nada esteja sendo mostrado, ou somente os nomes estejam aparecendo, você pode alterar a View Setting para visualizar as estatísticas. Para alterar as configurações de visualização, basta clicar com o botão direito do mouse em uma área branca do modelo e escolher a opção **“View Settings > Show Names and Stats”**.

Você pode agora mais estatísticas de um objeto ao clicar sobre o objeto e visualizar o painel Quick Properties ao lado direito da tela.

Quick Properties

▼ Statistics

State: processing

Throughput

Input	Output
3.00	2.00

Content

Curr	Min	Max	Avg
1.00	0.00	1.00	0.81

Staytime

Min	Max	Avg
10.00	10.00	10.00

Open as Window

Etapa 10: Salve o modelo

Salve seu modelo clicando no botão Save na barra de ferramentas. A janela “Save FlexSim Model file” aparecerá, permitindo-lhe escolher o lugar onde o arquivo será salvo. No campo “File name” escolha um nome apropriado para salvar seu modelo (exercicio1.fsm) e clique em Save. Lembre-se que a extensão do nome do arquivo precisa estar em .fsm.

Você acabou de completar o Exercício 1 deste Tutorial. Gaste um pouco mais do seu tempo revisando como o modelo foi feito e como ele roda. Parabéns!!

Para continuar com o Tutorial, siga para o Exercício 2.

Exercício 2 Tutorial

1. Introdução
2. Construção passo-a-passo do modelo

Introdução

O Exercício 2 tem como objetivo principal introduzi-lo ao conceito da utilização de operadores e transportes para executar tarefas no modelo, e também iniciar a exploração das propriedades dos objetos em maiores detalhes. O Exercício 2 também introduz algumas estatísticas gráficas como meios de análise do modelo. Tenha certeza que completou o Exercício 1 antes de iniciar o Exercício 2, pois este último utilizará o primeiro exercício como ponto inicial.

O que você irá aprender

- Como acessar e modificar as propriedades dos objetos
- Como adicionar uma equipe de operadores ao modelo
- Como adicionar uma empilhadeira para transportar os produtos
- Como selecionar um objeto para visualizar suas estatísticas através de gráficos
- Como habilitar a coleta de dados estatísticos
- Como visualizar as estatísticas dos objetos enquanto o modelo está executando

Novos objetos

Neste modelo será utilizado, além, dos já conhecidos, os objetos Dispatcher, Operator e Transporter.

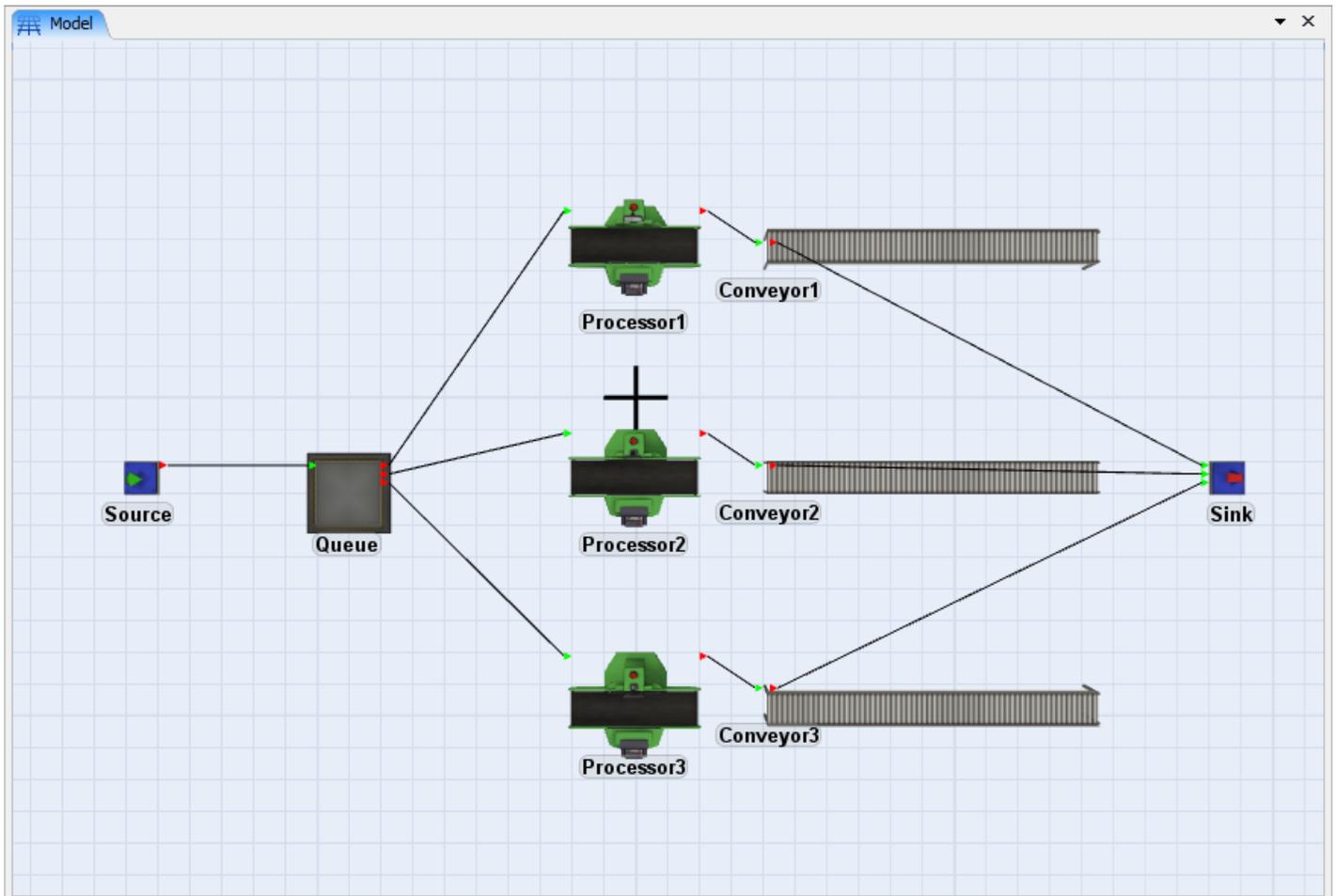
Tempo aproximado para completar este exercício

Este exercício levará de 30-45 minutos para ser completado.

Overview do Exercício 2

No modelo 2 será usado uma equipe de operadores para realizar o setup para o início dos testes que serão executados nos produtos. Um dos dois operadores precisarão iniciar o teste. Uma vez iniciado, o teste poderá ser feito sem a presença dos operadores. Os operadores precisam ainda carregar os itens do Queue para o equipamento de teste antes de iniciarem os testes. Quando o teste estiver completo, os itens seguem para o conveyor sem necessidade de assistência dos operadores.

Quando os itens chegam ao final do conveyor, os mesmos caem no Queue e aguardam pela empilhadeira que os leva ao Sink. Talvez seja necessário ter mais de uma empilhadeira para realizar o transporte, mas isso será visto logo que o modelo começar a rodar. Depois do modelo completado será possível visualizar os gráficos padrões e por meio deles identificar os possíveis gargalos do modelo juntamente com as preocupações relacionadas a eficiência.



Dados do Modelo 2

Tempo de preparação para o teste: 10 segundos

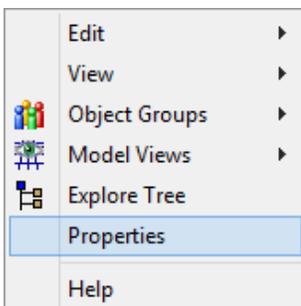
Movimentação do Produto: Do Queue para o Processor com o Operador. Do Queue para o Sink com a empilhadeira.

Capacidade do Queue: 10 itens.

Novos Conceitos

Propriedades dos Objetos

Iremos agora explorar um pouco mais o Object Properties Window de uma maneira mais organizada. Todos os objetos do FlexSim possuem uma Properties Window, ou seja, uma janela de propriedades. Para acessá-la, basta dar um clique duplo em cima do objeto, ou clicar com o botão direito e escolher a opção **Properties**.



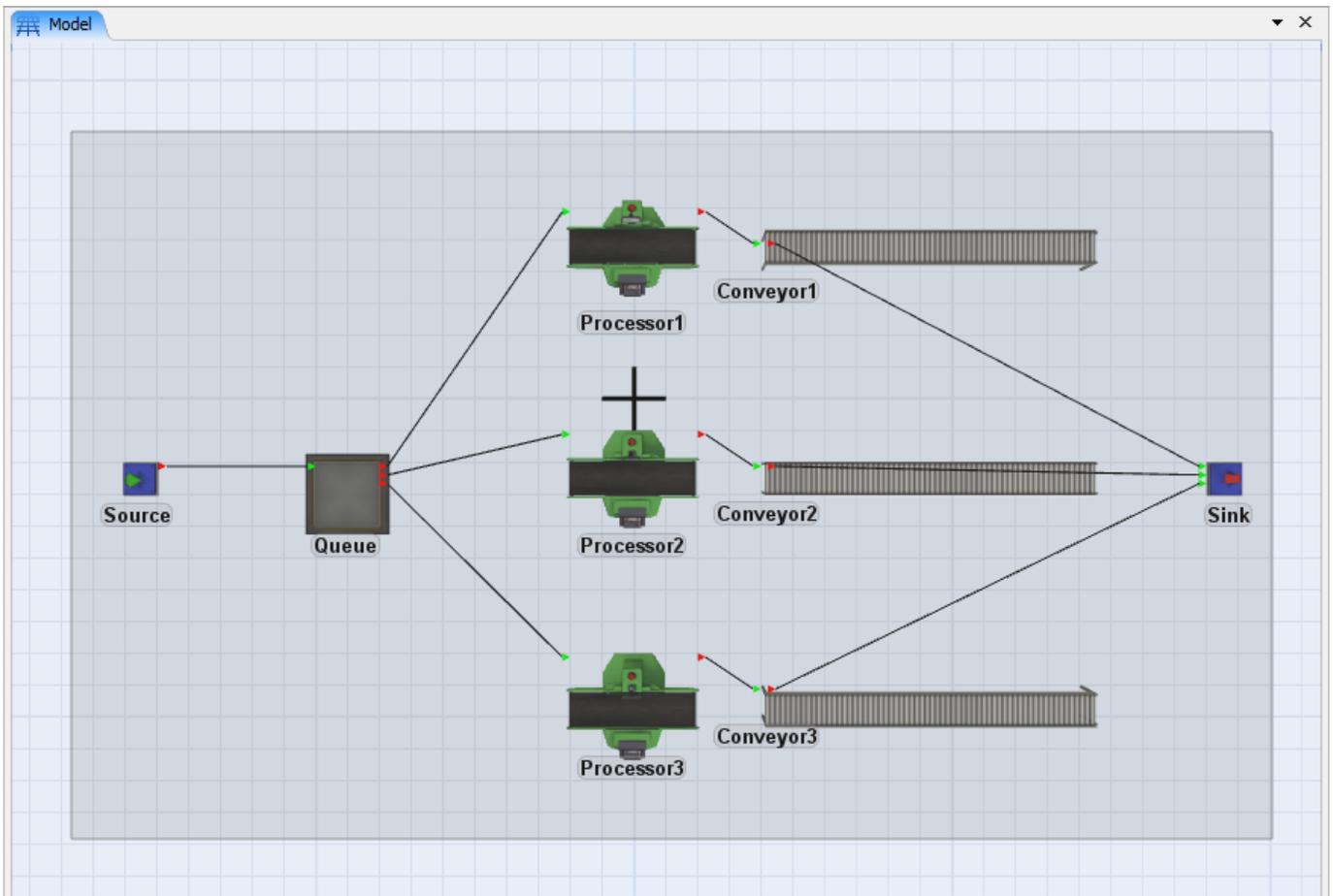
A janela de propriedades lhe permite configurar atributos que dependem do objeto que está sendo editado. Por exemplo, um Queue possui o conteúdo máximo, enquanto um Combiner possui uma lista de componentes para receber o itens. Portanto, a janela de propriedades do Queue é bem diferente da janela de propriedades do Combiner. Porém, existem algumas propriedades que são semelhantes nas Properties Windows. Por exemplo, tanto o Queue quanto o Combiner possui a capacidade de enviar os itens para a próxima operação, necessitando assim de uma estratégia para isso. Objetos semelhantes possuirão uma parte de suas propriedades iguais, o que agilizará seu aprendizado em relação as propriedades dos objetos.

Etapa 1: Selecione objetos para estatísticas

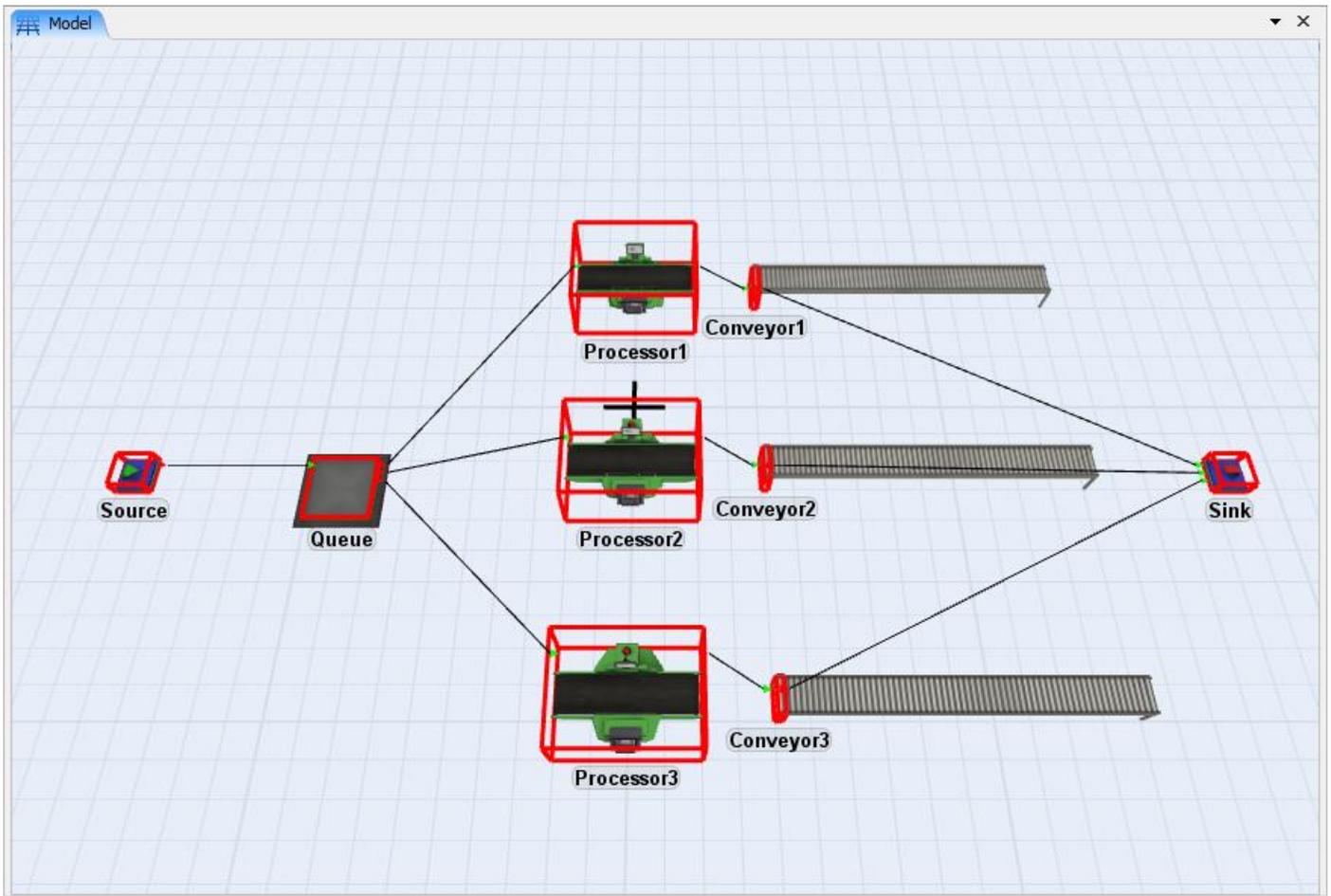
Na janela de visualização você precisará fazer a seleção dos objetos de acordo com os objetos que você gostaria de visualizar as estatísticas. Isso é feito mantendo pressionada a tecla “Shift” do teclado enquanto se faz um “retângulo” em volta dos objetos que se deseja selecionar. Esta seleção pode ser feita também clicando em  na barra de ferramentas.

 o qual colocará você no **New Selection Mode**.

Pressionando a tecla “Ctrl” e ir clicando individualmente nos objetos para selecionar/anular a seleção também é outra opção válida.

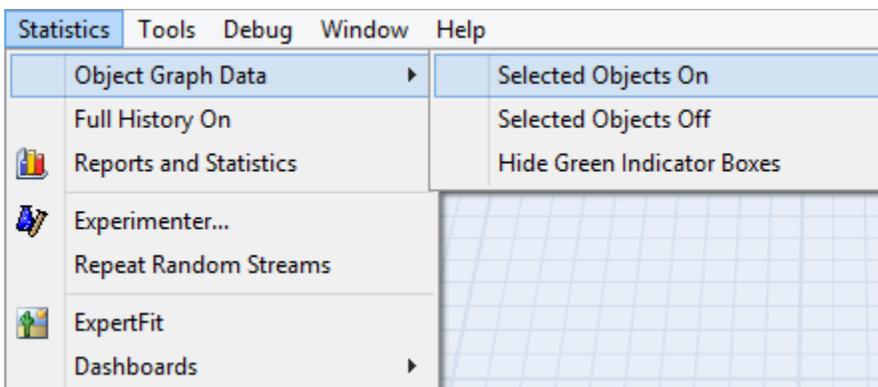


Uma vez que os objetos são selecionados, você verá uma caixa em vermelho ao redor de cada um.

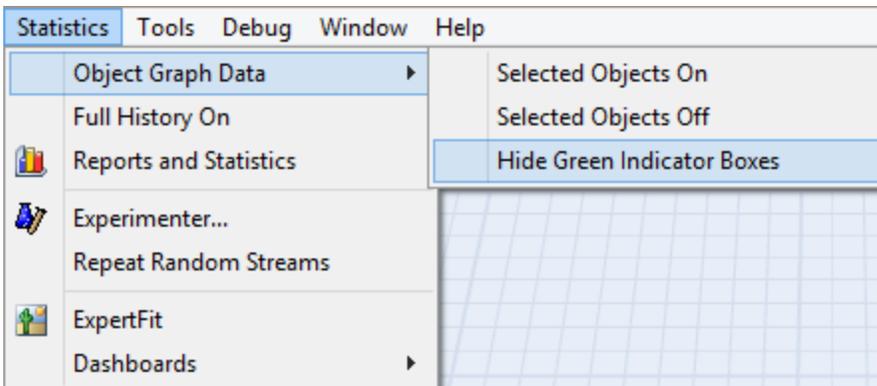
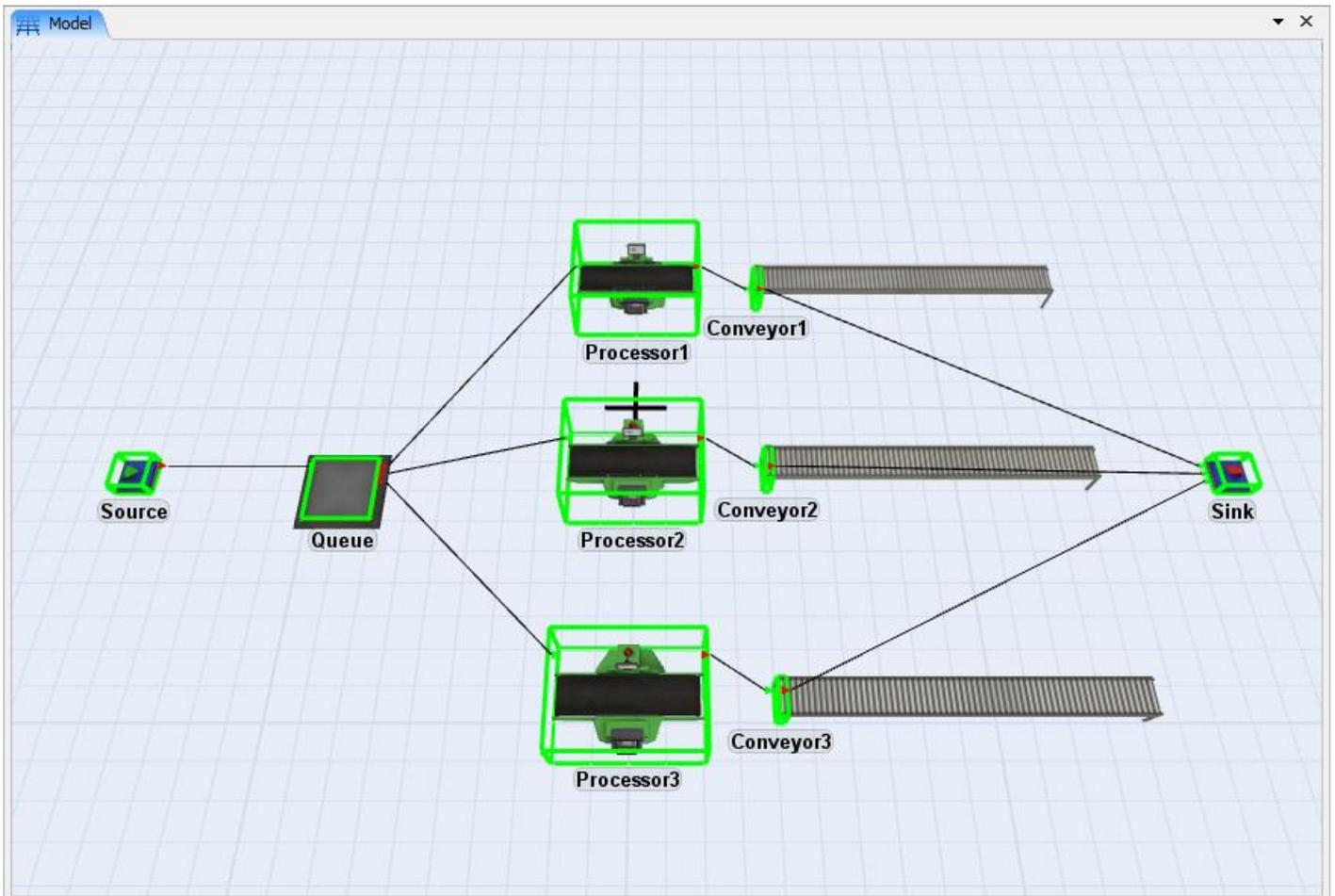


Etapa 2: Ativar as estatísticas

Para capturar o histórico estatístico de um objeto selecionado, clique em **Statistics > Object Graph Data > Selected Objects On**. **Na versão 7.3 isso não se faz mais necessário. As estatísticas são capturadas automaticamente.**



Uma vez que os objetos selecionados forem ativados para gravar as estatísticas, os mesmos ficarão com uma caixa verde em volta. Para não visualizar esta caixa verde, basta clicar na opção **Statistics > Object Graph Data > Hide Green Indicator Boxes**. **Na versão 7.3 isso não se faz mais necessário.**



Agora é possível coletar o histórico de dados dos objetos selecionados enquanto a simulação roda. Este foi mais um conceito aprendido do software FlexSim. Agora é a hora de iniciar o Exercício 2.

Construção do Modelo Passo a Passo

Construindo Exercício 2

Antes de construir o Modelo 2, é necessário carregar o modelo do Exercício 1.

Caso sinta alguma dificuldade a qualquer momento do exercício, existe um completo tutorial disponível para consulta no endereço <http://www.flexsim.com/tutorials>

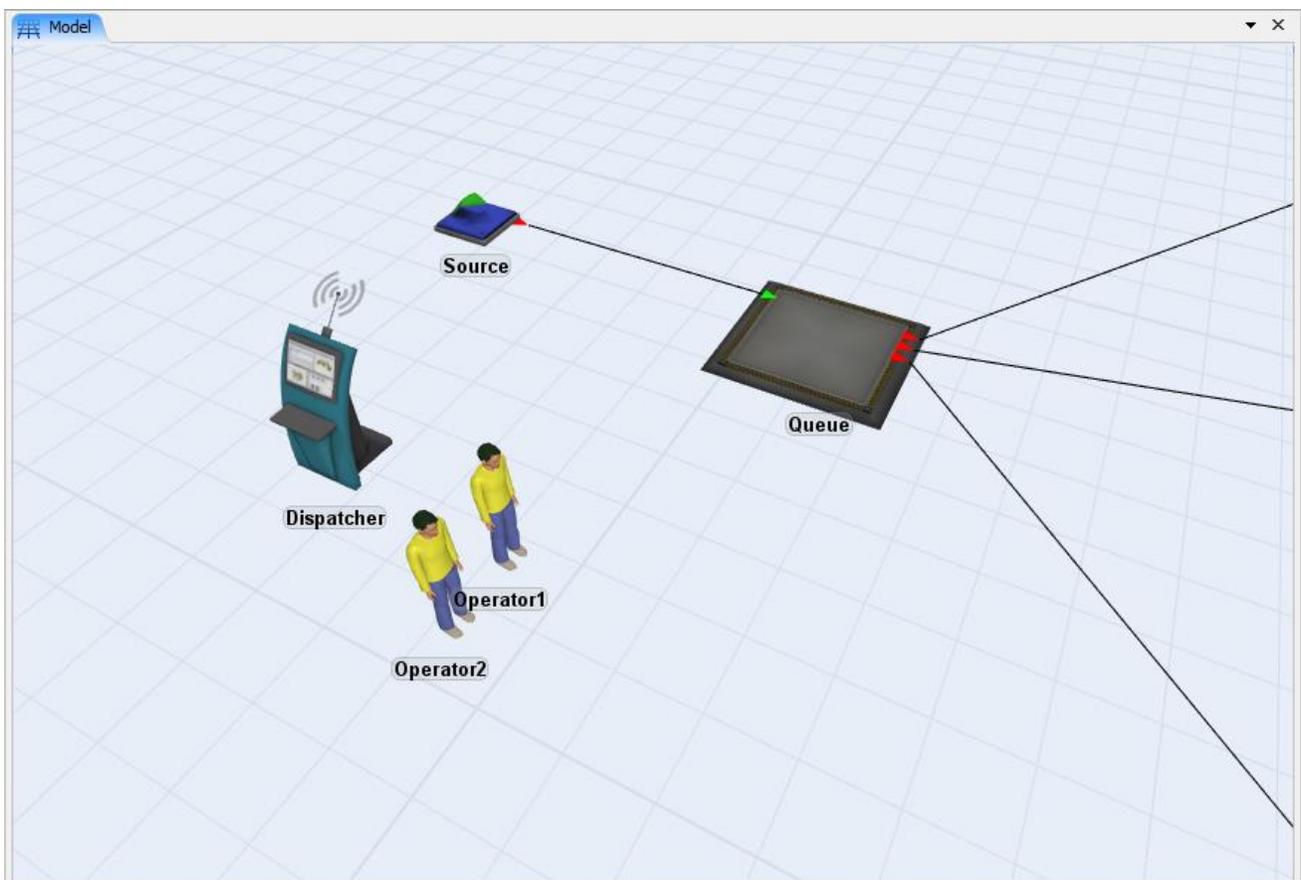
Etapa 1: Abra o modelo do exercício 1

- Caso o seu modelo do Exercício 1 não esteja aberto, carregue-o através do botão  na barra de ferramentas. Selecione o arquivo do Modelo 1 (arquivo do tipo .fsm) salvo no Exercício 1.
- Nós queremos fazer com que nossos flowitems saiam do Source a uma taxa mais rápida para este exercício. Dê um duplo clique sobre o Source para abrir sua janela de propriedades (**Properties**), e em **Inter-Arrivaltime** altere a média (**Mean**) para 12.

Etapa 2: Crie um dispatcher e 2 operadores

O Dispatcher é utilizado para organizar as seqüências de tarefas recebidas e colocá-las em ordem de modo a distribuir de acordo com a regra definida, normalmente se envia as tarefas para o primeiro operador disponível. Neste caso o Dispatcher será usado com 2 operadores que terão como tarefa transportar os itens do Queue aos Testadores.

- Crie um **Dispatcher** e o nomeie como *Dispatcher*.
- Crie 2 **Operators** e os nomeie como *Operator1* e *Operator2*.



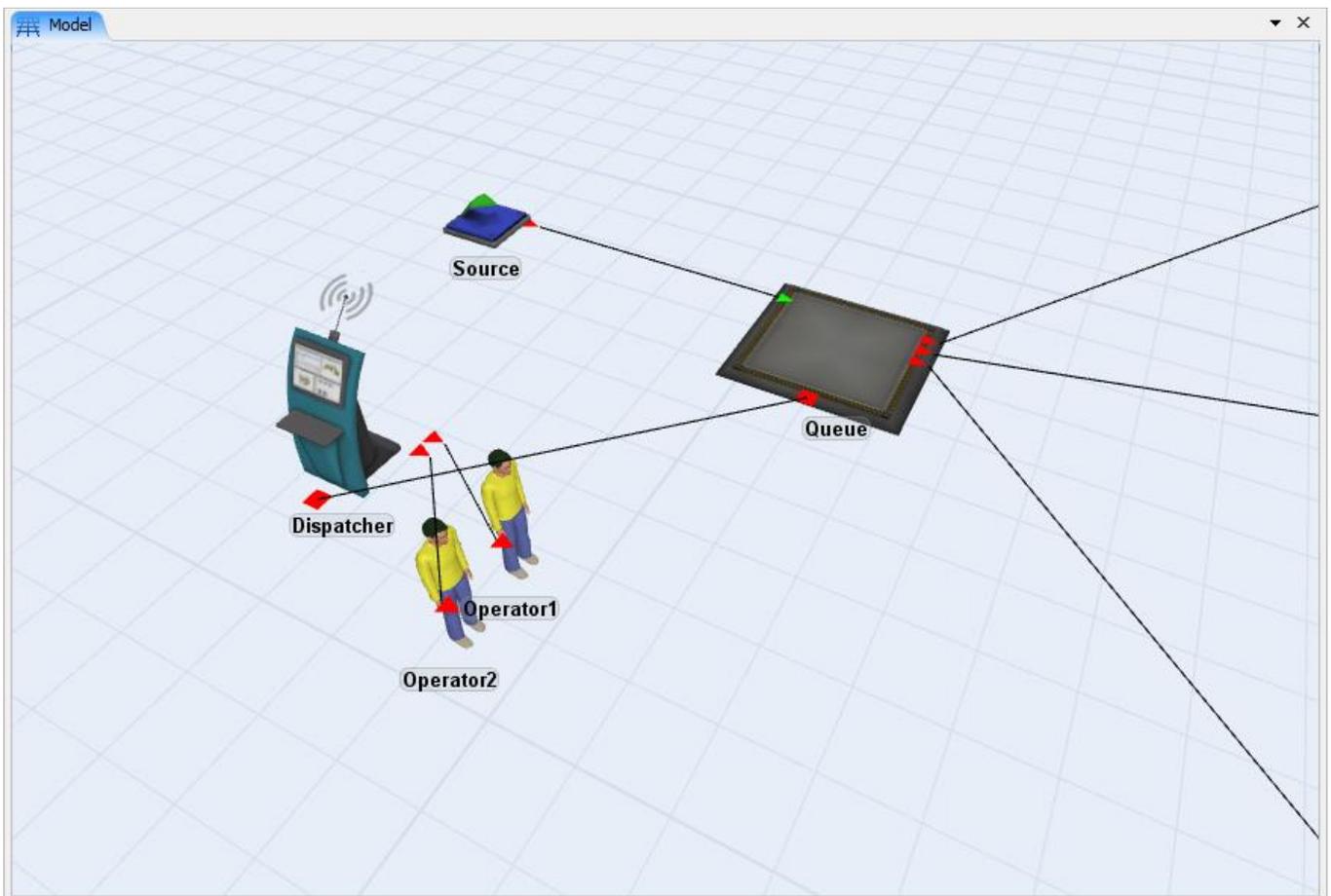
Etapa 3: Conecte os Dispatchers e os operadores

O Queue precisará de um transporte para realizar o transporte dos itens até a próxima operação. A lógica para decisão de qual Testador utilizar já foi definida no Exercício 1 e não precisará ser alterada. Será necessário apenas impor a lógica para que o Queue utilize o transporte. Sabendo que os dois operadores podem ser utilizados para realizar o transporte, é necessário a utilização de um Dispatcher para organizar as tarefas e emiti-las para aquele operador que estiver livre para executa-las. Caso tivéssemos apenas 1 operador, a conexão poderia ser feita diretamente nele, pois não precisaria de um “líder” para definir quem executará a tarefa, pois existiria apenas uma opção, o próprio operador.

Do mesmo modo que o Dispatcher é conectado a equipe de operadores, o mesmo também precisa estar conectado a porta central do objeto que fará a requisição do transporte, ou seja, do objeto que emitirá a sequência de tarefas. A porta central está fisicamente localizada no centro do objeto e na parte de baixo do mesmo. Para conectar um objeto a porta central do Queue, basta manter pressionada a tecla “S” do teclado e clicar no queue, e posteriormente clicar no Dispatcher, e depois soltar a tecla “S”. Após este procedimento, uma linha ligando os dois objetos aparecerá no modelo. Esta conexão também pode ser feita utilizando a opção Connect Center Ports do painel de ferramentas do FlexSim.

Sabendo que do mesmo modo que o Despache recebe as tarefas do Queue, ele também precisa passar essas tarefas para os operadores. Para conectar então o Dispatcehr aos operadores, basta conectar a porta de saída do Dispatcher à porta de entrada dos operadores, para que assim os operadores possam receber as sequências de tarefas. Isso precisa ser feito para todos os operadores.

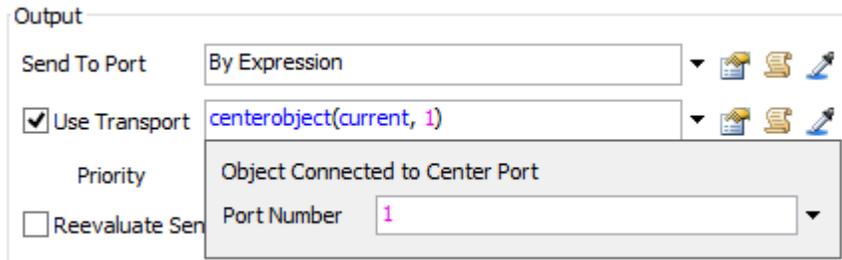
- Conecte o *Queue* ao *Dispatcher* com uma conexão central (tecla S).
- Conecte o *Dispatcher* ao *Operator1* e ao *Operator2* com uma conexão padrão (tecla A).



Etapa 4: Modifique o fluxo dos Queue para usar os (Operators)

Feito todas as conexões necessárias, basta configurar o Queue para utilizar o transporte para levar os itens até a próxima operação.

- Clique duplo no *Queue* (Fila) para abrir a sua janela **Properties**.
- Clique na guia **Flow**
- Marque a opção **Use Transport**. O campo **Request Transport From** será habilitado. Esta opção lhe permite escolher uma regra específica para definir qual objeto será utilizado para realizar o transporte do item, ou seja, qual objeto receberá a sequência de tarefas emitida pelo Queue. Neste caso será utilizado o objeto conectado a porta central número 1 (o Dispatcher), que passará a tarefa para o operador disponível, então a opção padrão é a adequada neste caso.
- Clique em **OK** para aplicar as configurações e fechar a janela de propriedades do Queue.



Etapa 5: Salve o Modelo e teste uma rodagem

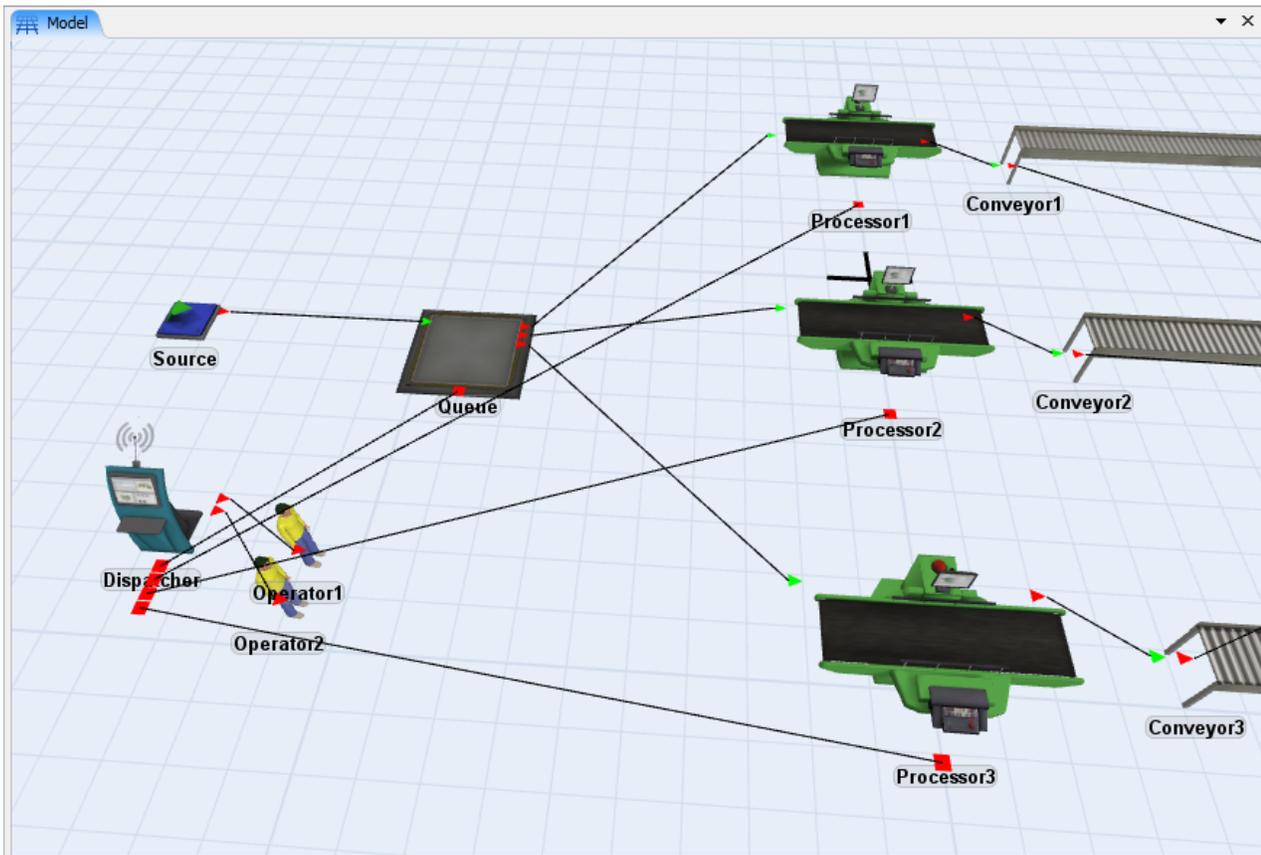
Podemos agora rodar simulação para ter certeza que o que foi feito até agora está correto.

- **Resete** o modelo e o Salve como um outro arquivo através da opção “Save Model As” encontrado no menu File ou clicando em .
- **Rode** a simulação e verifique se os operadores estão pegando os itens no Queue e os levando até os Equipamentos de Teste (Processor).

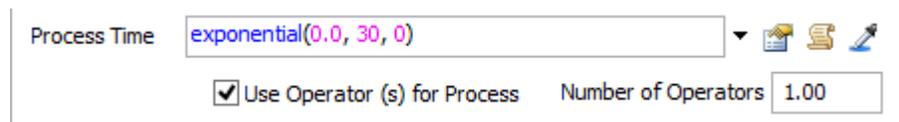
Etapa 6: Usando os operadores para os equipamentos

Para o testadores utilizarem os operadores, do mesmo modo que o Queue, cada testador (Processor) também precisará de uma conexão central com o Dispatcher. E posteriormente um parâmetro dos testadores também precisarão ser alterados para utilizar os operadores para realizar o setup.

- Conecte o *Dispatcher* ao *Processor1*, *Processor2* e ao *Processor3* com as conexões centrais (tecla S).



- Dê um duplo clique sobre o *Processor1* para abrir sua janela de propriedades (Properties)
- Na guia **Processor**, check, **Use Operator(s) for Process**. Número de operadores e Pick Operator se tornará disponível.

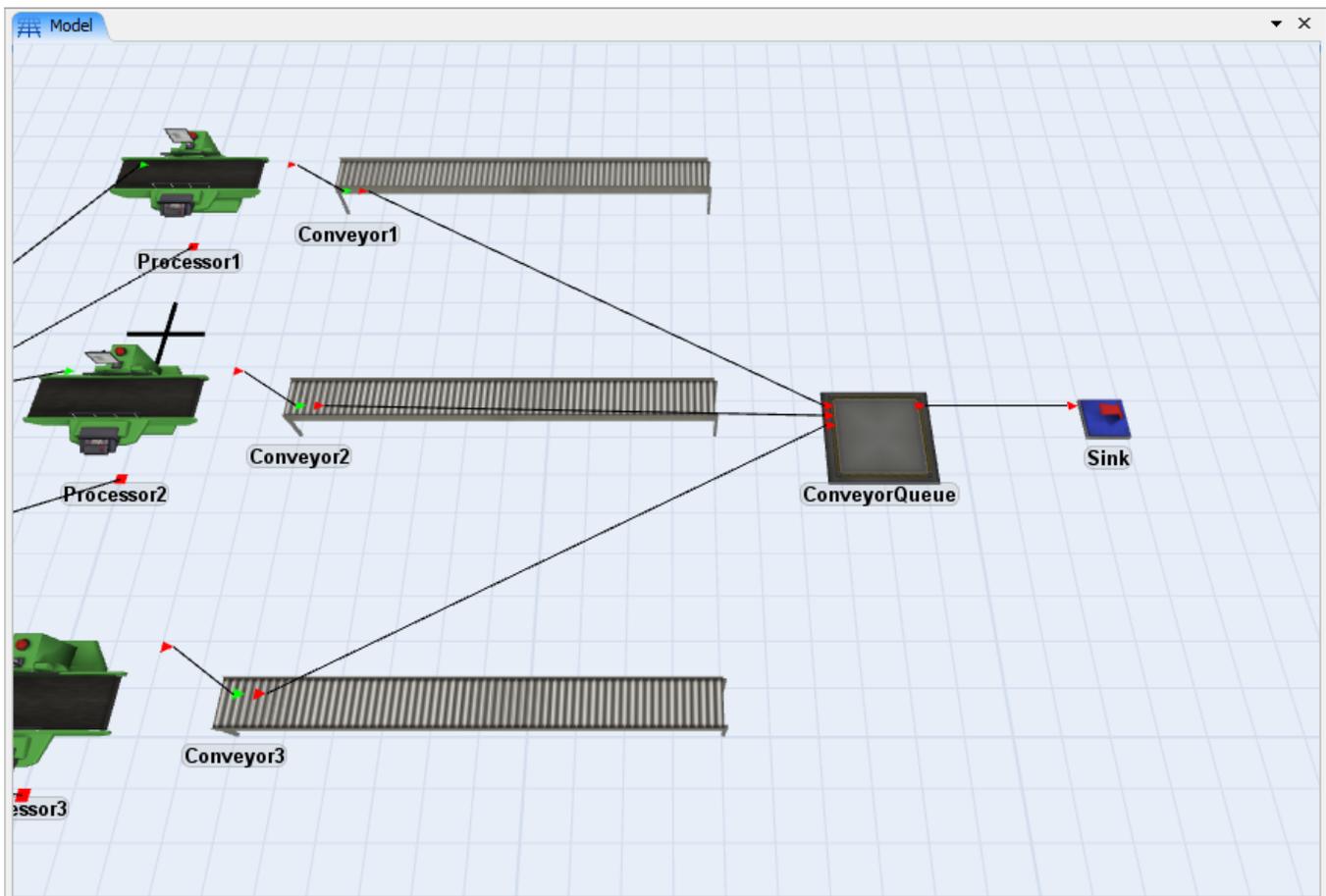


- Clique em **OK** para fechar a janela de propriedades.
- Repita esta etapa para o *Processor2* e *Processor3*.

Etapa 7: Desconectar os Conveyor do Sink

Antes de adicionar um Queue depois dos Conveyors é necessário desconectar suas portas de saída à porta de entrada do Sink. Para fazer isso, é necessário manter pressionada a tecla “Q” do teclado e clicar e arrastar do Conveyor para o Sink, nesta ordem.

- Desconecte o *Conveyor1*, *Conveyor2* e *Conveyor3* do Sink (tecla Q).
- Crie um Queue, o nomeie como *ConveyorQueue* e o coloque logo após os Conveyors.
- Conecte o *Conveyor1*, *Conveyor2* e *Conveyor3* ao *ConveyorQueue* (tecla A).
- Conecte o *ConveyorQueue* ao Sink (tecla A).

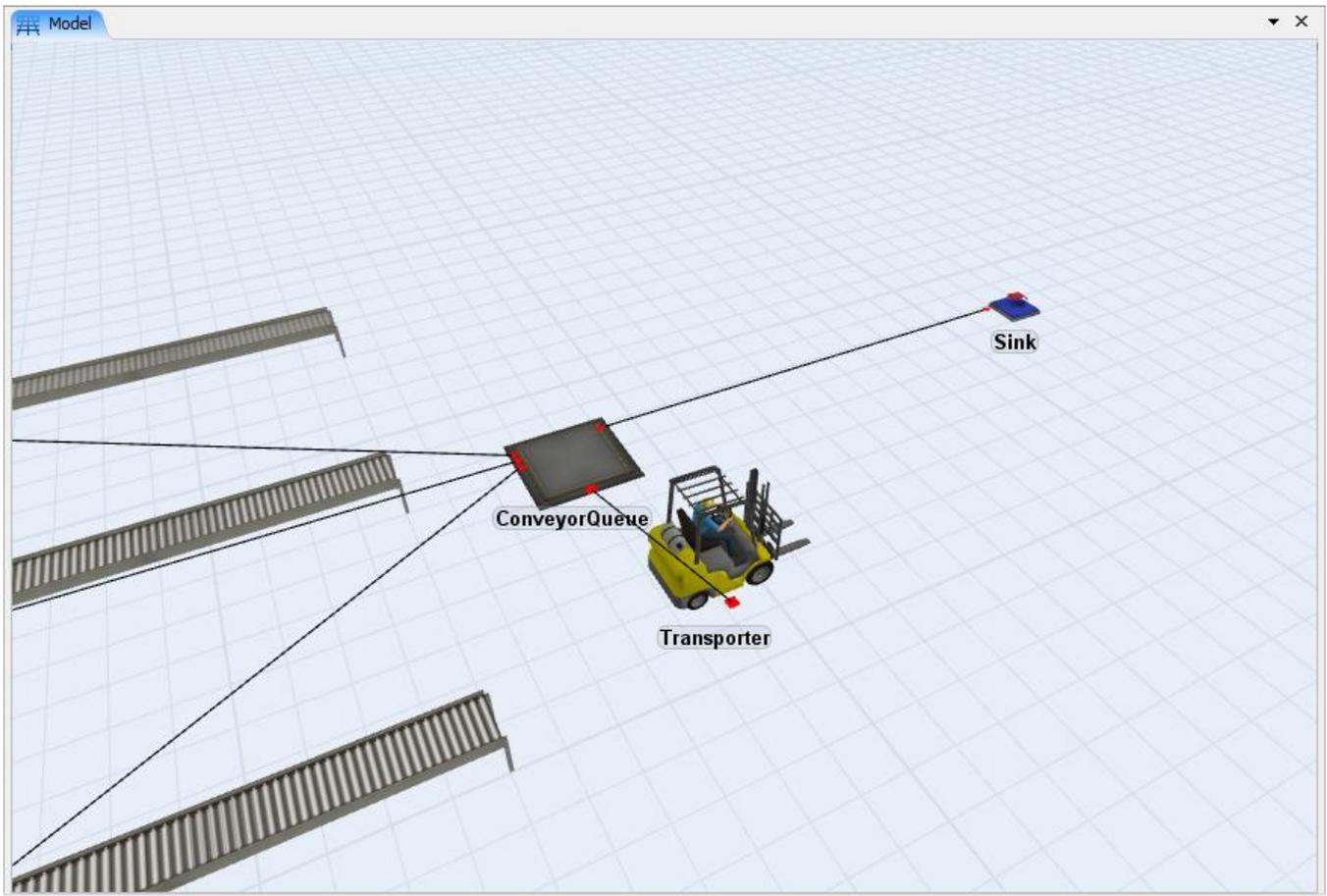


Com o layout pronto e as conexões já feitas, podemos adicionar a empilhadeira.

Etapa 8: Adicione a empilhadeira

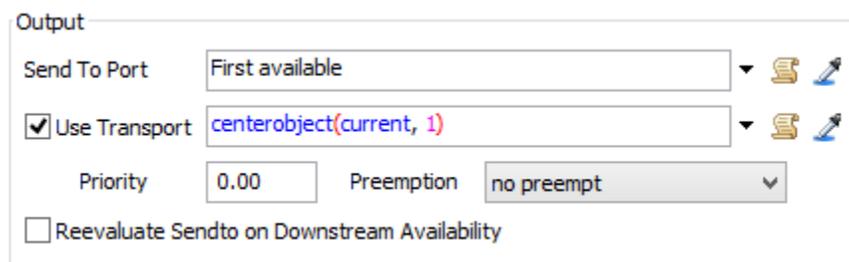
Adicione um objeto “Transporter” ao modelo para mover os itens do *ConveyorQueue* até o Sink exatamente do modo como foram criados os operadores. Porém, neste caso será adicionado apenas um Transporter, então não será necessário a criação de um Dispatcher. O Transporter será conectado direto a porta central do *ConveyorQueue*.

- Desloque o *Sink* para cerca de 10 metros à direita (10 quadrados) para simular a distância percorrida.
- Crie um Transporter, coloque-o próximo ao *ConveyorQueue*, e dê a ele o nome de *Transporter*.
- Conecte o *ConveyorQueue* ao *Transporter* através de uma conexão central (tecla S).



Etapa 9: Ajuste dos parâmetros do Queue para usar a empilhadeira

- Clique duplo no *ConveyorQueue* para abrir a sua janela **Properties**.
- Clique na Guia **Flow** e cheque a opção **Use Transport**. Como a opção padrão já é para utilizar o objeto conectado a porta central número 1, deixaremos esta opção, pois este objeto já é a Empilhadeira (Transporter).

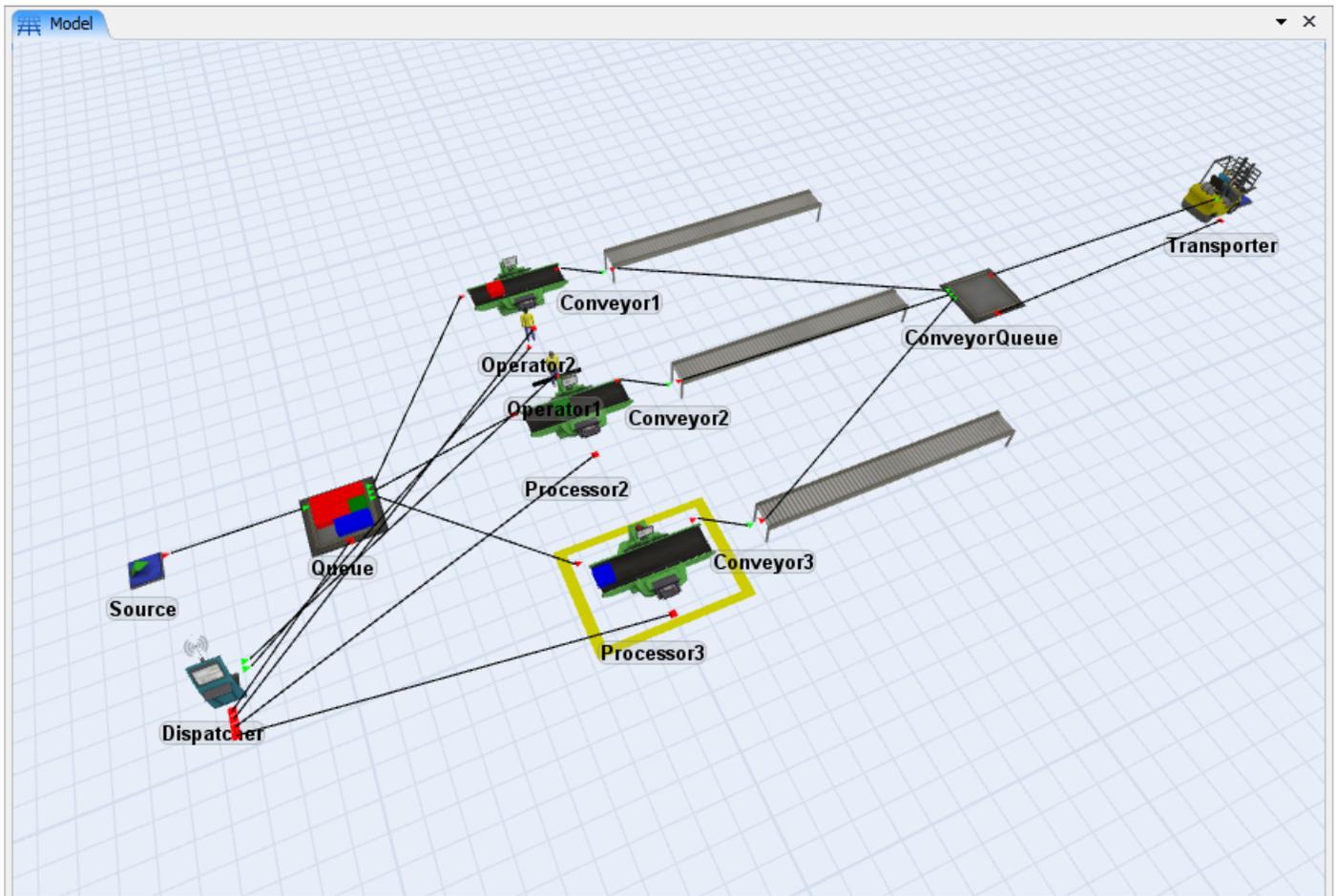


- Clique em **OK** para fechar a janela de propriedades
- **Resete** e **Salve** o modelo

Etapa 10: Execute o Modelo

- **Rode (Run)** o modelo.

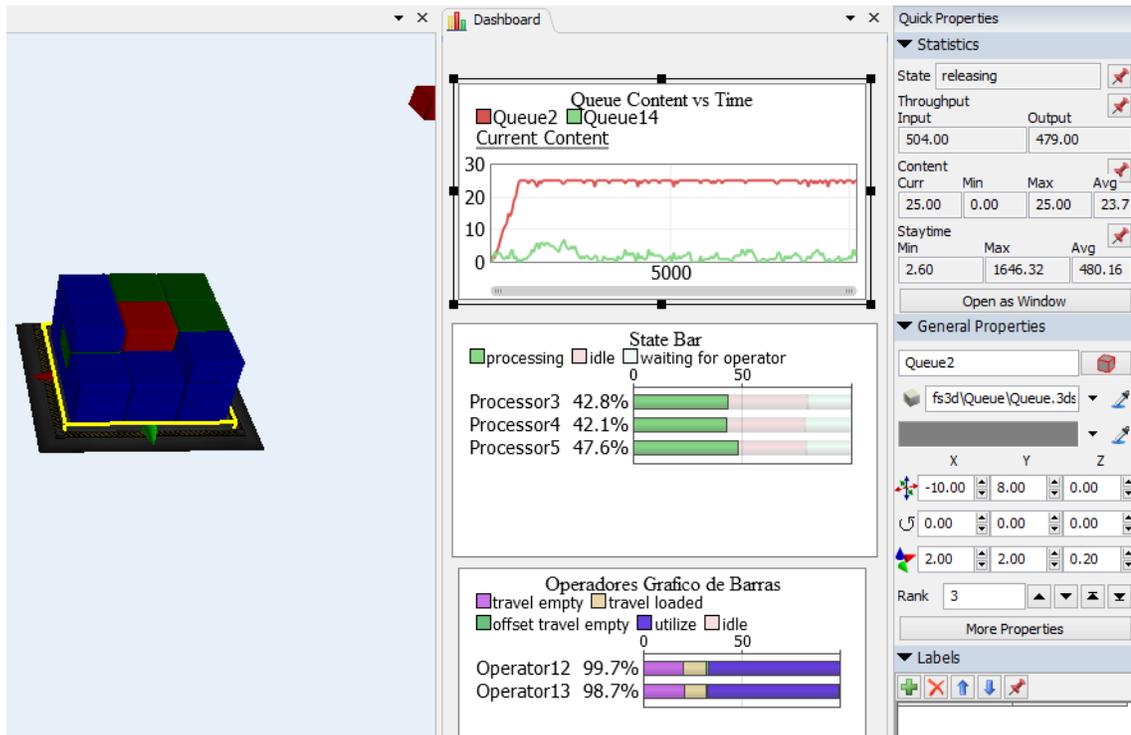
Esta é a parte recompensadora do Exercício. Este é o momento de verificar se tudo está funcionando da maneira que você quer. Durante a simulação, você pode visualizar todo o comportamento do modelo através das animações gráficas e ter a certeza que tudo está funcionando bem.



Você deverá ver os operadores se movendo entre o Queue e os Testadores (Processor), a empilhadeira pegando os itens do segundo Queue e os deixando no Sink. Você perceberá também, que quando os testadores (processors) aguardam pelo operador para iniciar os processos, os mesmos ficam com uma caixa amarela em volta, simbolizando que estão parados aguardando operador.

Etapa 11: Analise os dados de Output

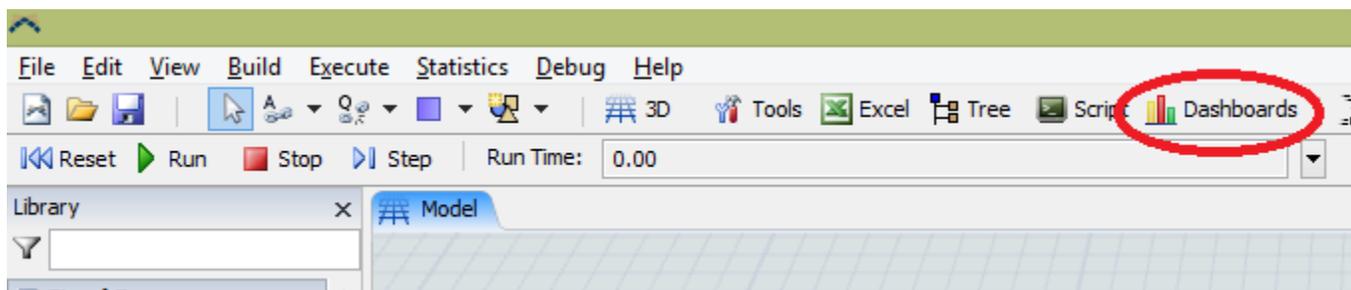
Continuando com as etapas para introdução a este modelo, ative a gravação de dados para todos os objetos deste modelo, execute a simulação novamente, e então visualize as estatísticas através clicando nos objetos do modelo e visualizando as informações estatística do lado direito de sua tela no painel Quick Properties. Abaixo estão alguns exemplos das estatísticas que foram coletadas ao longo da simulação.



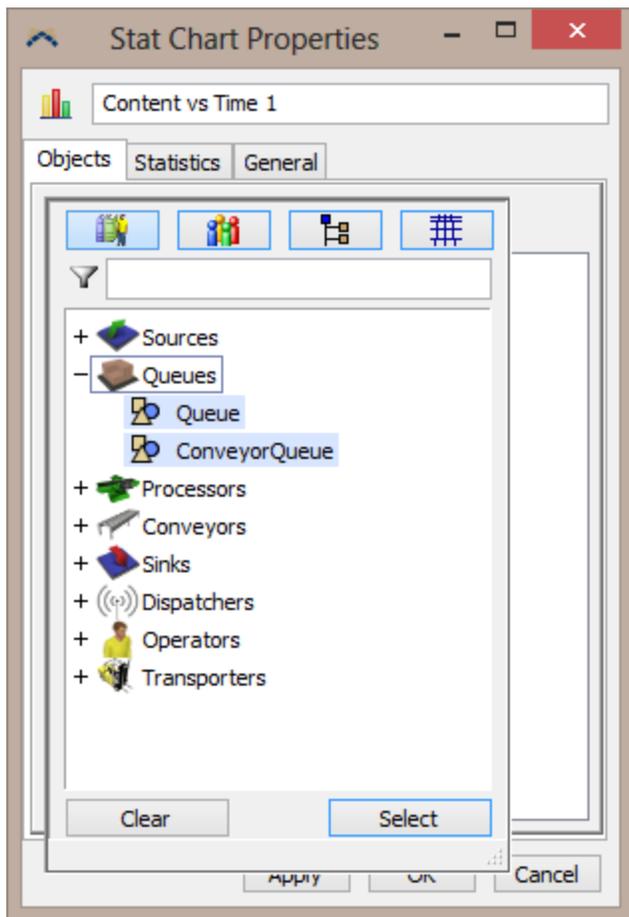
Dentre os gráficos acima temos a utilização de um dos testadores (Processors). É possível visualizar que alguns equipamentos ficaram em determinado momento ociosos. O primeiro gráfico, mostra o tempo de permanência dos itens no queue antes do Conveyor.

Mas em relação aos outros Processors, e o outro Queue? Provavelmente as informações dos operadores também seriam de grande utilidade. É possível visualizar suas estatísticas através de dados individuais de cada um deles, ou através do dashboard, que explicamos abaixo.

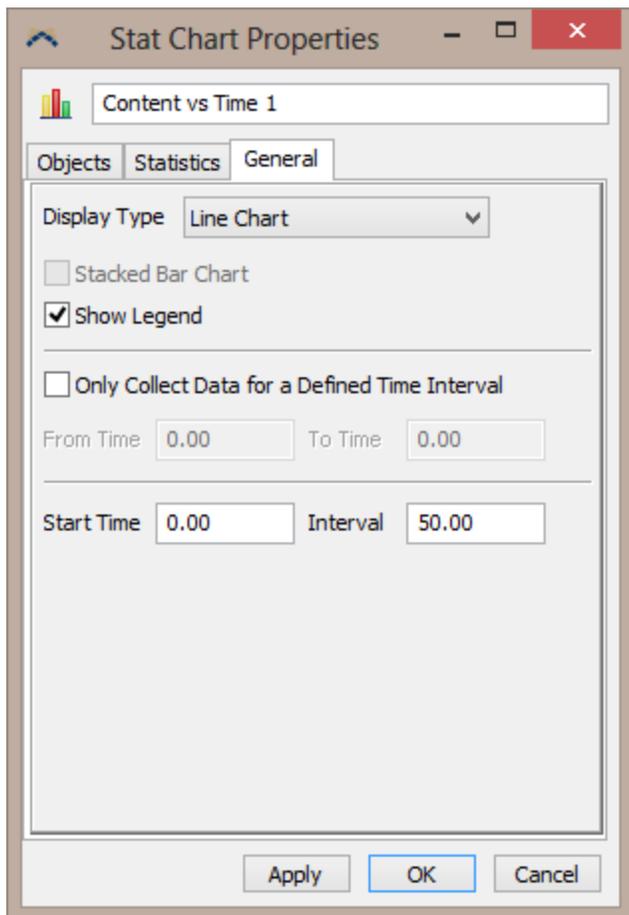
- Selecione a opção **Dashboard** conforme tela abaixo e uma área para arrastar os gráficos aparecerá.



- Arraste um gráfico de linhas clicando na opção **Content vs Time**. Arraste para a direita e solte no espaço do Dashboard. Uma janela para seleção do objeto aparecerá.
- Clique no botão para adicionar um objeto ao gráfico. Adicione os objetos *Queue* e *ConveyorQueue*. E clique em **Select**.
- Altere o nome do gráfico para **Queue Content vs Time** e clique em **OK**. Um gráfico em branco deverá aparecer no Dashboard.

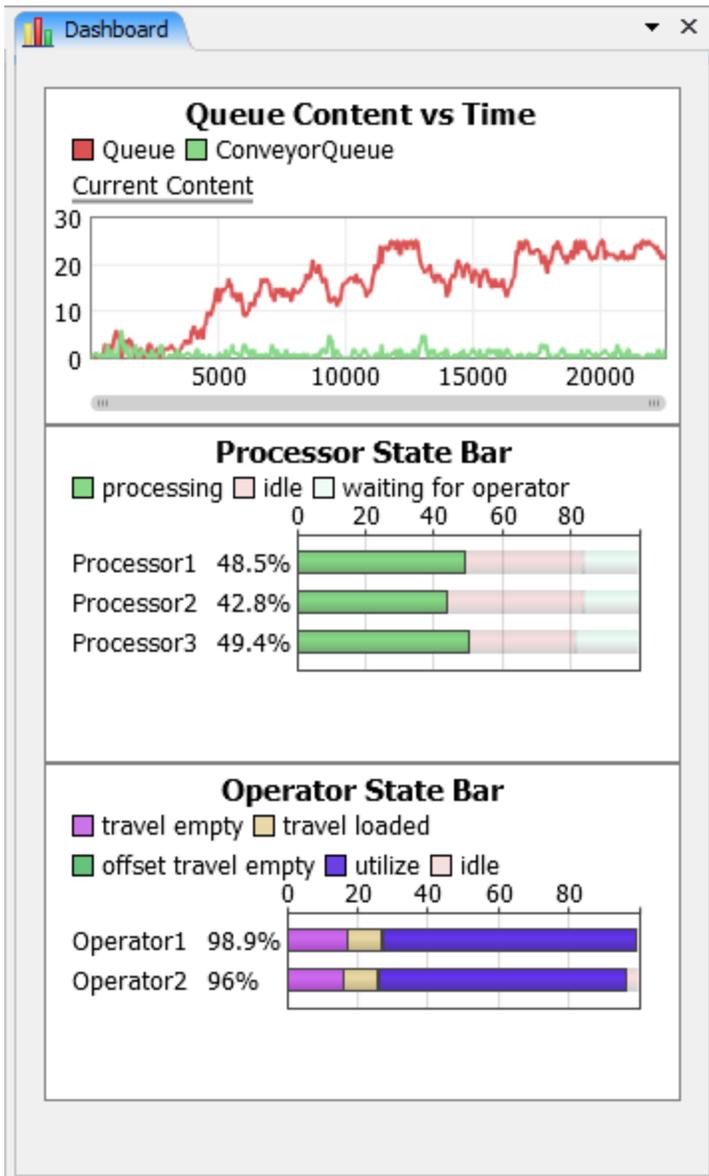


- Adicione uma legenda ao dar um duplo clique no gráfico no dashboard. Selecione a guia **General** e check a caixa **Show Legend**. Clique em **OK** para retornar ao dashboard.

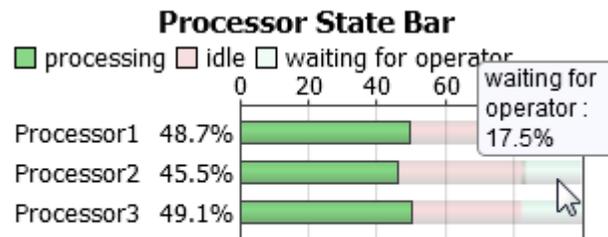


- Agora adicione um gráfico de barras clicando no botão  e selecionando a opção **State Bar**.
- Adicione todos os processors ao gráfico.
- Altere o nome do gráfico para **Processor State Bar** e clique em **OK**. Outra tela em branco deverá aparecer no Dashboard.
- Adicione um outro gráfico de barras ao Dashboard, mas agora para os operadores. Siga as mesma etapas acima, mas adicione os operadores ao invés dos processors e nomeie o gráfico como **Operator State Bar** e clique em **OK** para finalizar.

Resete (Reset) e **Rode (Run)** o modelo de simulação. Os gráficos serão atualizados a cada vez que uma atualização for feita no modelo construído



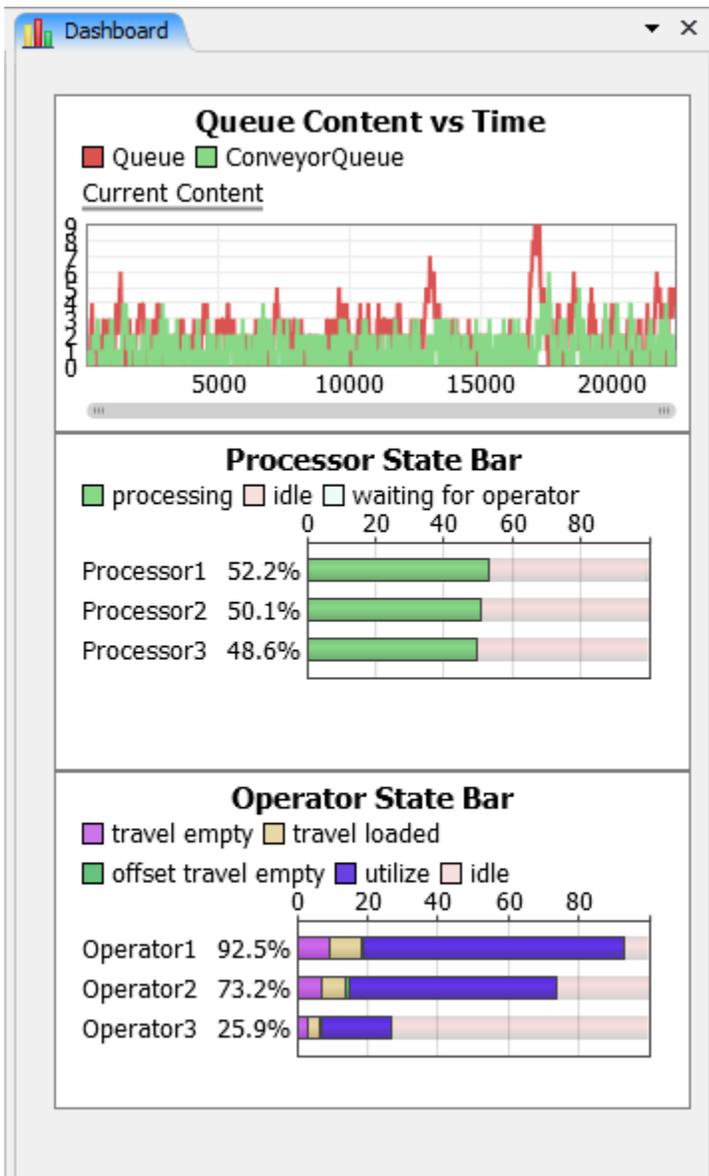
- Mantenha o mouse em cima do gráfico **Process State Bar**, e uma janela aparecerá para detalhar melhor as informações que ali são mostradas.



Obviamente, caso seja adicionado mais um operador a este modelo, o mesmo funcionará muito melhor. Mesmo que os itens permanecessem na fila de entrada, esta seria a configuração ótima com a adição de um terceiro operador.

- Crie um Operador, o nomeie como *Operator3* e o coloque próximo aos outros 2 já existentes.

- Conecte o *Dispatcher* ao *Operator3* (tecla A).
- Duplo clique no gráfico *Operator State Bar* e adicione o *Operator3*.
- **Resete (Reset), Salve (Save) e Rode (Run)** a simulação novamente.



Comum operador a mais, o tempo de espera do processos caiu consideravelmente, e o tamanho da fila também diminuiu.

Aqui a Guia o Exercício 2. Parabéns!! Já é possível ir para o Exercício Extra?

Para continuar com o Tutorial, siga para o Exercício 2 – Extra.

Exercício 2 Extra Mile

1. Introdução
2. Construção do modelo passo-a-passo

Introdução

Este exercício extra tem como objetivo demonstrar como colocar dentro do modelo alguns dados adicionais para melhor visualização do comportamento do modelo. Sendo assim, veremos como adicionar gráficos no formato 3D no Modelo 2.

O que você irá aprender

- Como adicionar um gráfico 3D para visualizar o tamanho da fila no Queue
- Como adicionar um histograma 3D para mostrar o tempo de espera no Queue
- Como adicionar um gráfico de pizza 3D para mostrar a utilização de cada operador
- Como adicionar um texto em 3D para mostrar o tempo de espera no *ConveyorQueue*
- Como posicionar os gráficos e texto para melhor visualização.

Novos objetos

Neste exercício você conhecerá um pouco mais sobre o Visual Tool e o Dashboard.

Tempo aproximado para completar este exercício

Este exercício deve levar de 20-30 minutos para ficar pronto.

Construção do modelo passo a passo

Construindo o modelo do exercício 2

Para iniciar o Modelo 2 – Extra será necessário o modelo do Exercício 2 completo que foi salvo no exercício anterior.

Caso seja encontrado qualquer dificuldade durante a construção deste modelo, existe um tutorial completo que pode ser acessado através do endereço <http://www.flexsim.com/tutorials>

Etapa 1: Abra o modelo 2

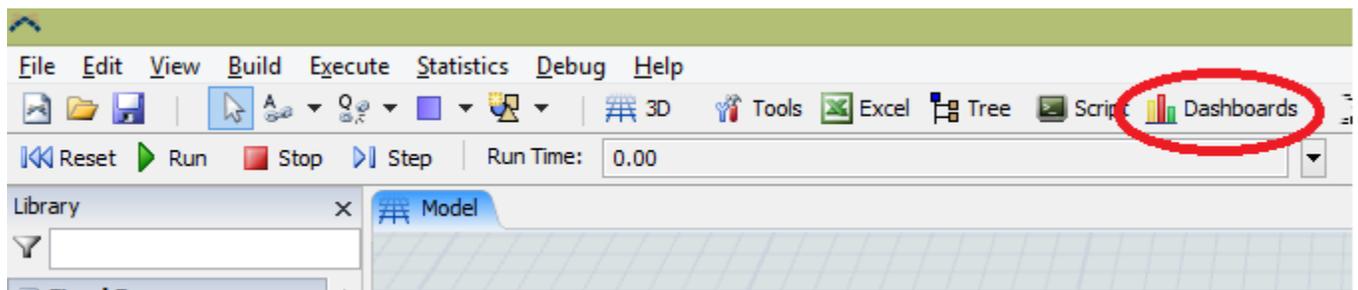
- Abra o modelo 2 se ainda não foi reaberto

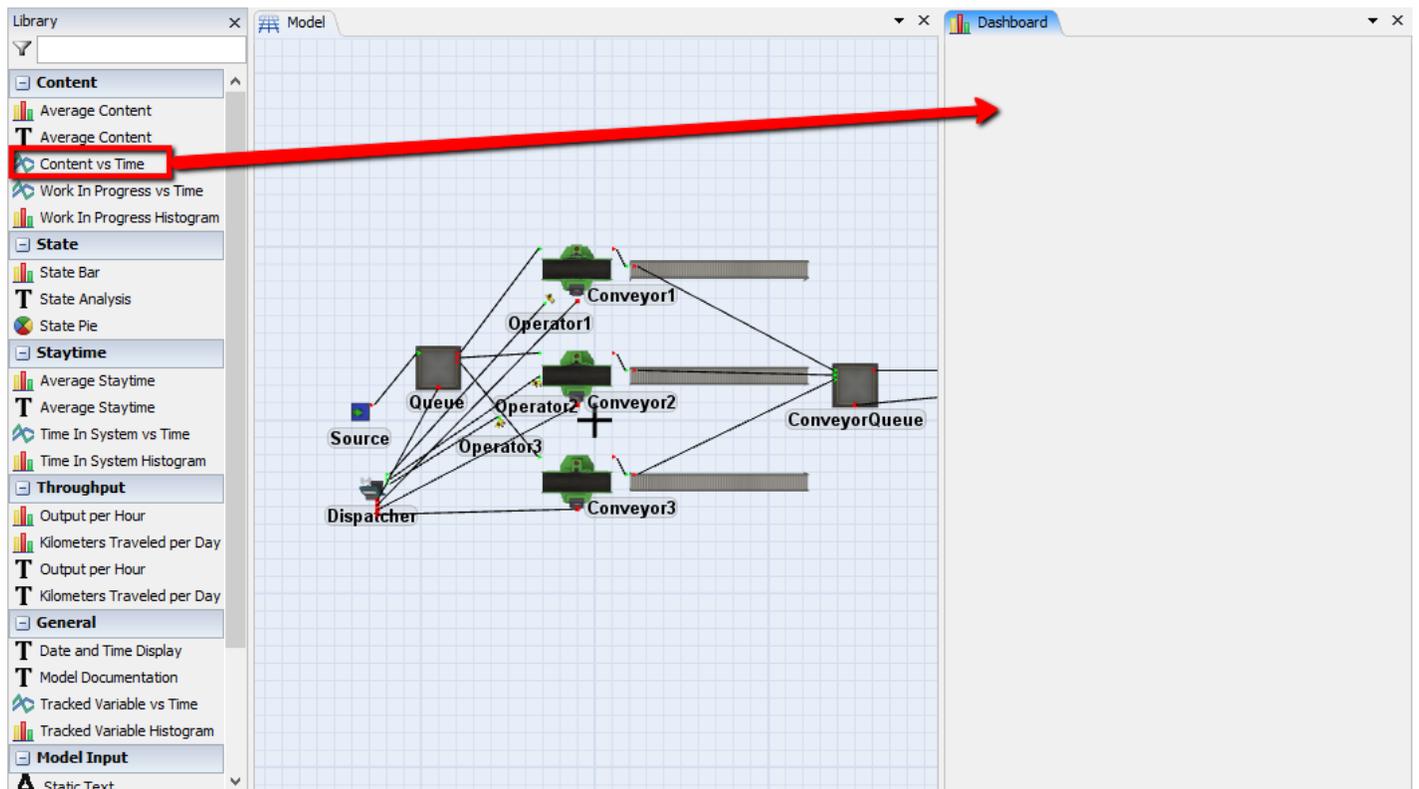
Etapa 2: Salve o modelo como "Modelo 2 exercício extra"

- Vá para o menu e em **File > Save Model As.** salve seu modelo conforme sugerido acima.

Etapa 3: Adicione um Dashboard para mostrar o conteúdo do Queue

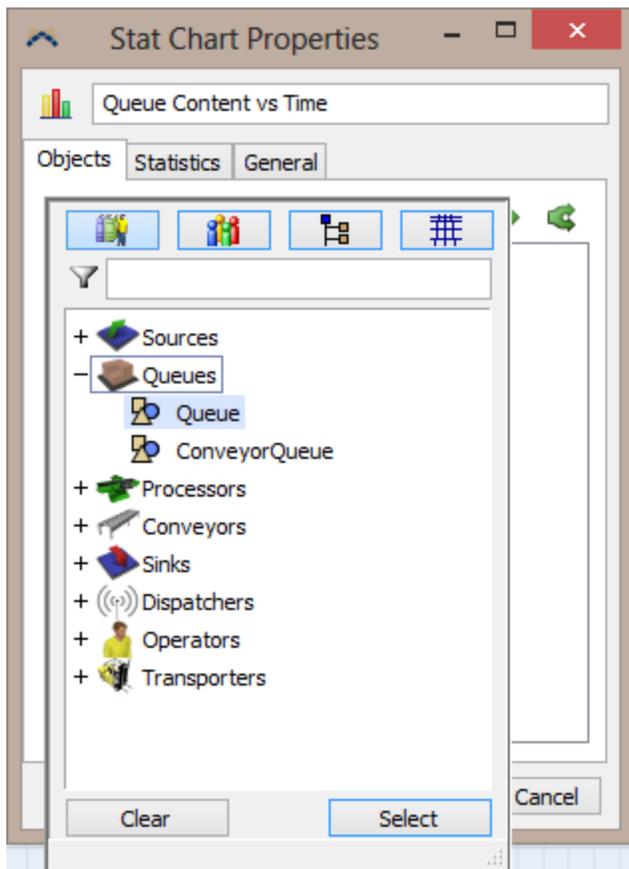
- Crie um Dashboard, selecionando "Dashboard" conforme imagem abaixo e então arraste o objeto "Content vs Time" da biblioteca para dentro do painel do dashboard.



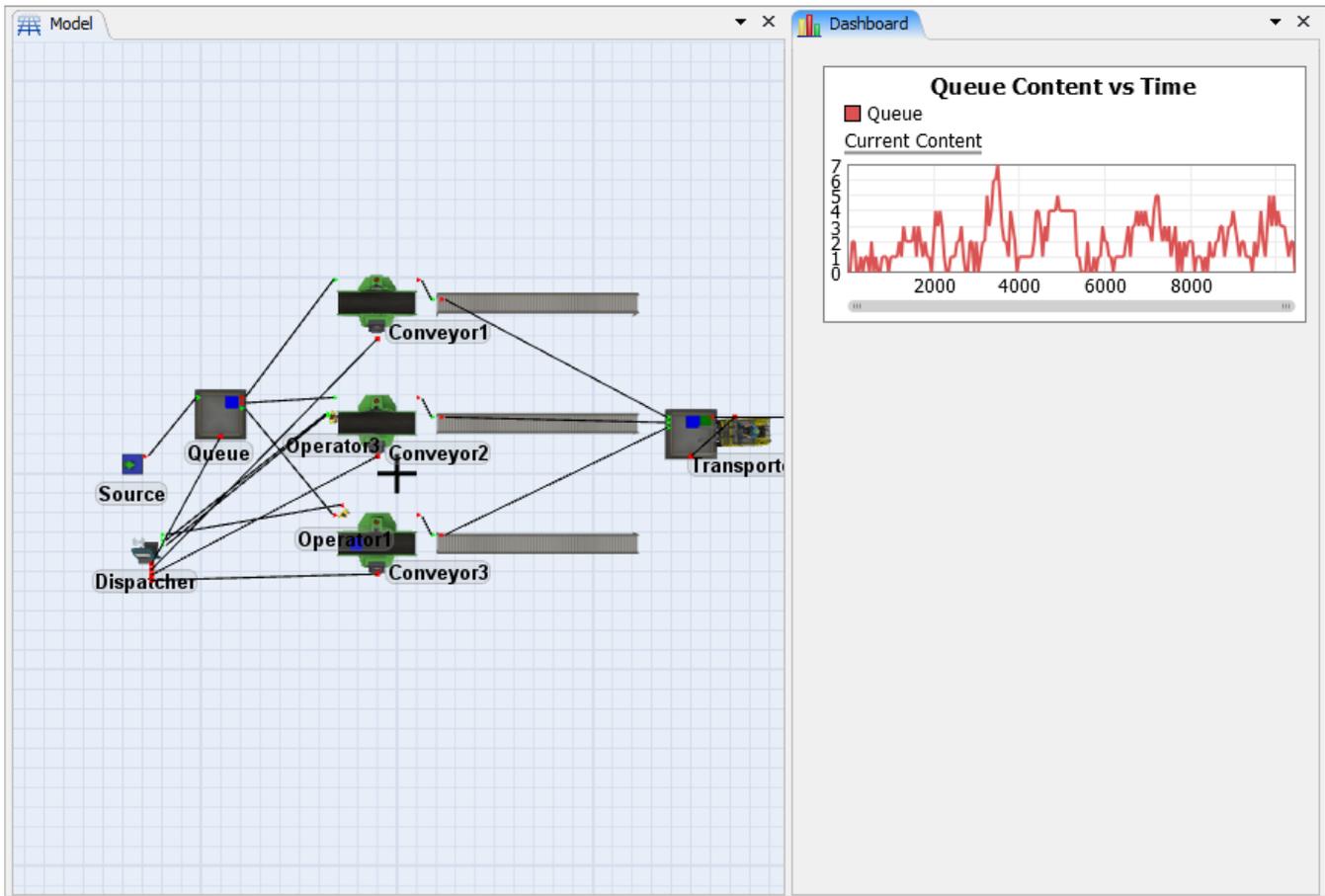


Etapa 4: Ajuste os parâmetros do Dashboard para mostrar um gráfico de conteúdo do Queue.

- Nas propriedades do **Dashboard**, clique sobre o sinal positivo em verde (**Green Plus**) na guia **Objects**. Você não será capaz de selecionar o Queue para mostrar seu conteúdo em função do tempo (**Content vs Time**).
- Clique no botão + próximo aos **Queues**, selecione **Queue**. Clique no botão **Select** para completar sua seleção.
- Altere o nome do Dashboard para **Queue Content vs Time**.
- Clique **Ok** para aplicar as mudanças e feche a janela.

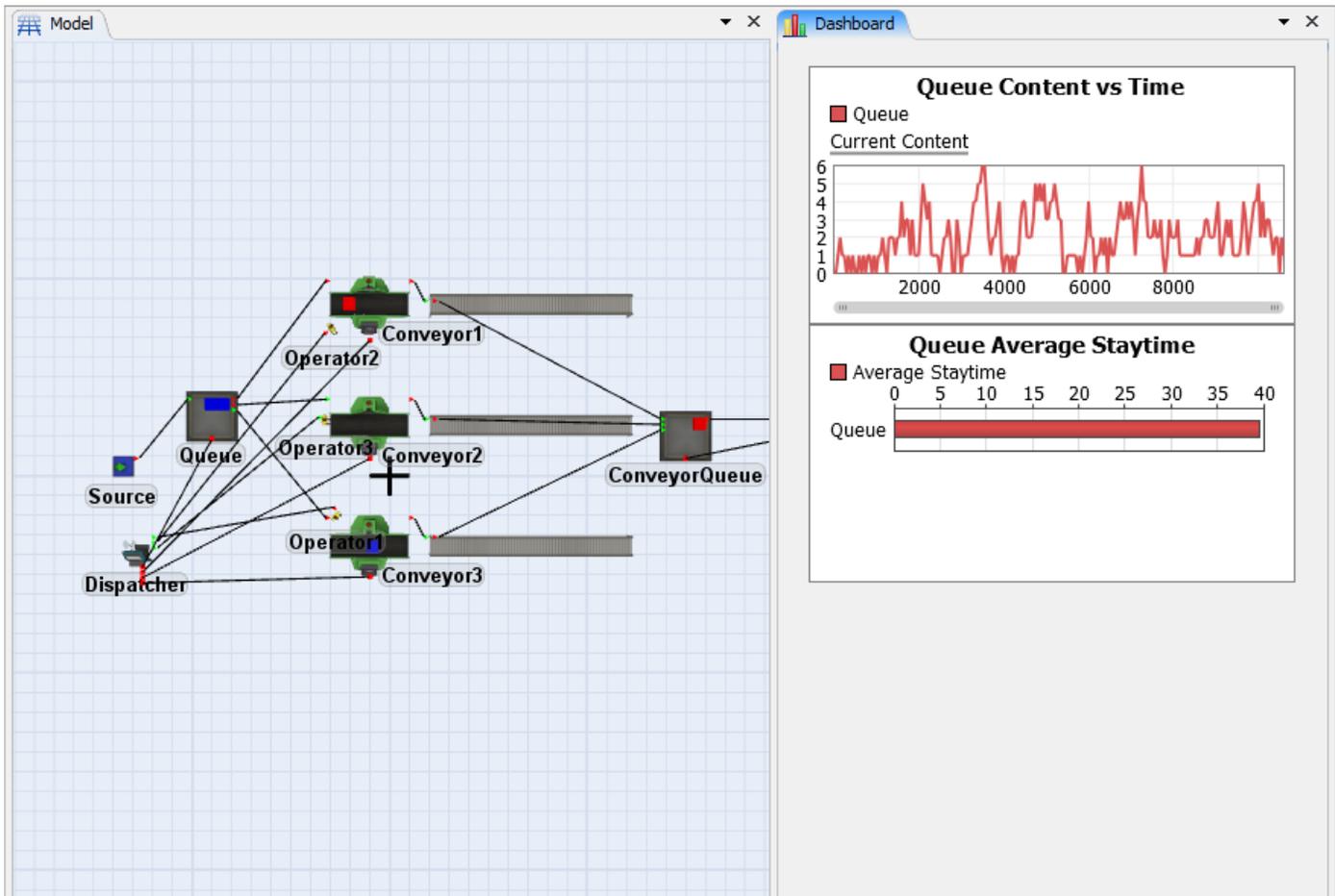


Alternativamente, você pode clicar sobre o  e então clicar sobre o Queue no ambiente em 3D para adicioná-lo ao seu dashboard.



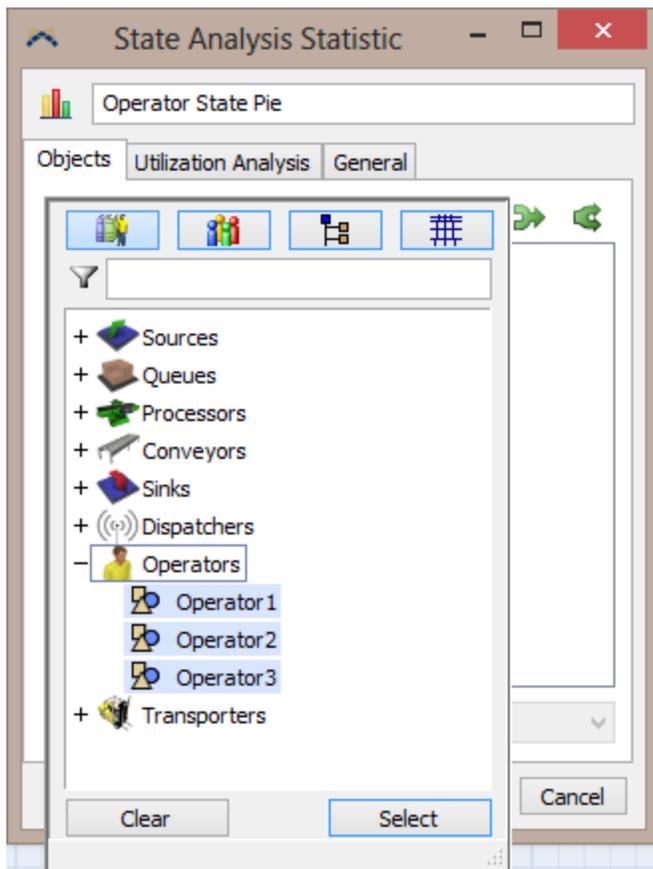
Etapa 6: Adicione um Dashboard para mostrar o tempo médio de permanência no Queue

- Siga as mesmas etapas da etapa 4, adicione um novo **Dashboard** e coloque-o Guiaixo do gráfico de conteúdo. A única diferença será que você precisa arrastar um objeto **Average Staytime Bar Graph** ao invés de “Content vs Time” da biblioteca (**Library**) e o título do gráfico (Graph Title) será **Queue Average Staytime**.

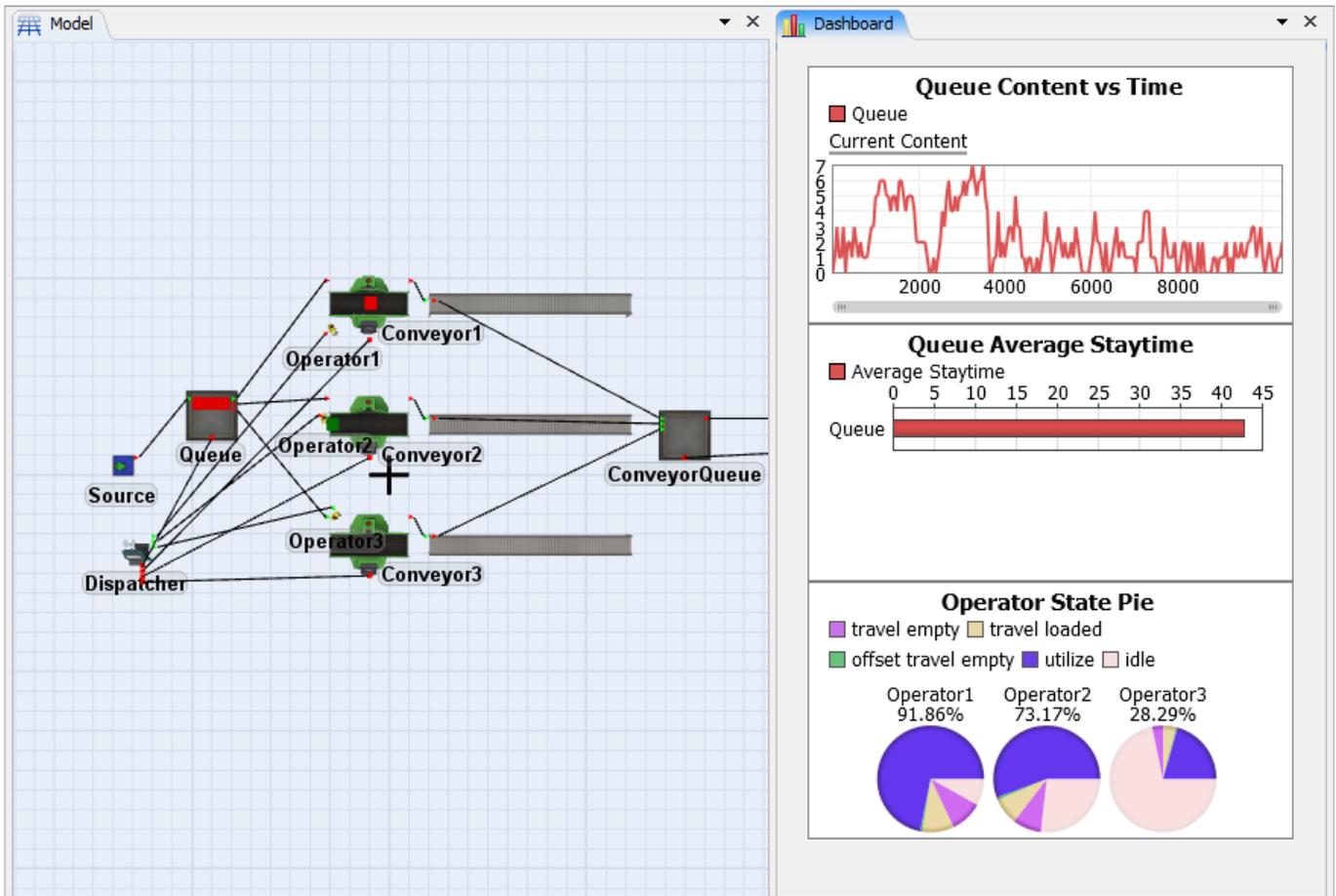


Etapa 7: Adicione um gráfico de pizza para visualização da utilização de cada operador

- Seguindo os mesmos passos da etapa 4, adicione um novo **Dashboard** para cada operador. Selecione **State Pie** ao invés de "Content vs Time" da biblioteca (**Library**) e configure o título do gráfico (**Graph Title**) para ser **Operator State Pie**.



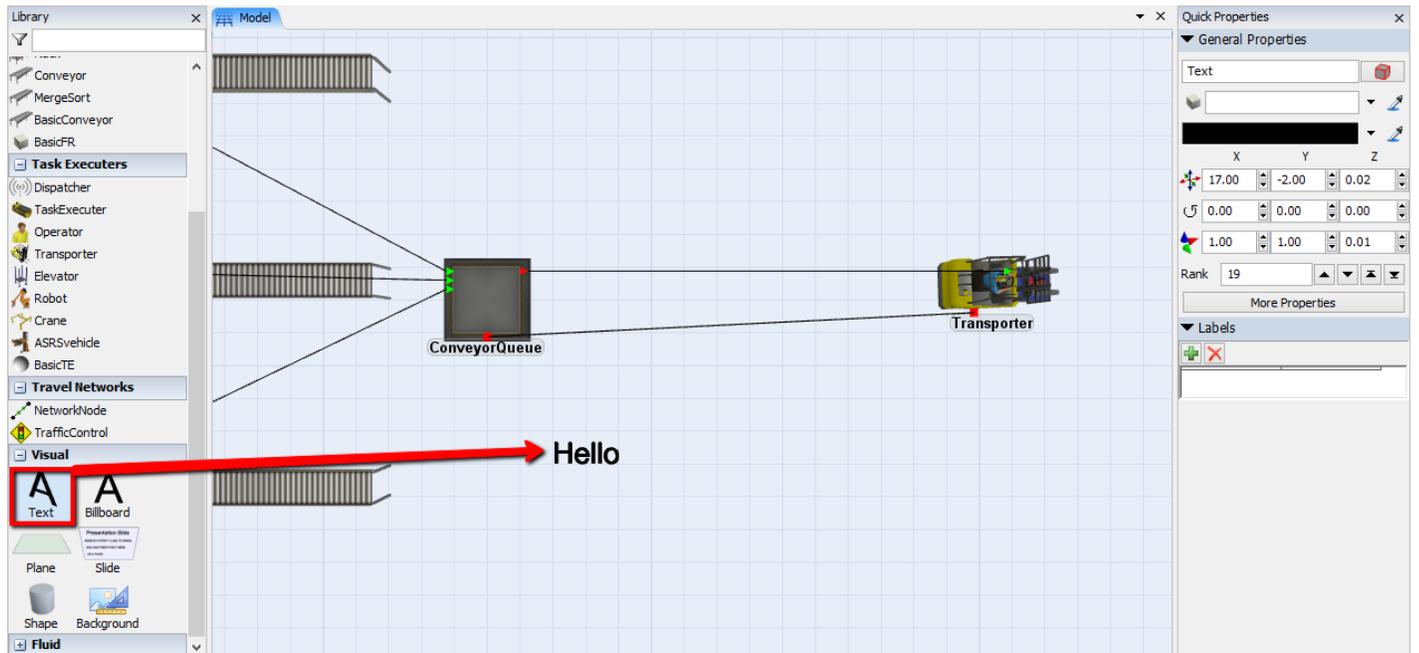
Quando resetar e rodar, os gráficos de pizza devem parecer-se com algo conforme abaixo.



Etapa 8: Adicione um texto 3D ao modelo

Existe ainda uma outra maneira de visualizar informações no modelo que podem ser usadas para mensurar seu comportamento. Textos em 3D podem ser colocados em pontos estratégicos para demonstrar o que está acontecendo enquanto o modelo está rodando. Para isso, é utilizado o objeto "Visual Tool". Neste modelo utilizaremos esta ferramenta para visualizar o tempo médio de espera dos itens no "Conveyor Queue".

- Adicione uma **VisualTool** ao modelo, o nomeie como *Text* e o coloque próximo ao **ConveyorQueue**.

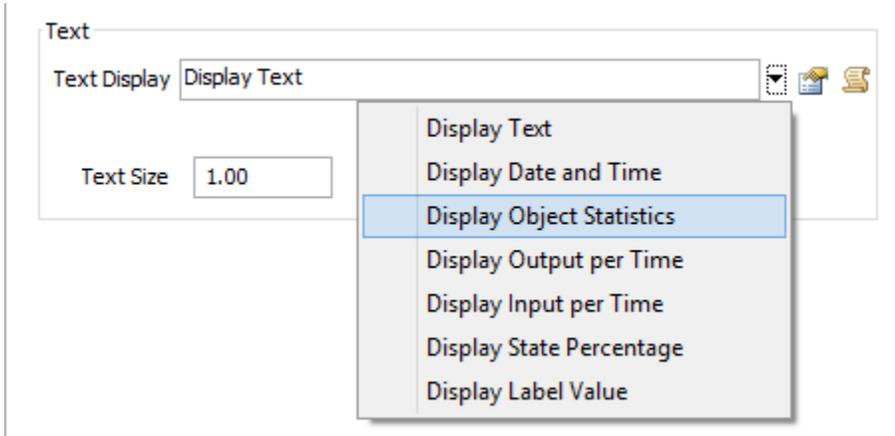


- Dê um duplo clique sobre uma **VisualTool** para abrir sua janela de propriedades (**Properties**).
- No campo Text Display selecione a opção **Display Object Statistics**. A janela do template de código irá aparecer. Altere os textos dos parâmetros para parecer-se conforme abaixo:

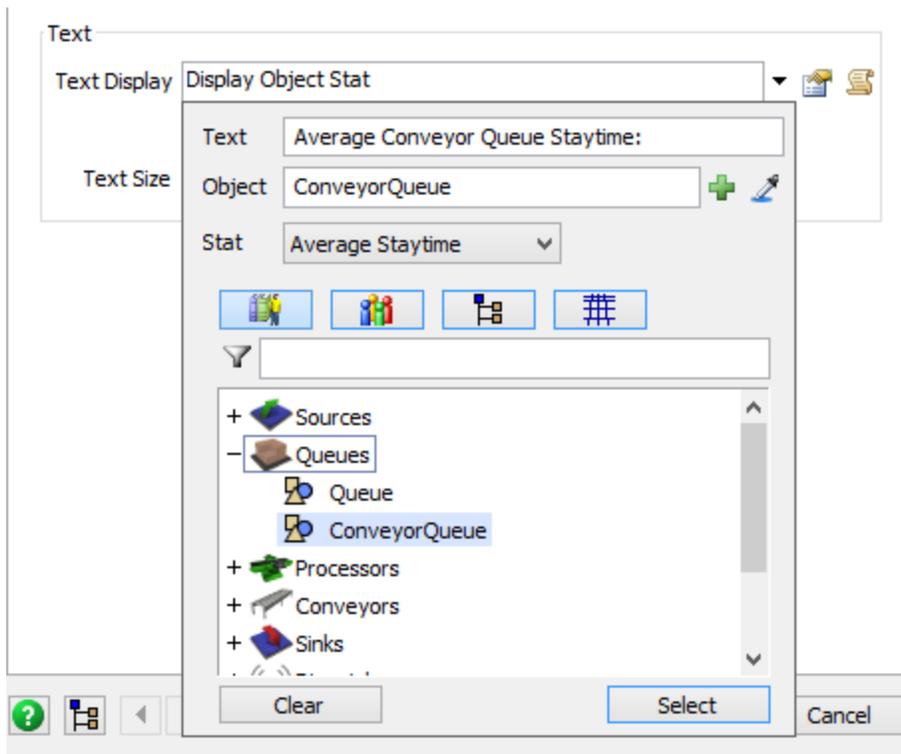
Text: "Average Conveyor Queue Staytime: "

Object: centerobject(current,1)

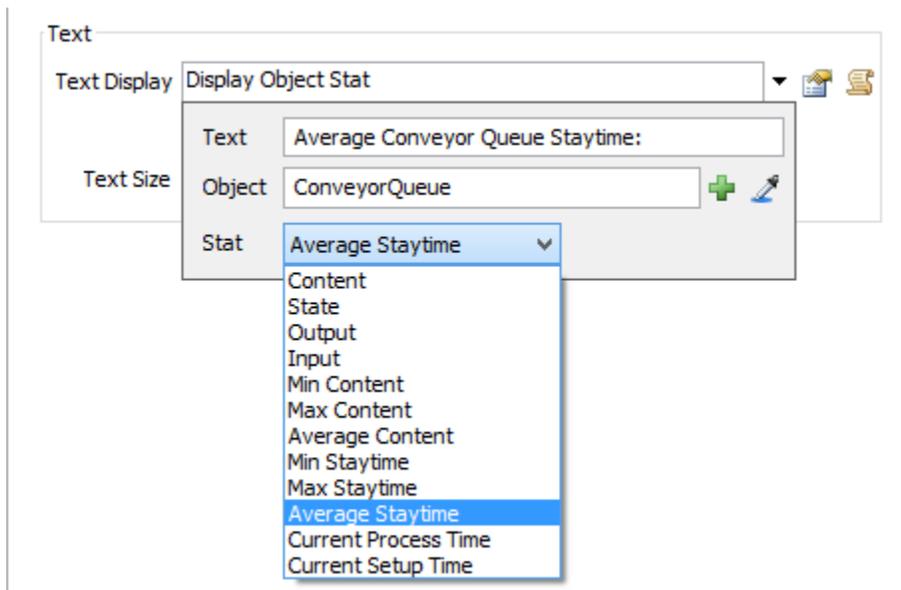
Stat: Average Staytime



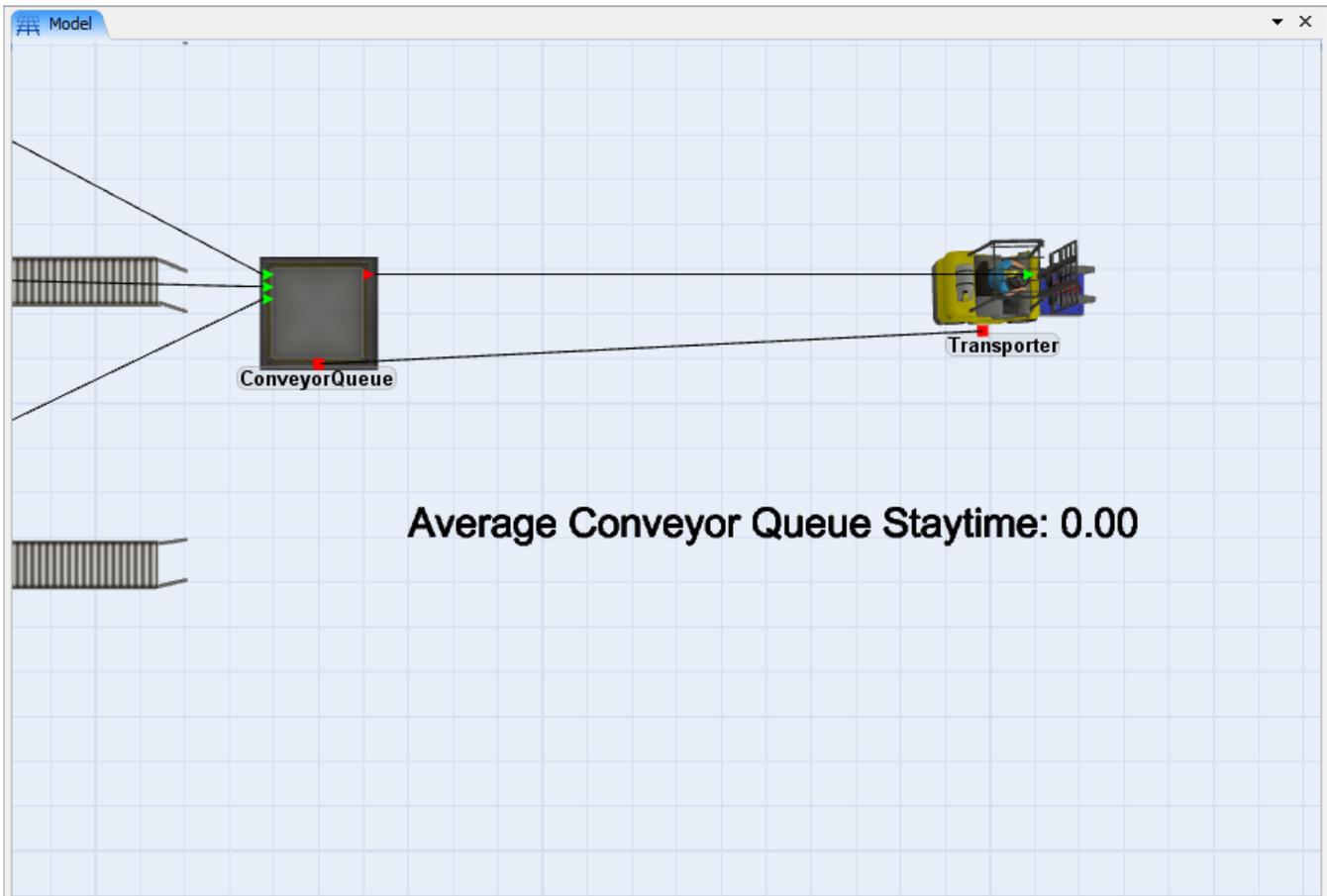
- Para o objeto (**Object**): valor, clique no sinal em verde, então clique no + próximo ao **Queues**, e selecione **Average Staytime**. Então pressione **Select**.



- Para a estatística (**Stat**): valor, clique no menu drop down, e selecione **Average Staytime**.

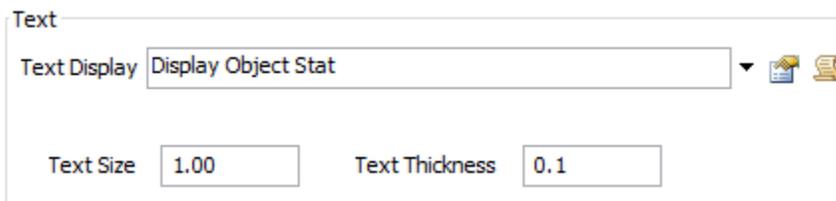


NOTA: Para clicar sobre a VisualTool, clique diretamente sobre o texto em 3D que está sendo mostrado.

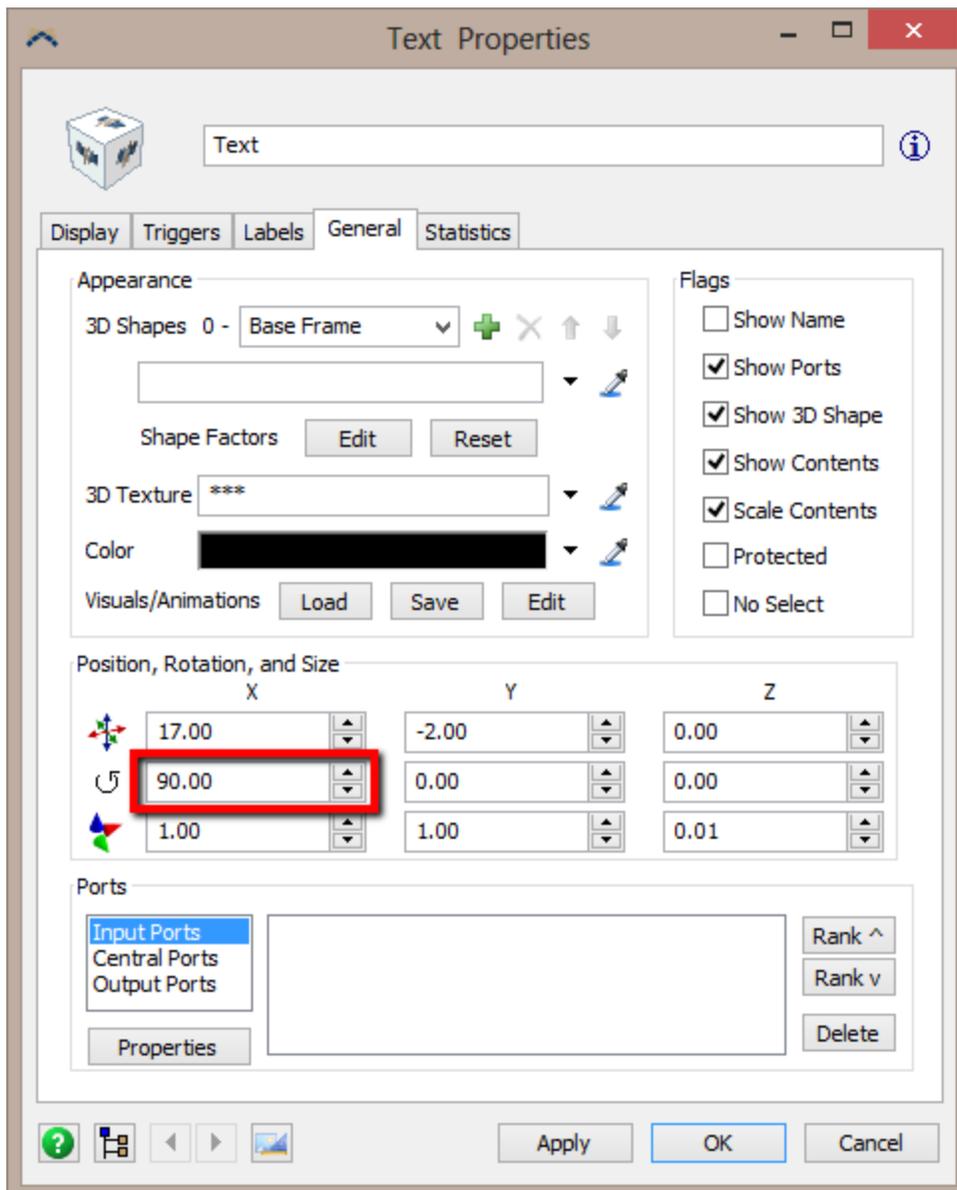


Nesta etapa você já deve estar querendo ajustar o modo como o texto está na tela. Por padrão, o tamanho deste texto é definido como 1, mas você pode querer deixá-lo menor. Pode também desejar que o texto fique suspenso no ar sobre o Queue. Para diminuir o tamanho do texto, digite o tamanho desejado no campo **Text Size** na janela de propriedades do Queue. É possível também ajustar a espessura das letras do texto para dar-lhe melhor aparência de um texto em 3D

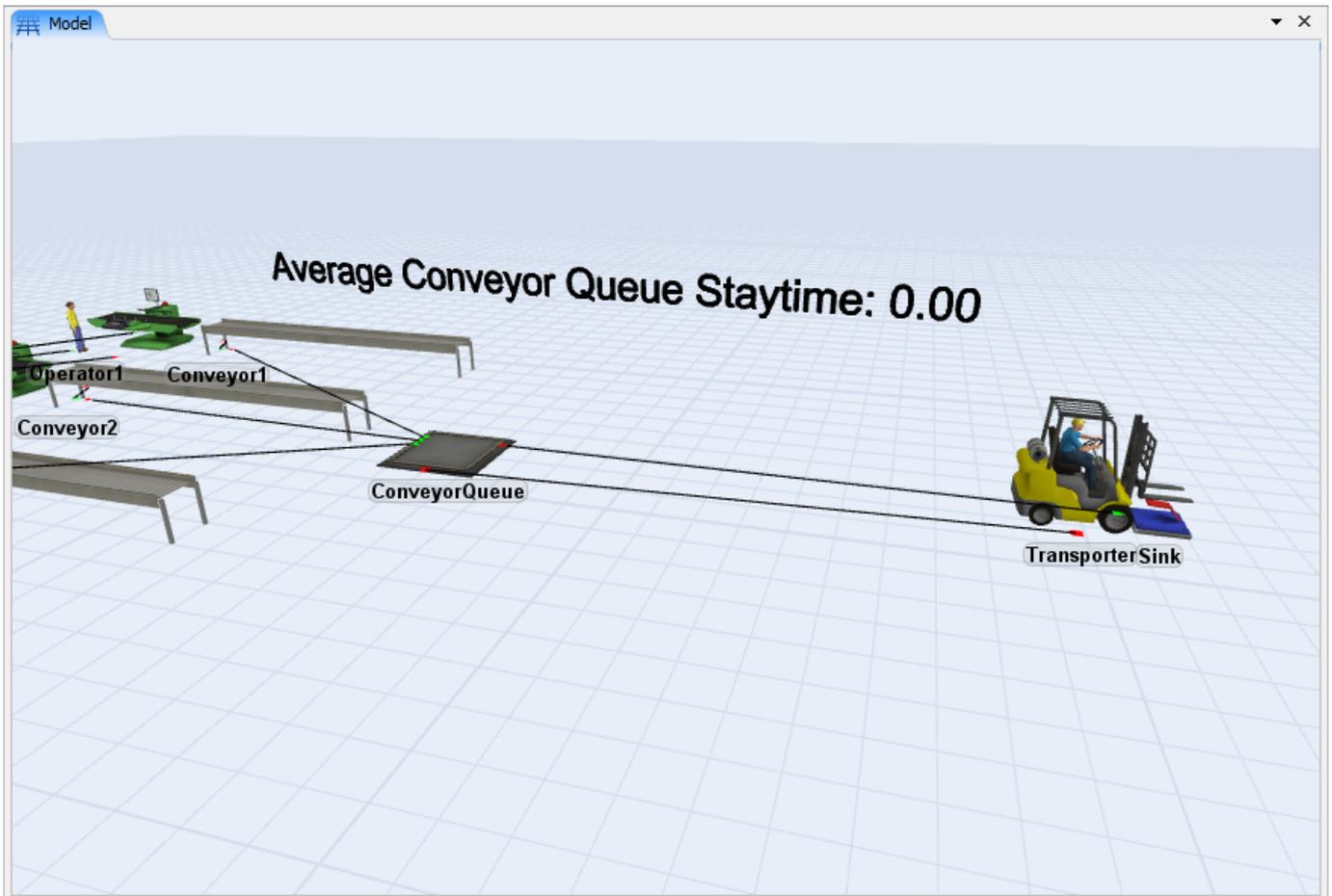
- Clique duplo sobre a Visual Tool para abrir sua janela **Properties**. Na Guia **Display**, altere a variável **Text Size** para 1 e a variável **Text Thickness** para 0.1.



- Clique na guia **General**.
- Altere a cor através do campo **Color** para preto.
- Altere o campo **RX** para 90.
- E clique em **OK** para fechar as propriedades do objeto e aplicar as alterações feitas.

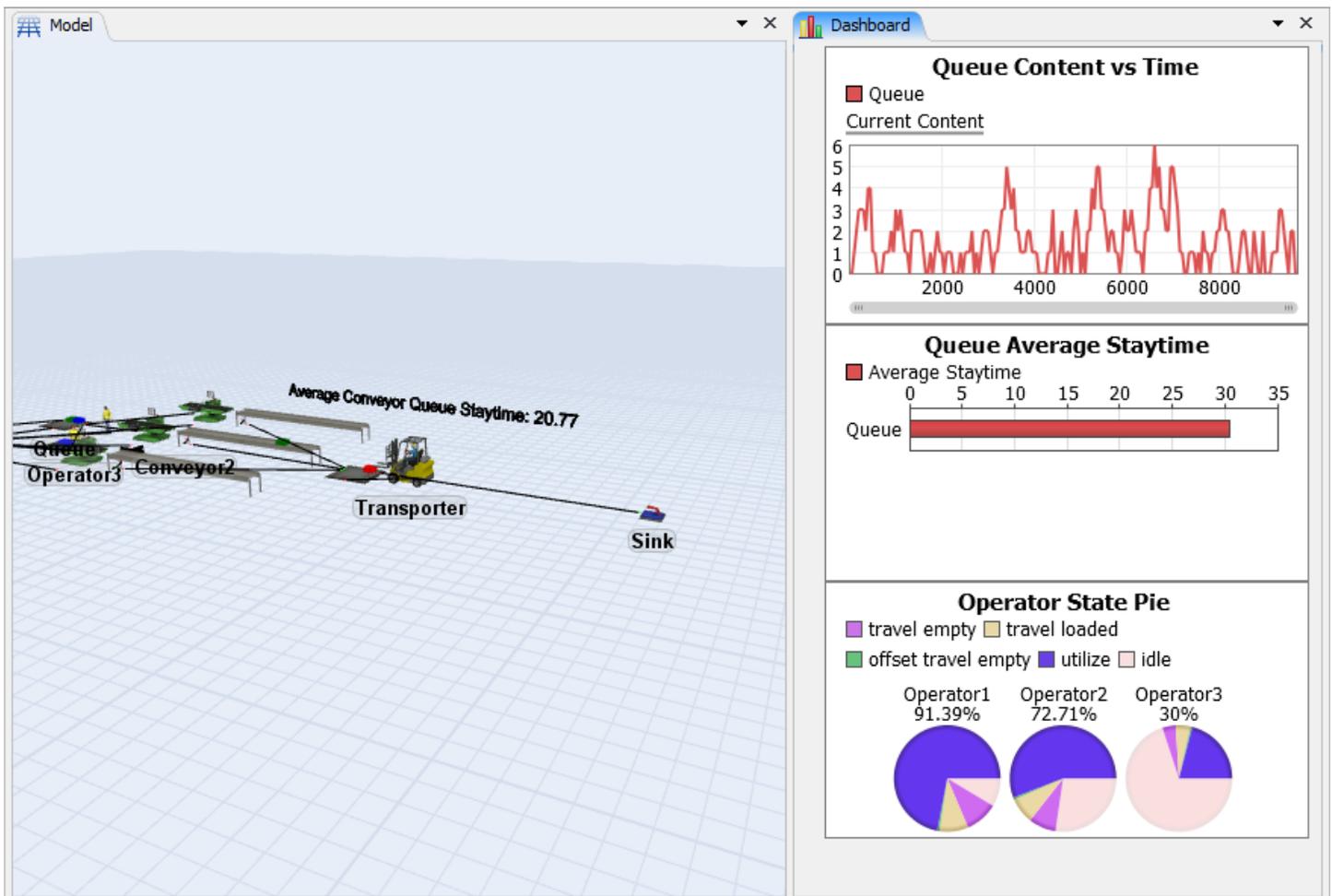


O texto agora está rotacionado no modelo. Use o mouse para selecionar e reposicionar o texto conforme desejado. Lembre-se que a posição no eixo z pode ser alterada selecionando o objeto com os dois botões do mouse simultaneamente e movimentando para baixo ou para cima, ou selecione o objeto e role o scroll do mouse para movimentar o texto para cima ou para baixo.



Etapa 9: Resete, Salve e Execute (Reset, Save and Run)

- **Resete** e **Salve** o modelo. Feito isso, você estará pronto para **Rodar** a simulação e visualizar seu comportamento através dos gráficos e textos 3D adicionados ao modelo



Isso finaliza o "Exercício 2 - Extra". Como você viu, é extremamente fácil adicionar gráficos bastante úteis para reportar informações estatísticas importantes para análise do modelo.

Para continuar com o Tutorial, siga para o Exercício 3.

Exercício 3 Tutorial

1. Introdução
2. Construção do modelo passo a passo

Introdução

O exercício 3 tem como objetivo introduzi-lo aos objetos **Rack e NetworkNode**. Você terá a chance de trabalhar com splines, conveyors, estatísticas avançadas e global tables. Outro ponto a ser demonstrado através do Exercício 3 é o *Experimenter*, do qual lhe permite simular diversas replicações ao mesmo tempo e posteriormente analisar e comparar os resultados destes múltiplos cenários simulados. O Exercício 3 utilizará o modelo desenvolvido no Exercício 2 como ponto inicial. Tenha certeza do que foi feito nos Exercícios 1 e 2 antes de iniciar o Exercício 3.

O Exercício 3 assume que você já está familiarizado com as janelas de propriedades utilizadas no Exercício 1 e 2. Nos exercícios anteriores, quase todas as etapas estavam ilustradas de modo que você tivesse certeza do que estava sendo feito para construção do modelo. Já no Modelo 3, algumas etapas simples, como por exemplo, a criação de um objeto no modelo arrastando-o para a área de trabalho, alteração de variáveis simples, etc, serão apenas descritas nas etapas, mas não serão ilustradas.

Nota: Caso você esteja usando a versão de avaliação, não será possível completar este modelo. Este exercício excede o número de objetos permitidos na versão de avaliação.

O que você irá aprender

- Como utilizar a Global Table para definir rotas dos produtos
- Como definir um trajeto específico para uma Empilhadeira
- Como definir os sentidos de deslocamento de uma rota através dos splines
- Como criar um relatório customizado
- Como rodar diversas simulações simultaneamente no mesmo modelo

Novos Objetos

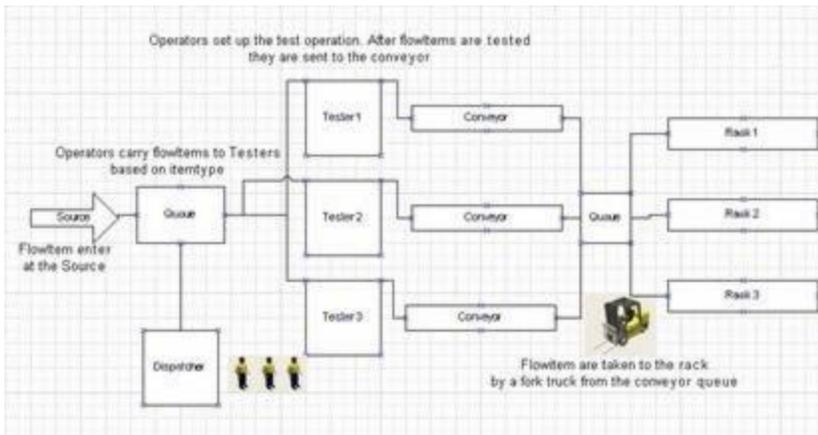
Neste modelo você será introduzido aos objetos Rack e NetworkNodes, assim como os pontos entre os nós, conhecidos como Spline.

Tempo aproximado para completar este exercício

Este exercício deverá demorar entre 45-60 minutos para ser completado

Overview do Modelo

No Modelo o Sink será substituído por 3 Racks que serão utilizados como um estoque para armazenar os itens produzidos antes da expedição (Figura 3-1). O layout dos conveyors 1 e 3 deverão ser alterados fisicamente para que os itens sejam conduzidos até o *ConveyorQueue*. Utilizando uma global Table como referência, os itens do tipo 1 serão direcionados ao Rack 2, itens do tipo 2 para o Rack 3 e itens com o itemtype igual a 3 irão para o Rack 1. E utilizando os objetos NetworkNode, você definirá um trajeto específico para a empilhadeira percorrer enquanto pega os itens do *ConveyorQueue* e os leva até os Racks. Depois disso você definirá uma simulação com várias replicações através do *Experimenter* para visualizar a variação estatística do modelo e assim calcular o intervalo de confiança dos indicadores chaves de performance (KPIs).



Dados do Modelo 3

Modificar o layout dos conveyors 1 e 3 para que sua extremidade fique direcionada ao *ConveyorQueue*.

Definir o destino dos produtos nos racks: Utilizar uma Global Table para definir os destinos de acordo com os tópicos abaixo:

- Itemtype 1 para o rack 2
- Itemtype 2 para o rack 3
- Itemtype 3 para o rack 1

Defina um trajeto específico para a empilhadeira se deslocar entre o *ConveyorQueue* e os Racks.

Defina um *flypath* para um voo através do modelo para apresentação.

Novos Conceitos

Pontos de Controle Spline

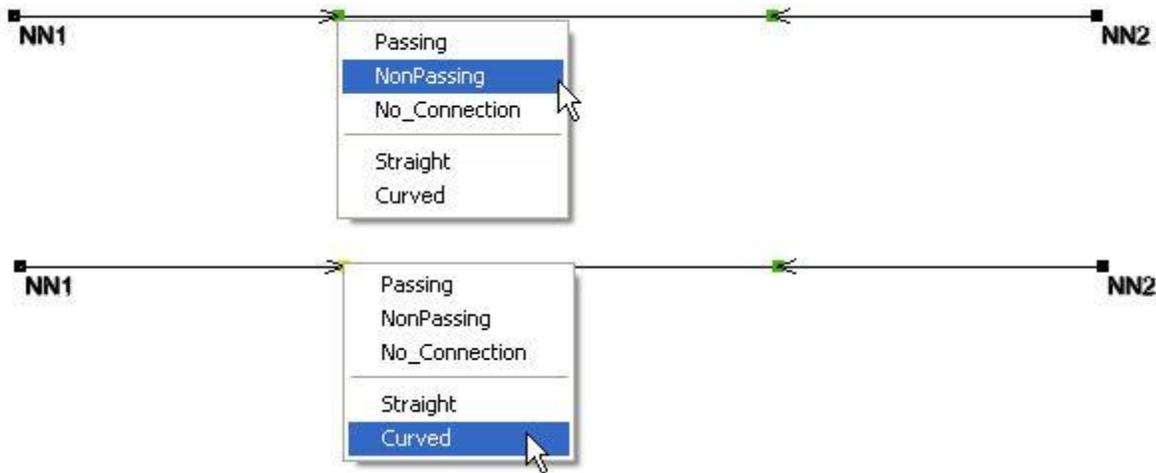
Os pontos de controle, ou Splines, são usados no FlexSim para “moldar” os caminhos dos objetos. O FlexSim utiliza a tecnologia dos Splines para mantê-lo familiarizado com o método convencional de criar curvas, inclinações e declinações aos caminhos definidos pelos Network Nodes.

Quando dois NetworkNodes são conectados através da tecla “A”, uma linha preta aparecerá na tela indicando a conexão (Figura 3-2). Também aparecerá nessas linhas pretas duas setas em sentidos opostos localizadas aproximadamente a 1/3 e 2/3 no comprimento total da linha.



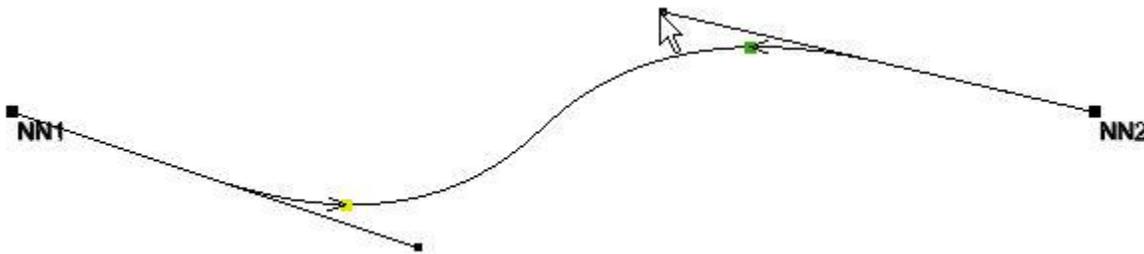
Estas setas com uma caixa verde em suas pontas indicam alguns atributos do caminho em determinada direção. As caixas verdes significam que os objetos podem se deslocar livremente naquele sentido, caixas amarelas significam que os objetos podem se deslocar naquele sentido, mas não é permitido ultrapassagens, e por fim, caixas vermelhas significam que os objetos não podem trafegar naquele sentido, tornando assim permitida a passagem em um único sentido naquele trecho. Para alternar entre essas três opções, basta clicar com o botão direito em cima das setas e selecionar a opção desejada (Figure 3-3). Para fazer uma curva no caminho, basta selecionar a opção “Curved” no menu que aparece quando se clica com o botão direito em cima da seta. Isso criará dois Splines que lhe permitirão definir as curvas ao longo

do caminho. É possível também alterar o padrão de criação dos caminhos através da ferramenta Travel Networks Utility.

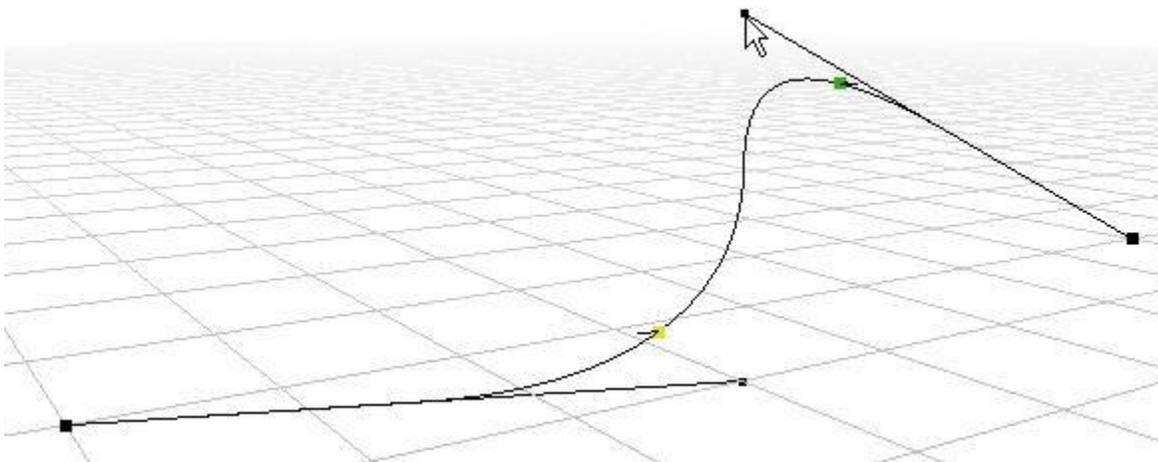


Parâmetros dos Splines

Uma vez criadas as curvas no caminho, mova as pequenas palhetas entre os nós com o mouse.



Para alterar a posição no eixo Z, selecione a palheta e role o scroll do mouse para cima ou para baixo



NetworkNodes podem ser configurados para especificar a direção do percurso a ser percorrido. Novamente, além de alterar a regra de circulação, também é possível alterar esta regra segurando a tecla "X" e clicando nas setas.



Quando um caminho é configurado utilizando os Splines, os objetos que fazem parte daquele conjunto de nós automaticamente se deslocarão no caminho definido. Tanto os caminhos definidos como as setas e caixas coloridas podem ser “ocultadas” do modelo através de cliques em cima de algum Networknode do modelo enquanto a tecla “X” é pressionada. Múltiplas visualizações são alternadas enquanto se clica no objeto segurando a tecla "X" pressionada.



[Clique aqui para um tutorial passo-a-passo.](#)

Construção do Modelo passo a passo

Construindo o Modelo 3

Para iniciar o Modelo 3, é necessário ter completado o Modelo 2 do exercício anterior.

Caso seja encontrado alguma dificuldade em qualquer momento da modelagem, existe um tutorial completo no endereço <http://www.flexsim.com/tutorials>

Etapa 1: Abra o modelo do exercício 2

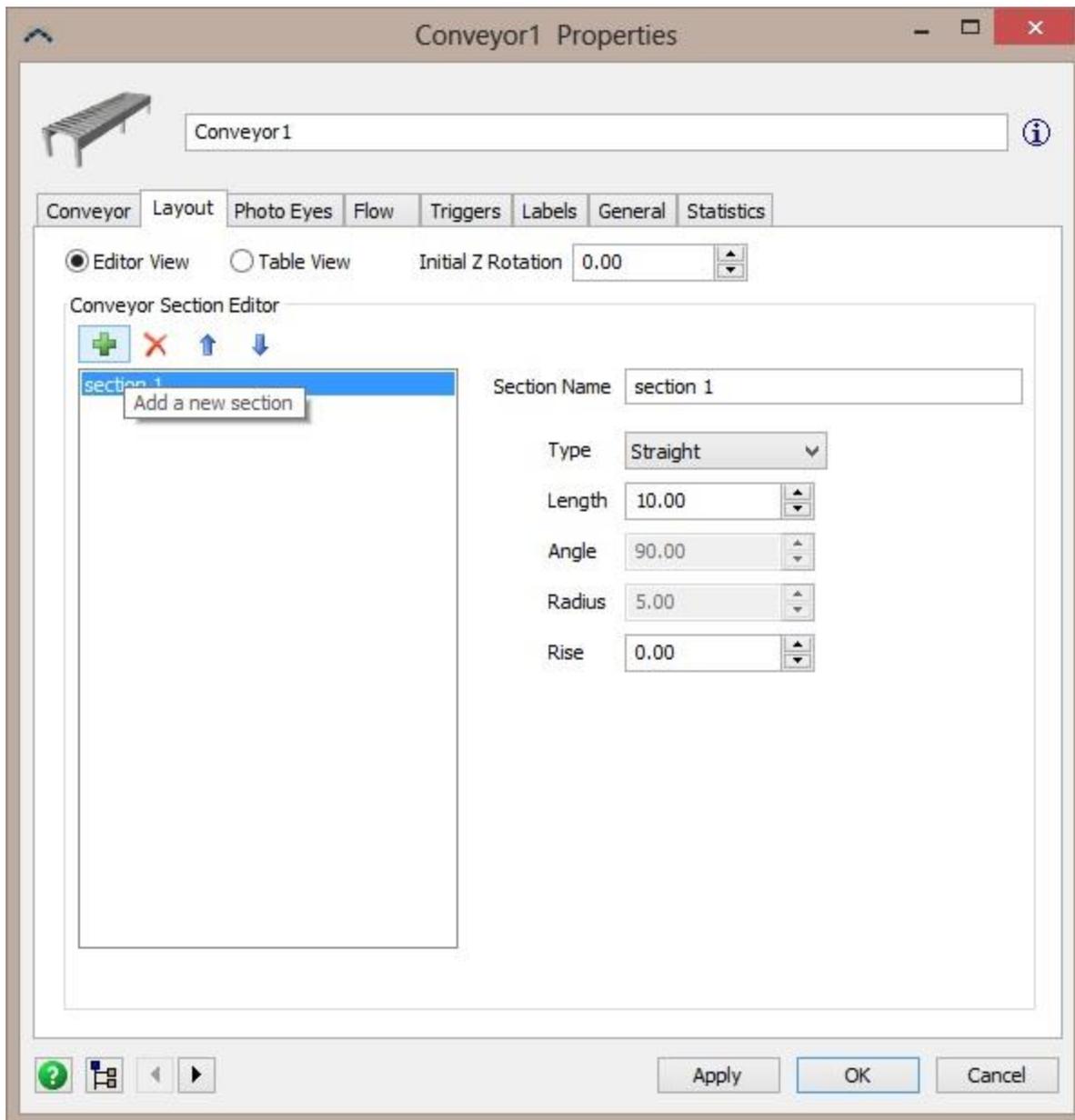
- Carregue o modelo 2 se você ainda não o abriu.

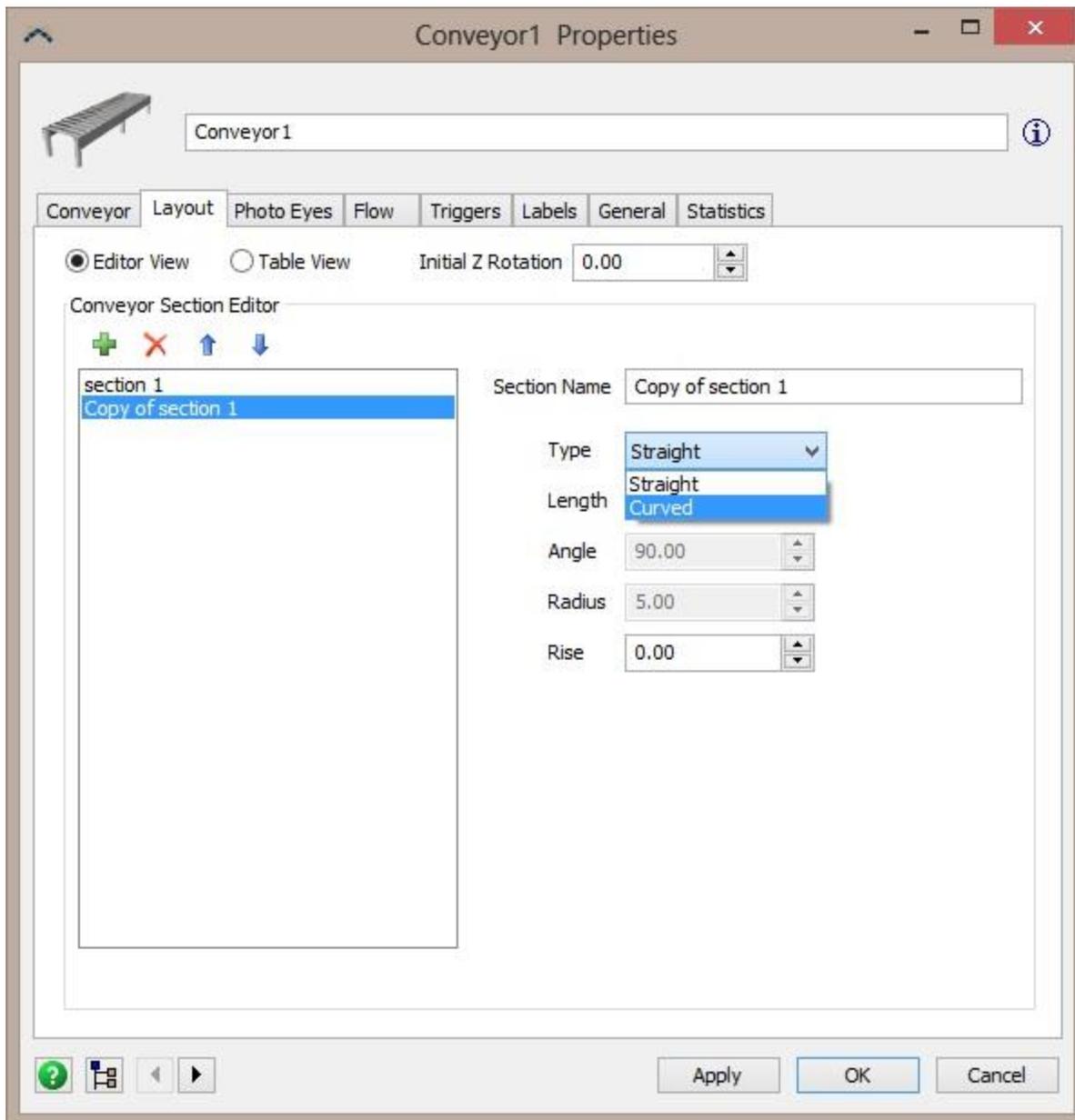
Etapa 2: Reconfigure o layout do Conveyor1 e Conveyor3

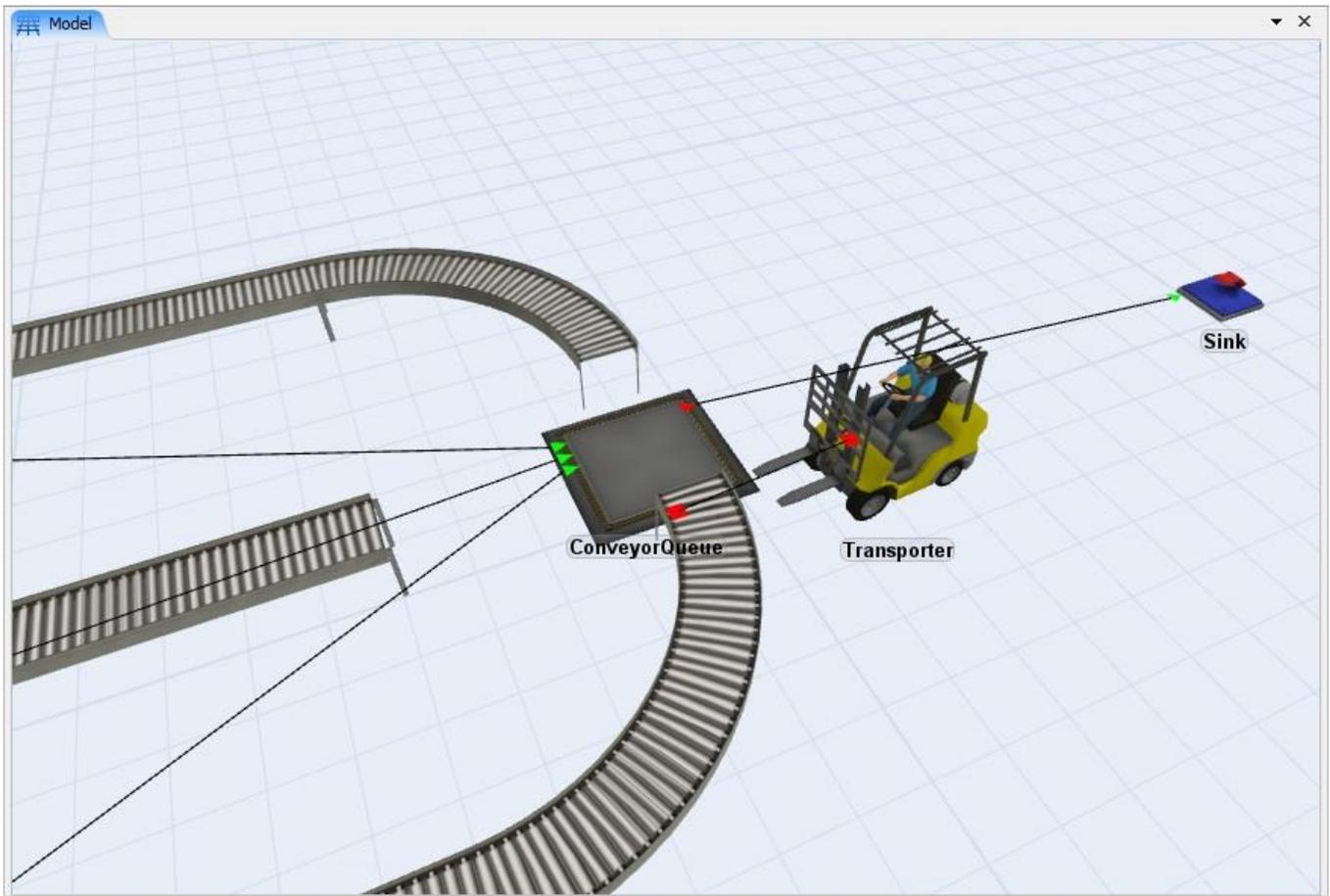
Agora iremos alterar o layout dos três conveyors de modo que o final deles fique posicionado próximo ao **ConveyorQueue**. Caso queira realizar testes para criar layouts complexos nos conveyors através da guia Layout do conveyor, fique à vontade!

- Clique duplo no *Conveyor1* para abrir sua janela **Properties**.
- Clique na guia **Layout**. Para mais informações sobre esta página, clique aqui.
- Clique em **Add** para adicionar mais seções ao fim do conveyor.
- Selecione **Curved** na lista **Type**.
- Edite o ângulo e o raio deste conveyor de modo que o final do **Conveyor1** fique posicionado próximo ao **ConveyorQueue** conforme mostrado Guiaixo.
- Repita o mesmo processo acima para o *Conveyor3*.

Nota: A seção do conveyor que você selecionar na lista da guia Layout será destacada com uma cor mais escura no modelo 3D, deste modo fica mais fácil visualizar as alterações que estão sendo feitas.







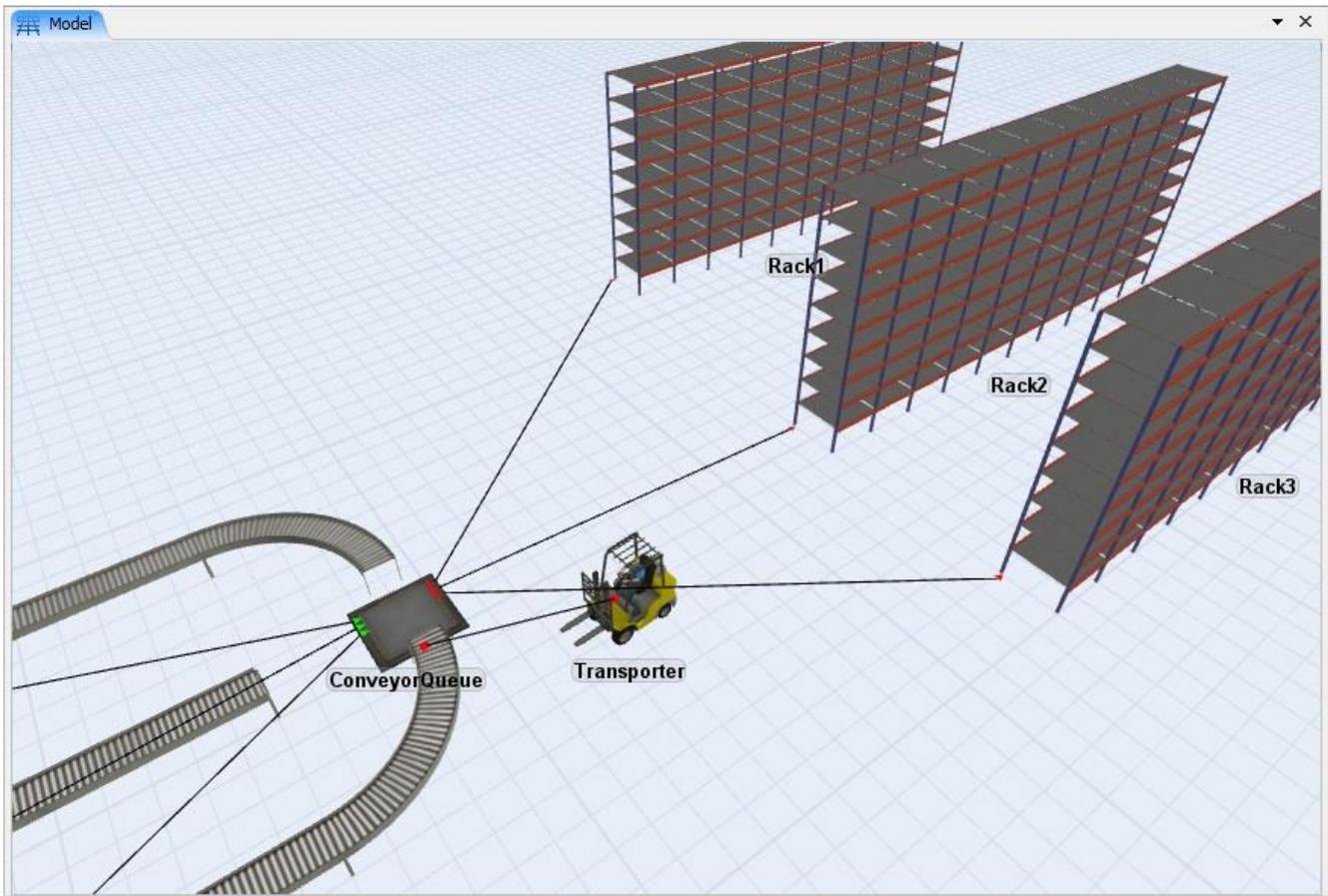
Etapa 3: Delete o Sink

- Destaque o *Sink* e pressione a tecla **Delete** no teclado.

Quando um objeto é deletado, todas as suas conexões são desfeitas. Fique atento, pois isso poderá alterar o número das portas do objeto que estava conectado ao objeto deletado.

Etapa 4: Crie 03 Racks

- Crie três Racks, os nomeie como *Rack1*, *Rack2* e *Rack3* e os coloque à direita do *ConveyorQueue*.
- Coloque os racks longe o suficiente do Queue para que exista uma distância a ser percorrida pela empilhadeira.
- Conecte o *ConveyorQueue* ao *Rack1*, *Rack2* e ao *Rack3* (tecla A).

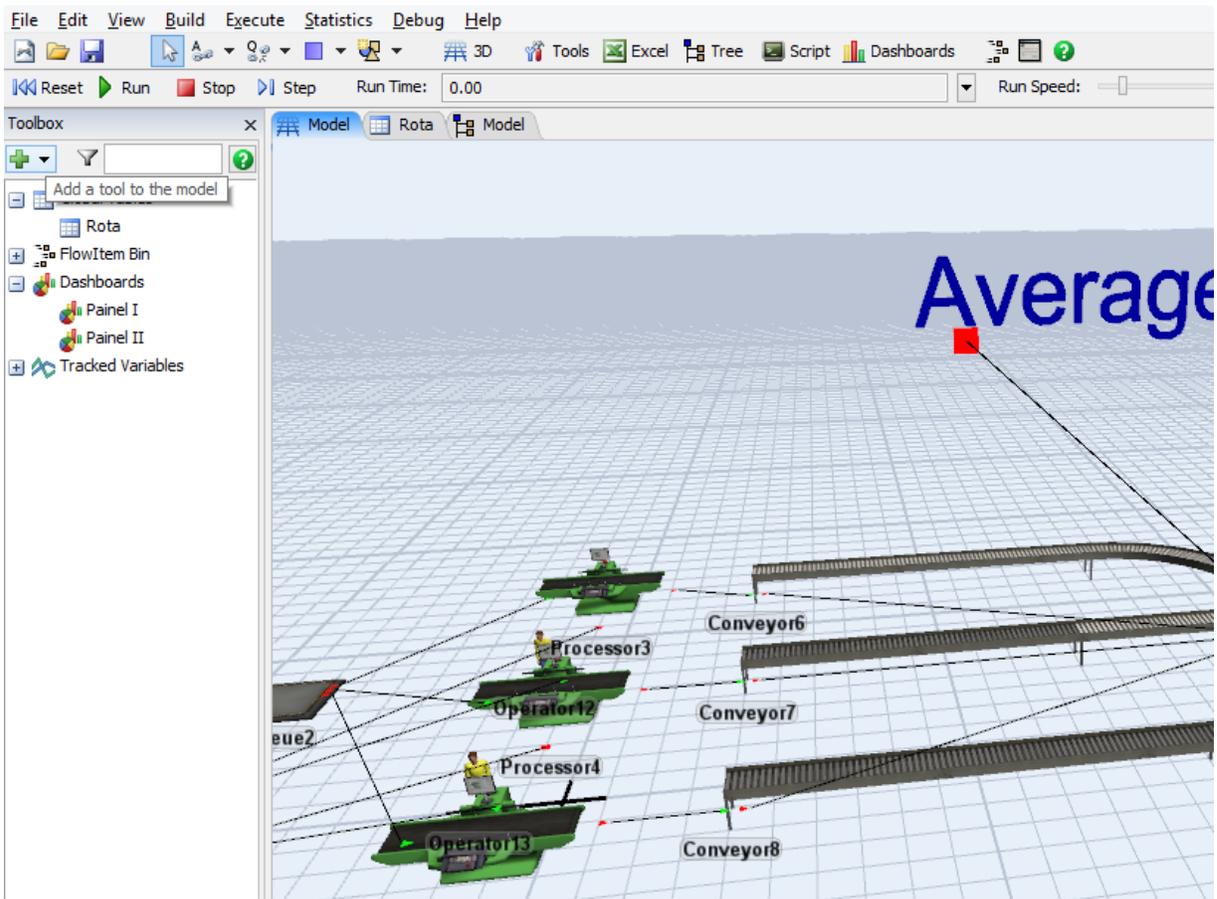


Etapa 5: Crie uma Global Table para controlar as rotas dos Itens

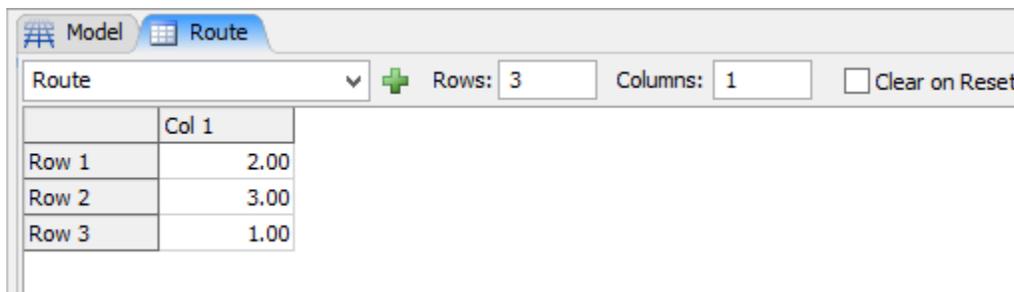
A próxima etapa é a de criar uma Global Table que será usada como base para o *ConveyorQueue* definir para onde enviará cada tipo de item (ou mais precisamente, qual será a porta de saída usada para cada tipo de item). De acordo com a ordem de conexão, a porta de saída número 1 deverá estar no *Rack1*, na porta de saída número 2 deverá estar o *Rack2* e na porta 3 o *Rack3*. Caso as conexões não estejam nesta ordem, você pode corrigi-las através do campo "Ports" na guia "General" nas propriedades do *ConveyorQueue*.

Enviaremos todos os itens que possuam o itemtype igual a 1 para o *Rack2*, todos itemtypes igual a 2 para o *Rack3*, e os itemtypes igual a 3 para o *Rack1*. Aqui se encontram as etapas para adicionar e configurar a Global Table:

- Clique em **Tools** e depois em "Add a tool to the model" localizado na seta em preto próximo ao .



- Alterar o campo **Name** da Global Table **Route**.
- Defina o campo **Rows** para 3 e **Columns** para 1.
- Os nomes das linhas são **Item1**, **Item2** e **Item3**, então preencha as linhas de acordo com as portas de saída que o itens deverão ir, ou seja, em quais Racks os itens serão armazenados.
- Clique no botão **Close** para aplicar as configurações e fechar a tabela.



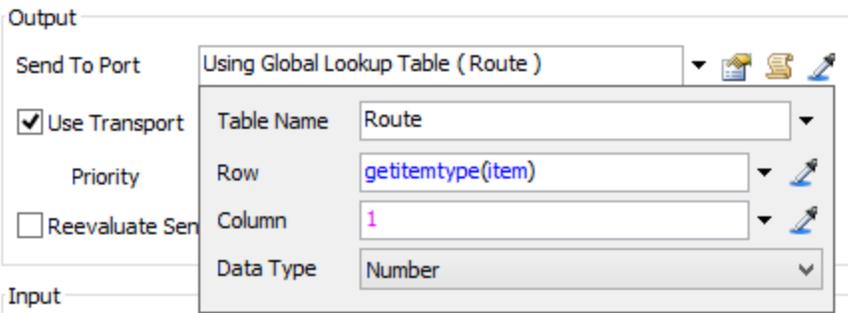
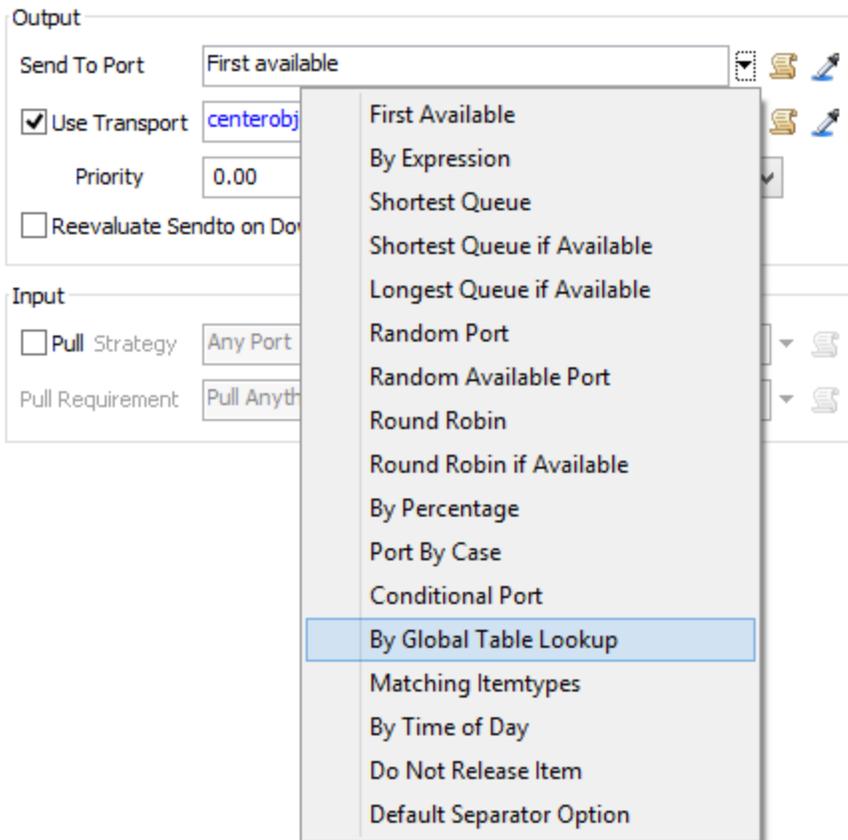
Agora com a Global Table definida, basta ajustar o campo "Send To Port" do Queue.

Etapa 6: Ajuste da Opção Send To Port para o ConveyorQueue

Você pode querer definir o fluxo do ConveyorQueue e opções de transporte ao clicar sobre o Queue no formato de visualização em 3D para mostrar suas propriedades na janela Quick Properties.

Alternativamente, você pode abrir a janela de propriedades dos objetos.

- Clique duplo no *ConveyorQueue* para abrir a sua janela **Properties**.
- Clique na guia **Flow**. Selecione a opção **By Global Table Lookup** na lista do campo **Send To Port**. A template de códigos irá aparecer. Edite os textos das caixas conforme imagens guia Guiaixo:



- Clique no botão **OK** para fechar sua janela de propriedades.

Etapa 7: Resete, Salve e Rode (Reset, Save, and Run)

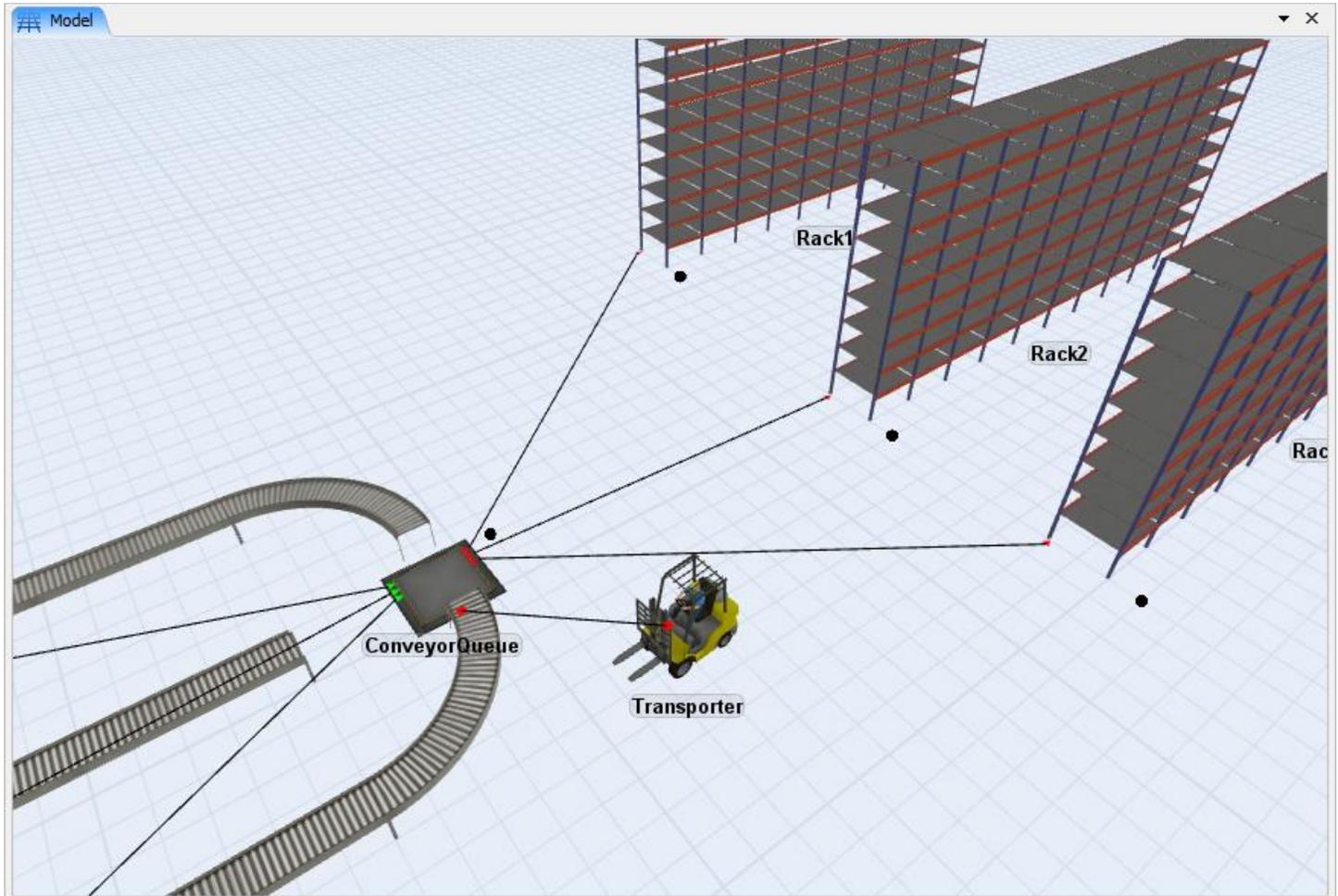
Neste ponto o modelo já está pronto para ser simulado, então **Resete**, **Salve** o modelo, e então **Rode** a simulação para verificar se tudo está funcionando como deveria. A empilhadeira deverá transportar os itens para os racks de acordo com o tipo de item, conforme definido na Global Table.

Etapa 8: Adição de NetworkNodes para criação de um caminho para a empilhadeira

NetworkNode's são usados para criar um caminho por onde qualquer task executer (executador de tarefas) podem passar, ou seja, qualquer Empilhadeira, Operador, ASRSvehicle, Crane, entre outros. No exercício anterior foi utilizado um operador e uma empilhadeira para transportar os itens ao longo do modelo. Até aqui, todos objetos se movimentam livremente ao longo do modelo, podendo assim ir em linha reta ao seu destino, independente do que existe neste meio. Agora gostaríamos de

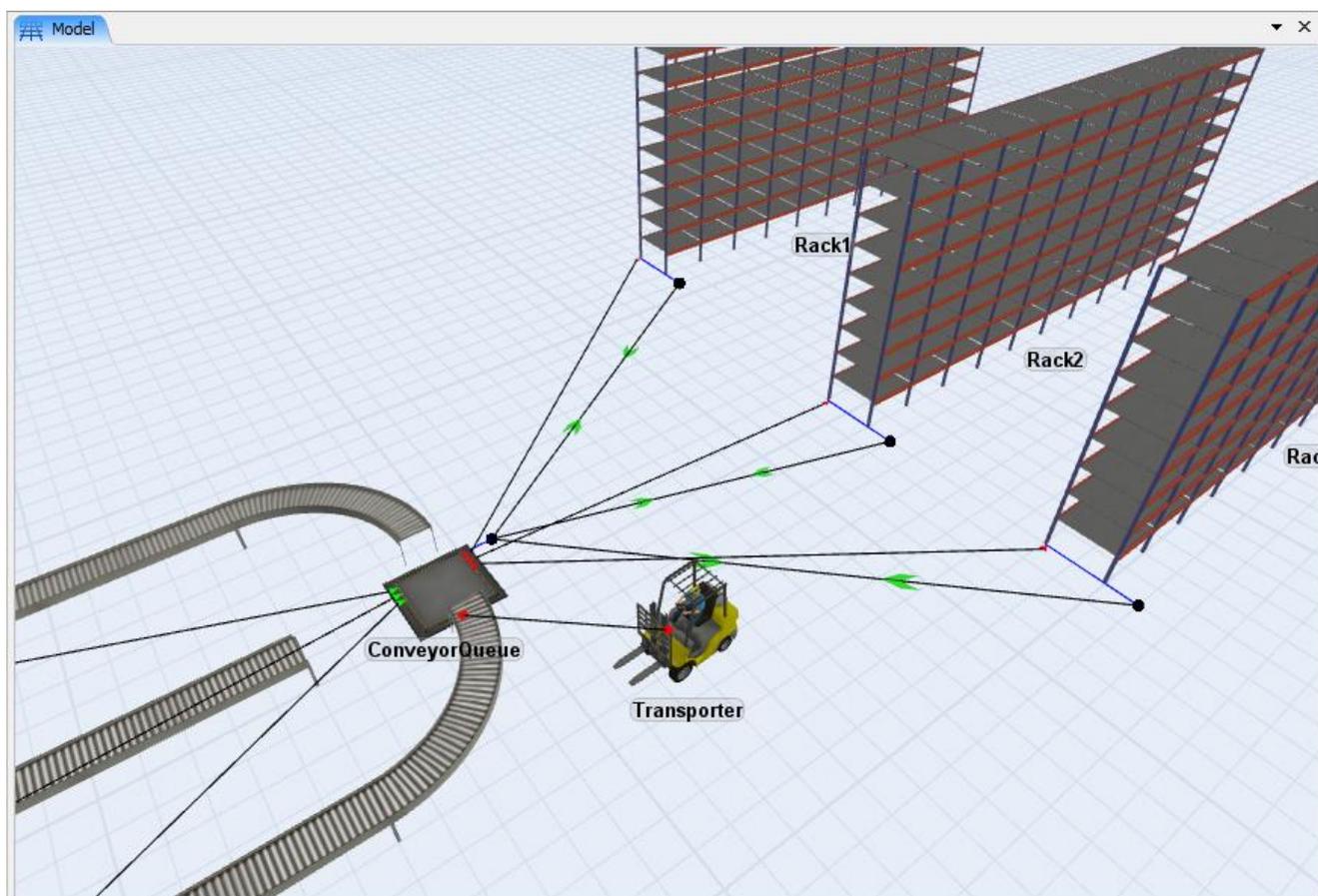
confinar a empilhadeira a um caminho específico, sendo assim, está empilhadeira só poderá chegar ao seu destino se este caminho a der acesso ao objeto final. O próximo passo então é criar este caminho para o queue e cada um dos racks. Nomeie os nós como *NN1*, *NN2*, etc.

- Crie um **NetworkNodes** arrastando-o das bibliotecas para dentro do modelo. Posicione-o perto do conveyorqueue e perto dos Racks, nomeie-os como NN1, NN2, NN3 e NN4. Estes nodes serão os pontos de carga/descarga dos objetos. Você pode também adicionar outros nodes entre estes, mas isso não é necessário.

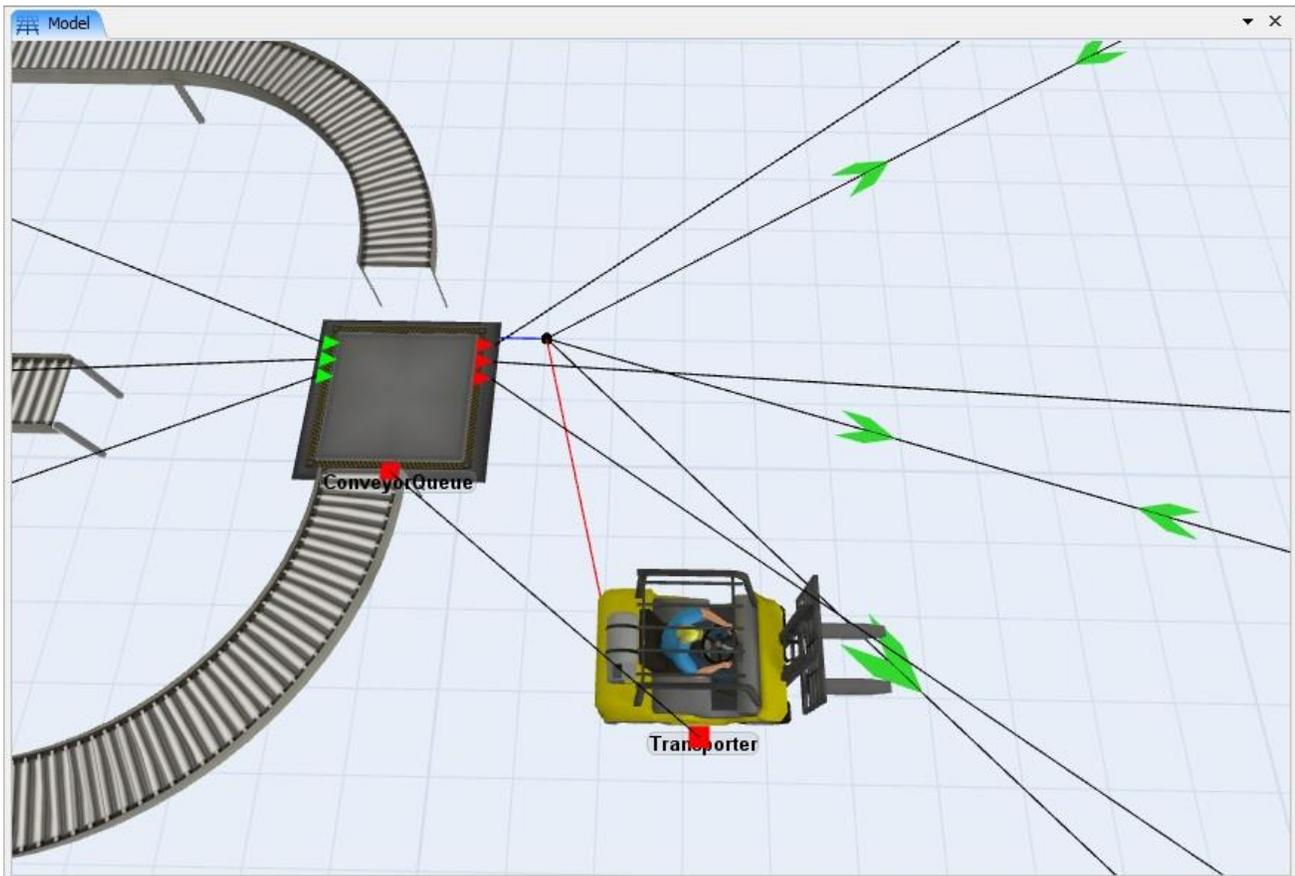


- Conecte o *NN1* ao *NN2*, *NN3* e *NN4* (tecla A). Uma linha aparecerá depois das conexões serem feitas com duas caixas verdes ao longo delas, indicando que os objetos podem se deslocar em ambos sentidos.

- Conecte cada NetworkNode ao objeto correspondente (*NN1* ao *ConveyorQueue*, *NN2* ao *Rack1*, etc.), todos com a tecla A. Uma fina linha azul aparecerá se a conexão for feita corretamente. Caso não consiga ver a linha azul, tente mover o NetworkNode.



- A última etapa é conectar a empilhadeira ao conjunto de nodes criados. Para a empilhadeira saber qual é o seu caminho, é necessário apenas conecta-la a um dos nodes da rede. Conecte o *Transporter* ao *NN1* (tecla A). Agora este node se tornou o ponto inicial da empilhadeira quando o modelo for resetado.

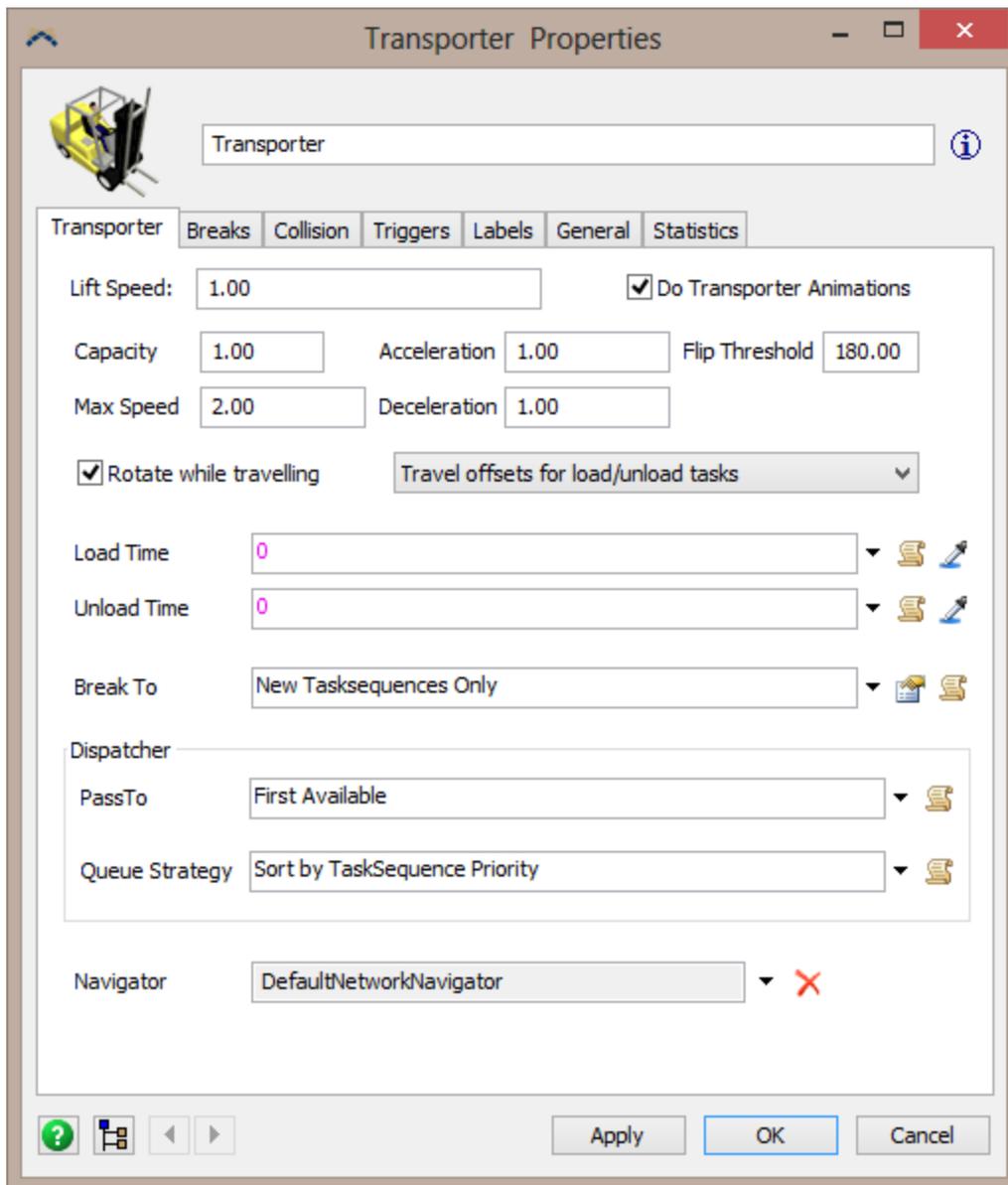


Etapa 9: Resete, Salve e Execute (Reset, save, and run the model)

Para checar novamente se todas as conexões foram feitas corretamente e se a empilhadeira está se deslocando apenas sobre o caminho definido, repita o mesmo procedimento de antes, **Resete**, **Salve** e então **Rode** a simulação.

Uma palavrinha sobre os offsets

Conforme o a simulação roda, você perceberá que, em alguns momentos, a empilhadeira sai do conjunto de nós para deixar ou carregar algum item no seu destino, quando isso acontece, está empilhadeira permanece no estado "Travel offsets for loa/enloda tasques". Isso acontece porque existe uma propriedade nos task executers que define o que ele irá fazer quando chegar ao objeto de destino para carregar/descarregar o item, por exemplo, se ele vai até o local do item, se vai ficar bloqueado apenas até o node de acesso daquele objeto ou se vai chegar apenas no objeto que guarda o item e já carregar/descarregar o item (sem precisar ir até o local específico onde o item está/ficará localizado).

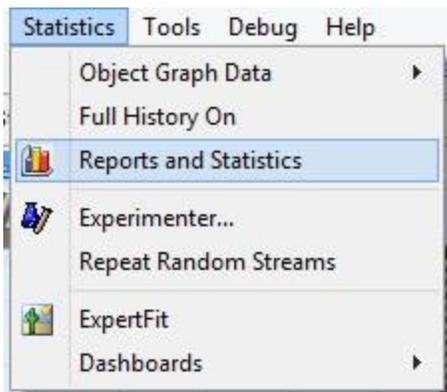


Offsets são usados pela empilhadeira para saber onde o item está localizado dentro do objeto que o comporta para ser carregado/descarregado. Este campo define, por exemplo, se a empilhadeira entrará dentro de um Queue para carregar o item ou não, ou se a empilhadeira irá até a célula onde o item se encontra no rack para carrega-lo. Para forçar a empilhadeira a não sair do conjunto de NetworkNodes para carregar/descarregar o item, selecione a opção "Do not travel offsets for loa/enloda tasques", localizada no campo guia abaixo da caixa "Deceleration"

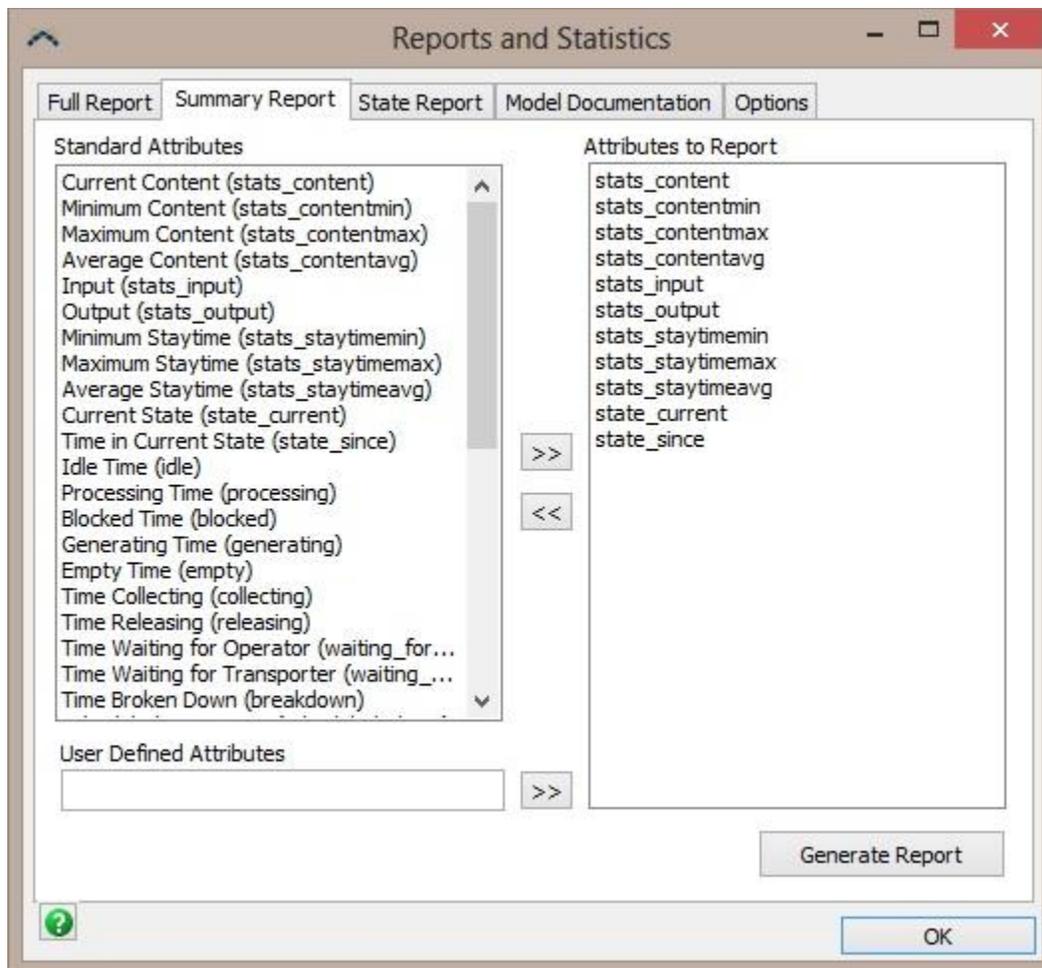
Por padrão, qualquer caminho escolhido pelo operador para atingir o seu destino é calculado pelo algoritmo Dijkstra, que escolhe a menor distância para se deslocar de um ponto a outro.

Etapa 10: Utilização de relatório para verificar as estatísticas de output

Para visualizar os resultados da simulação após um período específico de tempo, selecione a opção **Statistics > Reports and Statistics**.



Vá até a guia **Summary Report** da janela do Reports and Statistics.



Para gerar um relatório básico, pressione o botão **Generate Report**. Caso tenha algum outro atributo a ser inserido no relatório, é possível adicioná-lo utilizando a interface mostrada. Este relatório será exportado no formato csv e automaticamente mostrado no Excel ou outro programa padrão do computador que esteja configurado para abrir arquivos csv.

summaryreport.csv - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Add-Ins

Clipboard Font Alignment Number Styles Cells Editing

A1 Flexsim Summary Report

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Flexsim Summary Report												
2	Time:	37376.36											
3													
4	Object	Class	stats_conf	stats_conf	stats_conf	stats_conf	stats_inpu	stats_outp	stats_stay	stats_stay	stats_stay	state_curr	state_since
5	Source	Source	0	0	0	1	0	1339	0	326.6003	7.882783	5	37376.36
6	Queue	Queue	24	0	25	22.33462	1339	1315	1.777517	3185.945	618.1947	8	37376.36
7	Processor	Processor	0	0	1	0.46037	461	461	10.0347	162.5524	37.29556	1	37376.36
8	Processor	Processor	1	0	1	0.498806	466	465	10.037	183.433	40.07985	2	37376.36
9	Processor	Processor	1	0	1	0.923999	388	387	10.31463	2307.401	89.17754	2	37376.36
10	Conveyor	Conveyor	0	0	3	0.405734	461	461	14.71239	68.82785	32.89554	6	37376.36
11	Conveyor	Conveyor	1	0	13	1.42399	465	464	10	723.6134	114.6433	4	37376.36
12	Conveyor	Conveyor	25	0	25	23.34033	387	362	14.71239	5121.235	2343.239	4	37376.36
13	Dispatche	Dispatche	0	0	0	0	0	0	0	0	0	2	37376.36
14	Operator1	Operator	0	0	1	0.016472	262	262	1.844796	2.949462	2.346952	1	37376.36
15	Operator2	Operator	0	0	1	0.068573	1053	1053	1.844763	2.958871	2.433187	1	37376.36
16	Conveyor	Queue	10	0	10	9.881959	1287	1277	0.729	339.848	288.1934	10	37376.36
17	Transport	Transport	1	0	1	0.516961	1277	1276	8.288835	19.38042	15.14272	15	37376.36
18	Rack1	Rack	359	1	359	157.3819	359	0	0	0	0	2	37376.36
19	Rack2	Rack	457	1	457	236.5265	457	0	0	0	0	2	37376.36
20	Rack3	Rack	460	1	460	239.7007	460	0	0	0	0	2	37376.36
21													

summaryreport

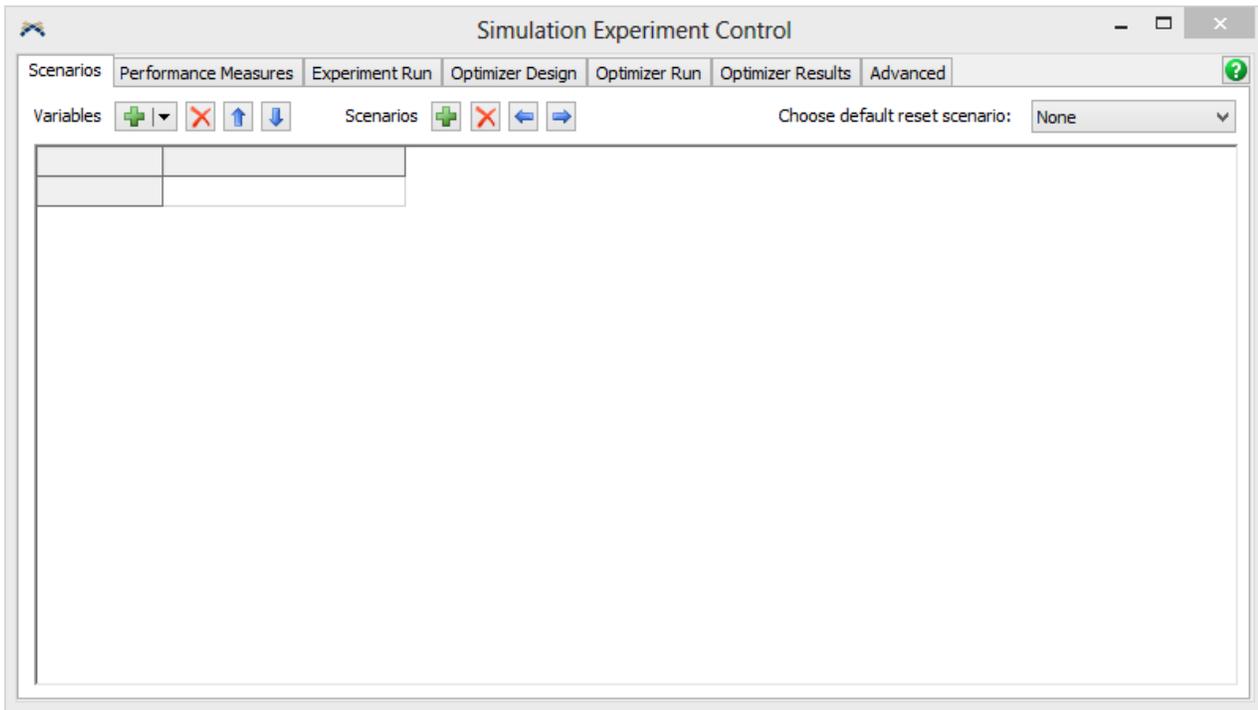
Ready 100%

Para gerar um relatório de estados, vá até a guia **State Report** da janela Reports and Statistics e pressione **Generate Report**.

Object	Class	idle	processing	busy	blocked	generatin	empty	collecting	releasing	waiting_f	waiting_f	breakdow	sc
Source	Source	0.00%	0.00%	0.00%	28.24%	71.76%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	0.22%	0.00%	99.76%	0.00%	0.03%	0.00%	0.00%
Processor	Processor	54.00%	32.60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.07%	0.00%	0.00%	0.00%
Processor	Processor	50.10%	37.10%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.33%	0.00%	0.00%	0.00%
Processor	Processor	7.59%	30.23%	0.00%	50.74%	0.00%	0.00%	0.00%	0.00%	1.05%	0.00%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	21.45%	0.00%	63.70%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	50.57%	0.00%	42.68%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	96.68%	0.00%	2.09%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Dispatch	Dispatch	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Operator1	Operator	69.46%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Operator2	Operator	76.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	0.49%	0.00%	0.00%	0.00%	99.51%	0.00%	0.00%
Transport	Transport	0.38%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Rack1	Rack												
Rack2	Rack												
Rack3	Rack												

Etapa 11: Rodando múltiplas simulações do modelo com o Experimenter

Para acessar o Experimenter no FlexSim, selecione no menu a opção **Statistics>Experimenter**. A janela de controle do Experimenter aparecerá na tela.



A janela de controle do Experimenter é usada para rodar múltiplas replicações de um determinado modelo, e rodar diferentes cenários do mesmo modelo. Quando se simula múltiplos cenários, é possível declarar um número de variáveis e atribuir os valores que aquelas variáveis assumirão em cada cenário. O intervalo de confiança é calculado automaticamente para cada cenário e indicador de performance (KPI) definido em Performance Measure. Para mais informações, consulte a seção Experimenter do material de apoio do FlexSim.

Isto completa o exercício 3. Parabéns!

Tutorial sobre Labels

1. Introdução
2. Construção do Modelo passo a passo

Introdução

Este tutorial irá introduzir você no uso de Labels em seu modelo. Labels podem ser usados para armazenar informações em seus objetos e items que podem ser acessados a qualquer tempo. Conforme alguns exemplos, label podem ser úteis para especificar o fluxo no processo de seus items, ou podem ser usados para armazenar dados financeiros sobre um objeto.

O que você irá aprender

- Como criar e acessar labels
- Como você pode usar labels para alterar a forma como seu modelo funciona.

Novas Funcionalidades

- Você irá aprender um pouco sobre Pull Requirements
- Você irá aprender sobre o FlowItem Bin

Tempo aproximado para completar este exercício

Este exercício levará de 20-30 minutos para ser completado

Overview do Modelo

Neste modelo, nós iremos criar um modelo simples que cria entidades (items) com três diferentes tipos de itens (itemtypes). Nós iremos rastrear quantos items serão criados de cada tipo e ao ser processados, nós iremos modificar o tamanho de nossos items.

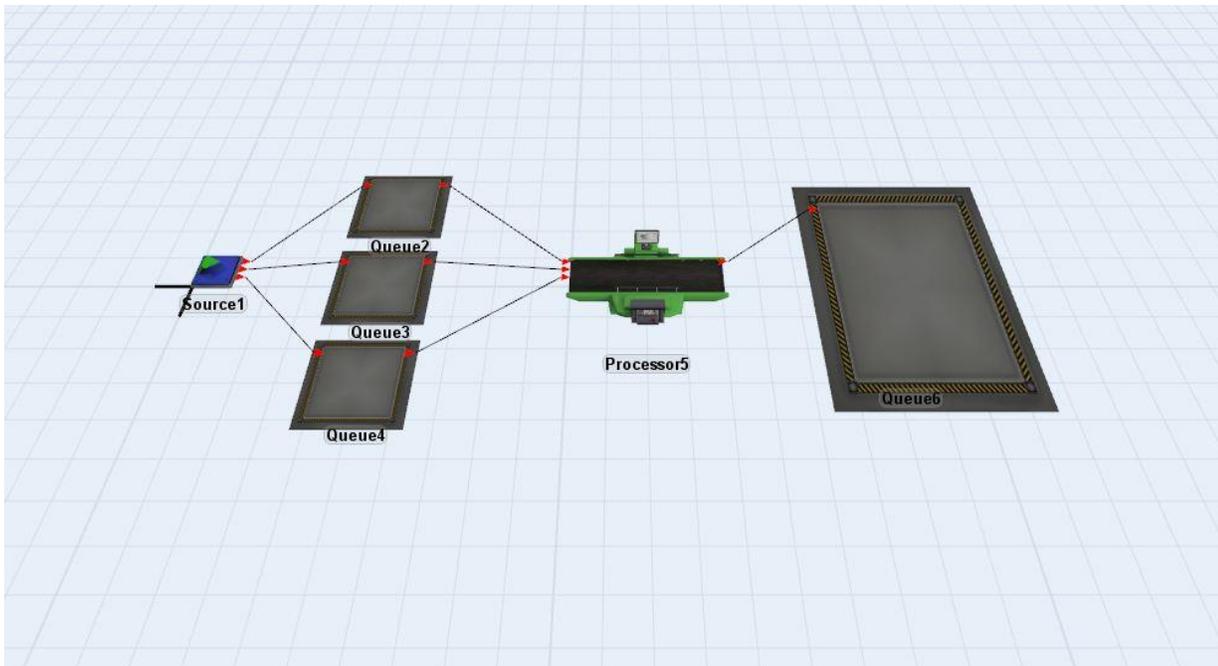
Construção do Modelo passo a passo

Construindo o modelo de labels

Comece um novo modelo ao clicar no botão  na barra de ferramentas. Clique em OK na janela de unidade do modelo (Model Units window) – nós iremos usar as unidades padrões do nosso modelo.

Etapa 1: Crie os objetos

- Arraste os objetos da biblioteca (Library Icon Grid) para dentro da superfície de simulação 3D para criar o modelo mostrado abaixo.



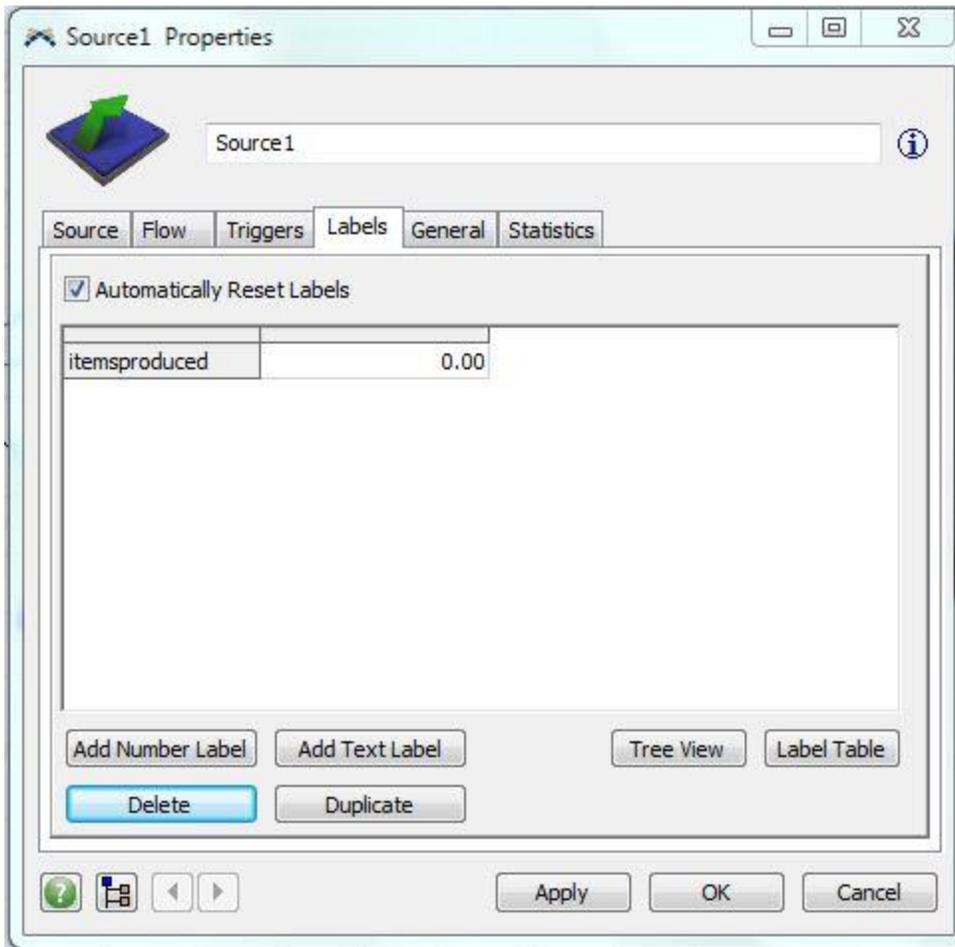
Conecte todos os objetos conforme mostrado:

- Conecte o *Source1* ao *Queue2*, *Queue3* e *Queue4*
- Conecte o *Queue2*, *Queue3* e *Queue4* ao *Processor5*
- Conecte o *Processor5* ao *Queue6*

Etapa 2: Configure as propriedades dos Sources

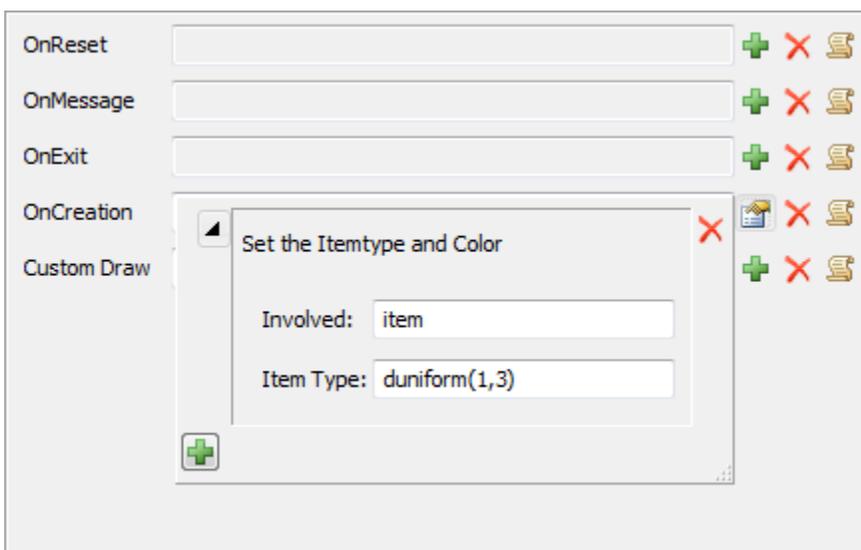
Todos os objetos podem armazenar suas próprias Labels. A guia Labels permite você criar, visualizar e manipular as labels do objeto.

- Duplo clique sobre o *Source* para abrir sua janela de propriedades
- Vá para a guia **Labels**
- Marque **Automatically Reset Labels**. Isto fará com que as labels sejam resetadas para o valor que você havia atribuído a ela, quando você resetar o modelo.
- Clique em **Add Number Label** e duplo clique em "*newlabelname*" para renomear a label para "*itemsproduced*".



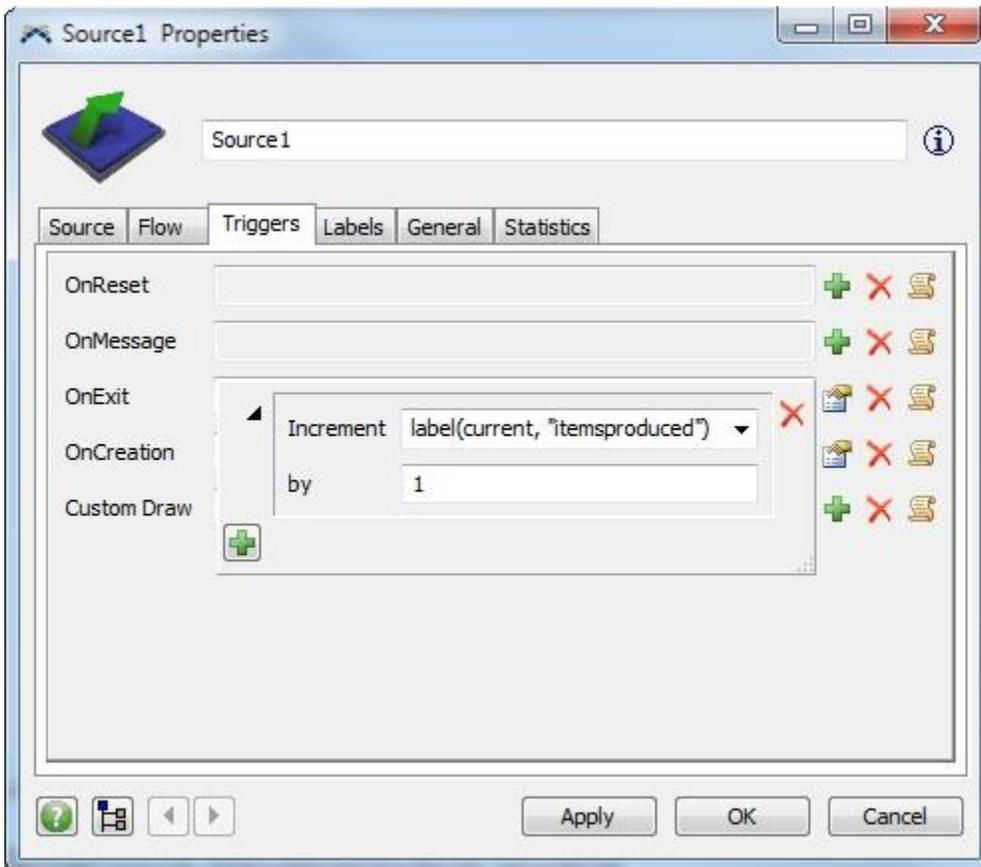
O *Source* irá criar 3 tipos de itens diferentes (itemtypes). A cada itemtype será atribuída uma cor específica.

- Vá para a guia **Triggers**.
- Clique no botão **+** na trigger **OnCreation**, e selecione a opção *Set Item Type and Color*
- Nós usaremos 3 itemtypes (tipos de itens), então mantenha **Involved** com **item**, ou seja, como está e em **Item Type** mantenha o padrão.



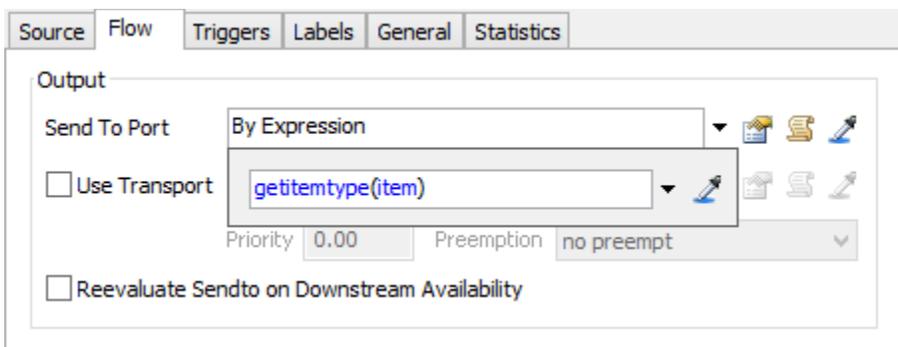
Nós também queremos incrementar a label "*itemsproduced*" para manter rastreando quantos itens tem sido produzidos pelo source.

- Vá para a guia **Triggers**.
- Clique no botão **+** na trigger **OnExit** trigger, e selecione a opção de pick list **Increment Value**
- Na caixa editável para **Increment** substitua a `label(item, "labelname")` com a `label(current, "itemsproduced")`.
- Deixe **By** com **1**. Isto irá incrementar o valor da label `"itemsproduced"` em 1 cada vez que um item deixar o Source.



Nós iremos atribuir itens de tipos diferentes para ir aos buffers específicos (queue) baseado nos tipos de itens (itemtype).

- Vá para a guia **Flow** do source.
- Escolha **By Expression** na opção **Send to Port**. Deixe conforme padrão `getitemtype(item)`.



- Clique em **Apply** para aplicar as mudanças que você fez no Source.

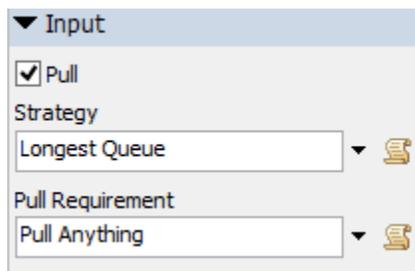
Resete e execute o seu modelo. Se você retornar para a guia **Label** do source e olhar a label "itemproduced", ela atualizará o valor conforme um novo flowitem é criado.

Feche a janela de propriedades do Source. Ele poderá perguntar se você quer "Update Label Reset Values?". Clique em Não. (Você quer que as labels resetem para zero quando você resetar o modelo).

Etapa 3: Configure as propriedades do equipamento (Processor)

Queremos que o equipamento (processor) pegue os itens da fila (**Queue**) que tem o maior número de itens em espera, ou a fila mais longa. Uma vez que cada Queue não sabe quantos itens estão nas outras filas (**Queues**), é melhor configurar para que o Processor determine de qual fila (**Queue**) o item será puxado.

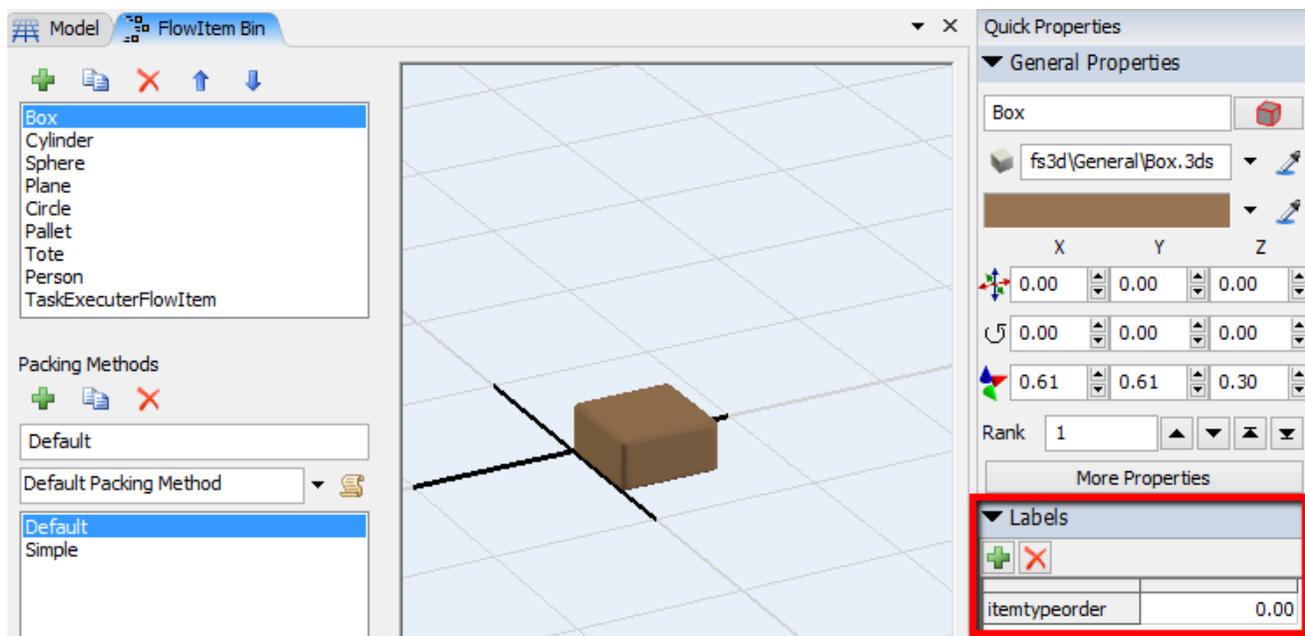
- Clique sobre o Processor para abrir suas propriedades na janela Quick Properties.
- Clique sobre a flecha próximo a seção **Input** para expandir.
- Marque a opção **Pull** e configure a estratégia **Pull Strategy** como Longest Queue.



Etapa 4: Customize os Flowitems

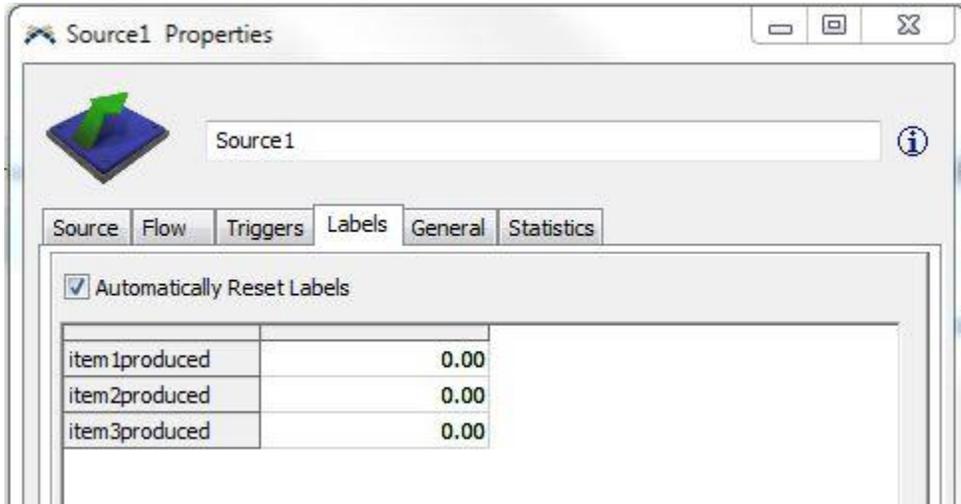
Nós vamos armazenar informações sobre todos os flowitems para que eles possam ser processados através do modelo.

- Abra o **Flowitem Bin** ao ir em **Tools > Flowitem Bin**
- Com o **Box** selecionado clique em **+** e na seção de Labels (em vermelho abaixo) na janela de Quick Properties adicione uma nova label (number).
- Duplo clique em "**LabelName**" para renomear a label como "**itemtypeorder**".



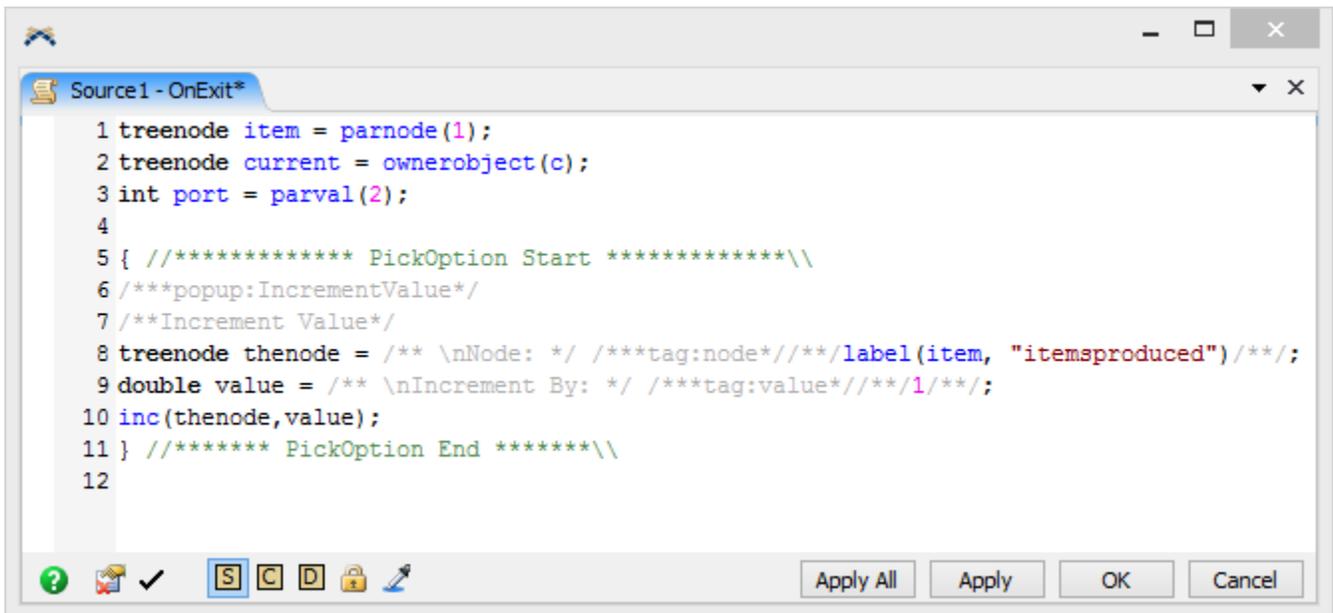
Nós agora teremos o source incrementando a label "itemtypeorder" de cada flowitem para cada tipo específico de item. Isto irá permitir nós conhecermos em qual ordem o flowitem (entidade) foi criado.

- Abra a janela de propriedades para o Source.
- Vá para a guia **Labels**.
- Altere a label "itemsproduced" para "item1produced".
- Selecione a label "item1produced" e clique no botão **Duplicate** duas vezes.
- Renomeie as duas novas labels como "item2produced" e "item3produced".



Para incrementar cada uma destas labels corretamente, nós precisamos customizar a trigger OnExit. Isto irá requerer a edição do código manualmente, mas não se preocupe, escrever estes código é fácil!

- Vá para a guia Triggers, e clique sobre o botão  , localizado na parte direita da Trigger OnExit, para abrir o editor de código (**Code Editor**).



- Clique sobre o ícone  para remover o template para adicionar a linha de comentários informando que é um código customizado.
- Sinta-se à vontade para deletar as linhas de "PickOption Start/End".

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 { /******* PickOption Start *****\
7
8
9 treenode thenode = label(item, "itemsproduced");
10 double value = 1;
11 inc(thenode,value);
12 } /******* PickOption End *****\
13

```

- Nós queremos a função “inc” para incrementar cada label individualmente. Escreva uma declaração “e se” que referencia cada tipo de item (itemtype) para sua própria label. Nós também queremos configurar os novos valores das labels para o flowitem.

```

if (getitemtype(item) == 1)
{
    int value = inc(label(current, "item1produced"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

```

- Escreva a mesma declaração para o itemtype 2 e 3. O código deve se parecer com alguma coisa como abaixo:

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 if (getitemtype(item) == 1)
7 {
8     int value = inc(label(current, "item1produced"), 1);
9     setlabelnum(item, "itemtypeorder", value);
10 }
11 if (getitemtype(item) == 2)
12 {
13     int value = inc(label(current, "item2produced"), 1);
14     setlabelnum(item, "itemtypeorder", value);
15 }
16 if (getitemtype(item) == 3)
17 {
18     int value = inc(label(current, "item3produced"), 1);
19     setlabelnum(item, "itemtypeorder", value);
20 }

```

Basicamente o que está dizendo acima é, se o tipo de item for o 1 que foi produzido no Source naquele instante, então incremente 1 para a quantidade produzida (ou seja, conte 1). Estamos incrementando para na verdade possibilitar a criação de um novo item e também contar a quantidade de itens do tipo 1, 2 e 3 que estão sendo produzidos. Além disso ao produzir o tipo de item 1 naquele instante, registre a ordem em que o tipo de item foi criado (por tipo de item).

- Clique em **OK** para aplicar e feche a janela de código.
- Clique em **OK** para aplicar e fechar a janela de propriedades.

Etapa 5: Processe os Flowitems

Nesta etapa, nós iremos alterar como os equipamentos (Processors) processam as caixas.

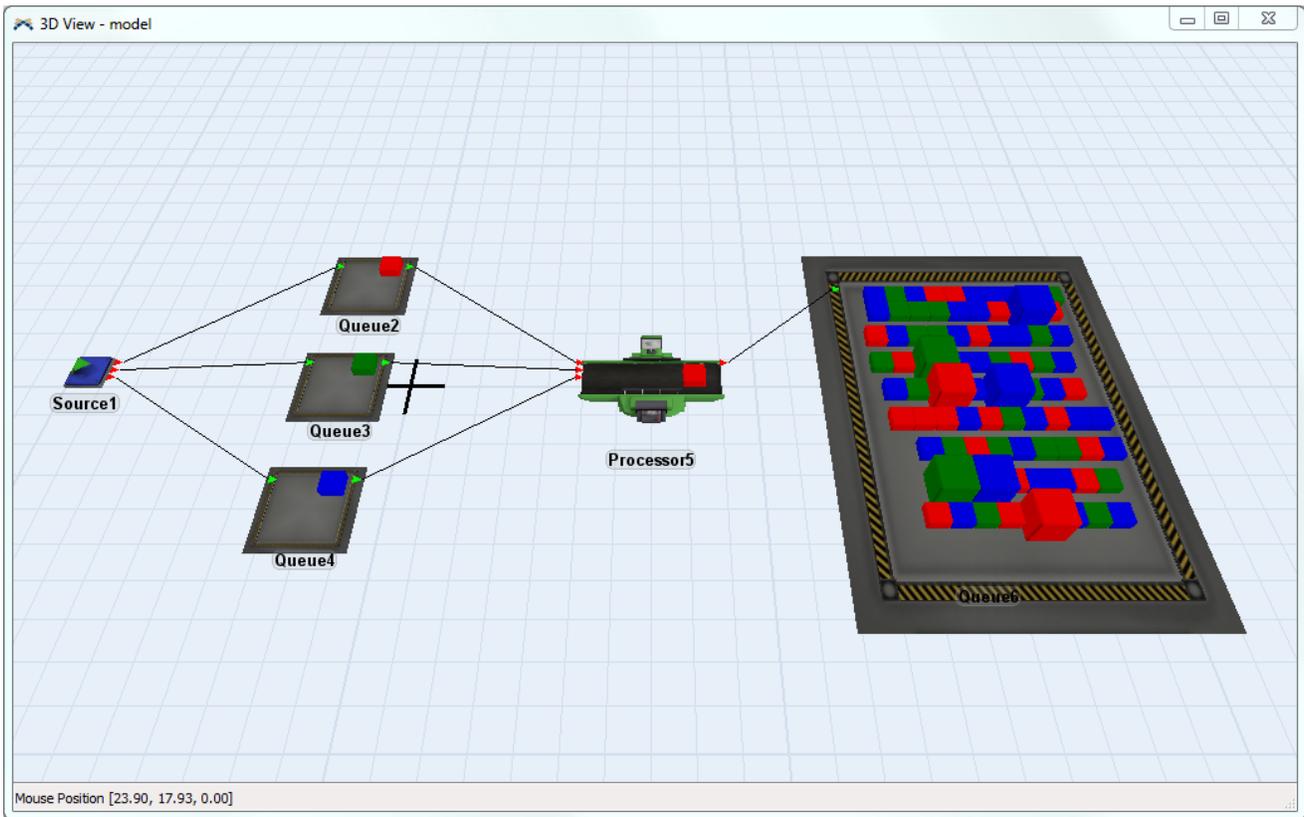
- Abra a janela de propriedade dos *Processor's*.
- Vá para a guia *Triggers*.
- Clique sobre o botão de edição do código  próximo a trigger *OnProcessFinish*.
- Escreva uma declaração que irá permitir apenas dez item de cada tipo ser configurado para um tamanho específico (por vez):

```
if (getlabelnum(item, "itemtypeorder") % 10 == 0)
{
    setsize(item,1,1,1);
}
```

O décimo item a ser criado (por tipo de item, ou seja, item 1, 2 ou 3) terá um tamanho maior em relação aos demais itens do mesmo tipo. Ao terminar toda a configuração rode o modelo e pare o modelo exatamente no momento que for criado o primeiro item de tamanho maior. Dê um duplo clique sobre ele e abra a guia label. Dê um duplo clique no Source e abra a guia label para verificar o que está acontecendo

Nota: Você não precise fazer isto mais duas vezes, por que isso se aplica a qualquer flowitem com a label "itemtypeorder".

- Clique em **OK** na janela de edição de código assim como na janela de propriedades.
- **Resete (Reset)** e **Rode (Run)** o seu modelo. Seu modelo deve parecer-se com algo conforme Guiaixo:



Isto completa o tutorial de labels. Parabéns!

Tutorial das Ferramentas Globais de Simulação

1. Introdução
2. Construção do modelo passo a passo

Introdução

Este exercício tem como objetivo introduzi-lo ao uso de conceitos globais de modelagem com o FlexSim que podem ser usados para diversas finalidades no modelo. Estas ferramentas podem simplificar a modelagem e permitir mudanças dinâmicas em seu modelo. Será apresentado também algumas aplicações dos objetos Combiner e Separator na vida real.

Nota: Para mais informações sobre as ferramentas de modelagem global, veja o tópico Modeling Tools.

O que você irá aprender

- Como criar e acessar as Global Tables
- Como criar e acessar as Global Variables e os Global Macros
- Como criar um simples User Command para uso no modelo
- Como usar os objetos Combiner e Separator

Novos Objetos

Neste exercício você será introduzido aos objetos Combiner e Separator.

Tempo aproximado para completar este exercício

Este exercício levará de 45-60 minutos para ser completado.

Overview do Modelo

Neste modelo será apresentado alguns modos diferentes de executar algumas tarefas em oposição aos modos apresentados nos modelos anteriores. Ambos métodos geram o mesmo resultado, mas alguns usuários preferem um enquanto outros preferem o outro.

Ainda veremos os Combiners e os Separators e como eles podem ser usados no modelo para representar objetos da vida real

Construção do Modelo passo a passo

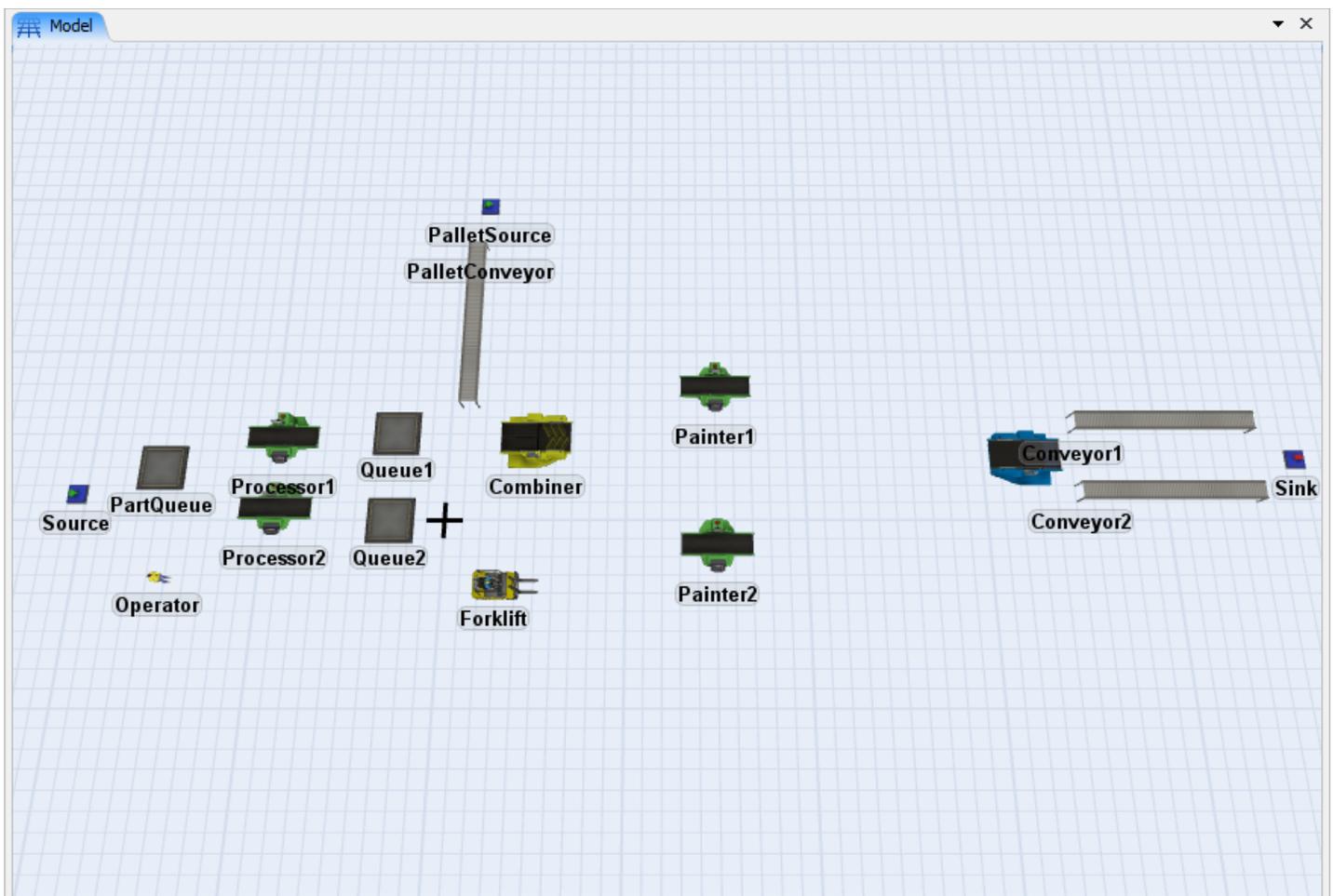
Construção do Modelo com Ferramentas Globais de Simulação

Inicie um novo modelo ao clicar no botão  que encontra-se na barra de ferramentas. Clique em **OK** na janela de unidade do modelo “Model Units”; nós iremos usar as unidades padrões para este modelo.

Se você encontrar alguma dificuldade enquanto constrói este modelo, um tutorial totalmente funcional e completo pode ser encontrado em <http://www.flexsim.com/tutorials>.

Etapa 1: Crie os objetos

- Arraste os objetos da “Library Icon Grid” (biblioteca) para a janela “3D View” para criação dos objetos mostrados na imagem abaixo.
- Renomeie os objetos conforme mostrado Guiaixo, pois os mesmos serão referenciados pelo nome no tutorial.



Etapa 2: Conecte as portas

Conecte todos os objetos conforme descrito abaixo.

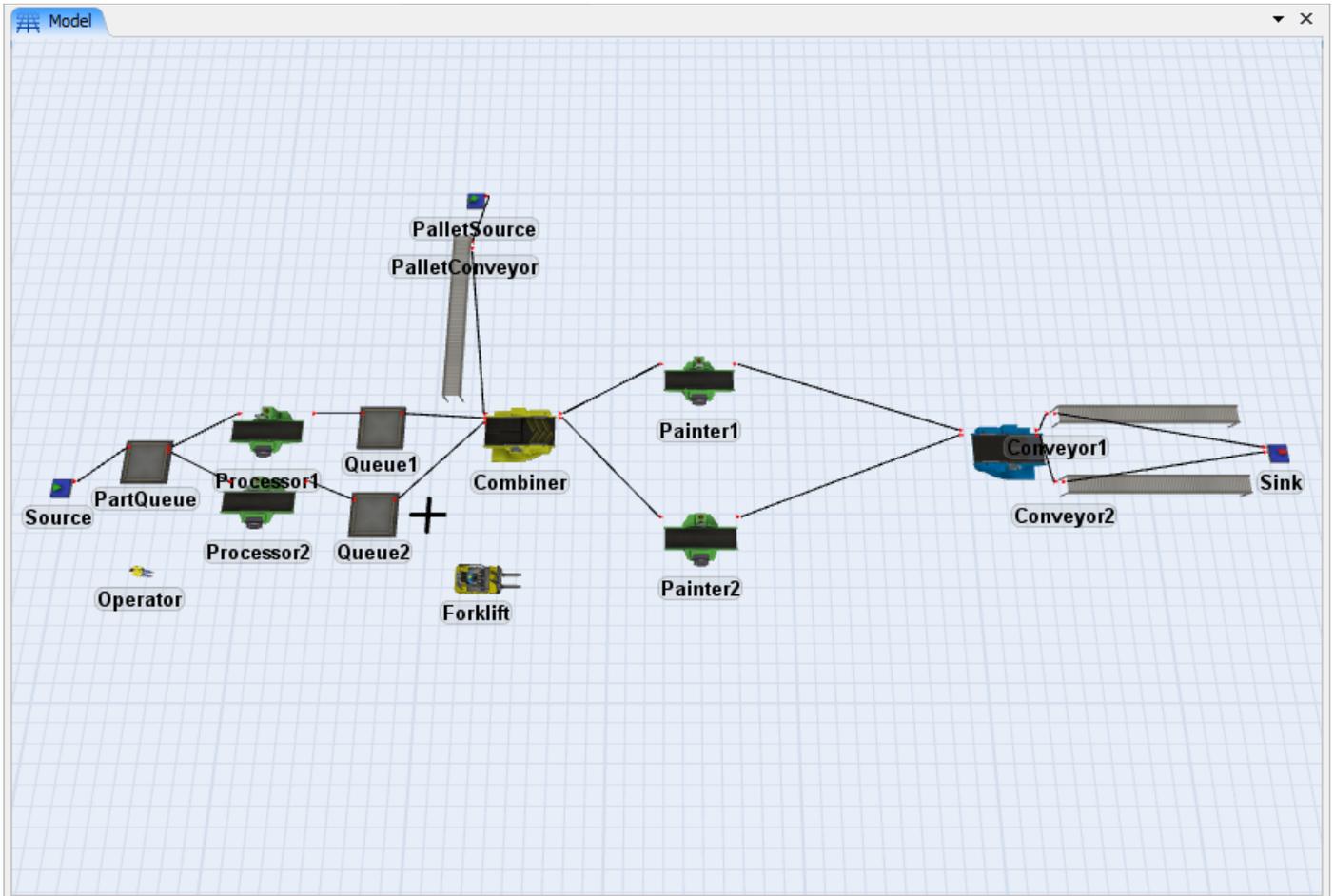
- Conecte o *Source* ao *PartQueue*.
- Conecte o *PartQueue* ao *Processor1* e ao *Processor2*
- Conecte o *Processor1* ao *Queue1* e o *Processor2* ao *Queue2*

- Conecte o *PalletSource* ao *PalletConveyor*
- Conecte o *PalletConveyor* ao *Combiner*

Nota: É necessário que o *PalletConveyor* esteja conectado a primeira porta de entrada do *Combiner*.

- Conecte o *Queue1* e o *Queue2* ao *Combiner*
- Conecte o *Combiner* ao *Painter1* e ao *Painter2*
- Conecte o *Painter1* e o *Painter2* ao *Separator*
- Conecte o *Separator* ao *Conveyor1* e ao *Conveyor2*
- Conecte o *Conveyor1* e o *Conveyor2* ao *Sink*

Note que não foi feita nenhuma conexão central ao *Operator* ou a *Forklift*. Nós faremos estas referências através de Global Variables.



Etapa 3: Definição das propriedades do source

Definição do Source para criar 2 itemtypes.

- Deixe a opção do campo **Inter-Arrival Time** do *Source* como *exponential(0, 10, 0)*.
- Defina na trigger **OnCreation** a opção **Set Itemtype and Color** com a distribuição *duniform(1,2)*.
- Clique em **OK** para aplicar as alterações e fechar a janela

Etapa 4: Configure as propriedades do PalletSource

Os pallets agirão como um suporte para as peças que irão para o *Combiner* e para o *Painter*. Definindo o campo *FlowItem Class* para pallet irá fazer com que o *Combiner* empilhe os itens ordenadamente em cima do pallet.

Este pallet também será usado para determinar quantos itens serão colocados em cima do pallet pelo *Combiner*. Isso será definido através da definição do *itemtype* do pallet. Este procedimento será explicado depois.

- Defina o campo **FlowItem Class** do *PalletSource* para Pallet.
- Para ter um fornecimento infinito de pallets, defina o campo **Inter-Arrival Time** para 0.
- Vá até a guia **Triggers** e adicione uma trigger *Set Type, Name or Label* no campo **OnCreation**.
- Altere o campo **Type** para *duniform(1,6)*.
- Clique em **OK** para aplicar e fechar a janela de propriedades.

Neste ponto fique à vontade para testar o seu modelo. Você deverá ver os itens fluindo sobre todo o modelo, combinando um pallet e um item do *Queue1* e outro item do *Queue2*. E então separando estes itens no *Separator*.

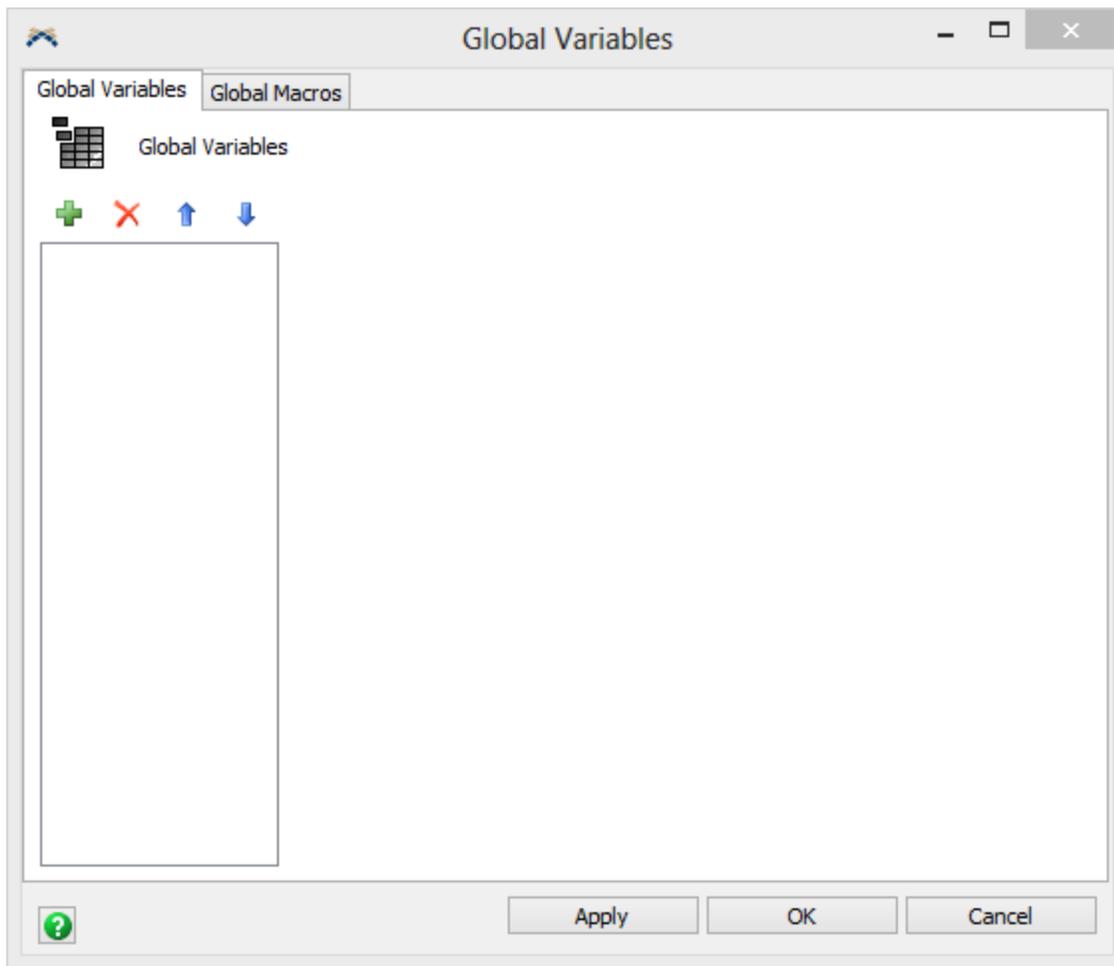
Etapa 5: Adicione os Operadores

Para demonstrar um outro modo de referenciar objetos, não usaremos as portas centrais para conectar este operador. Ao invés disso, usaremos a ferramenta *Global Variables*.

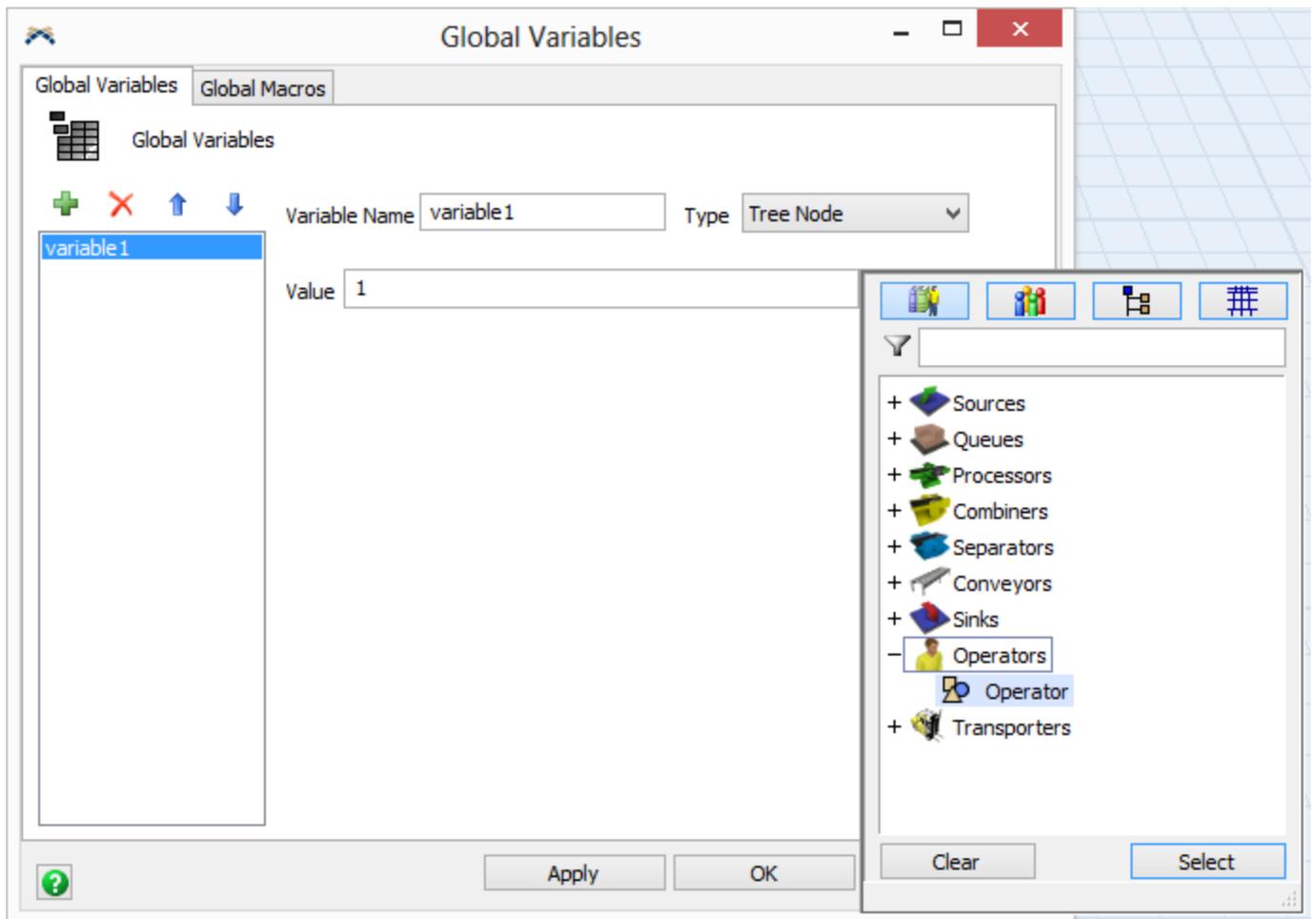
Definiremos também um *Global Macro* para definir o tempo de processo dos dois processores.

Veja a janela do tópico *Global Variables* para aprender mais sobre as *Global Variables*.

- Crie as **Global Variables** em Toolbox e clique no  e depois em “**Modeling Logic**” para então selecionar *Global Variables*

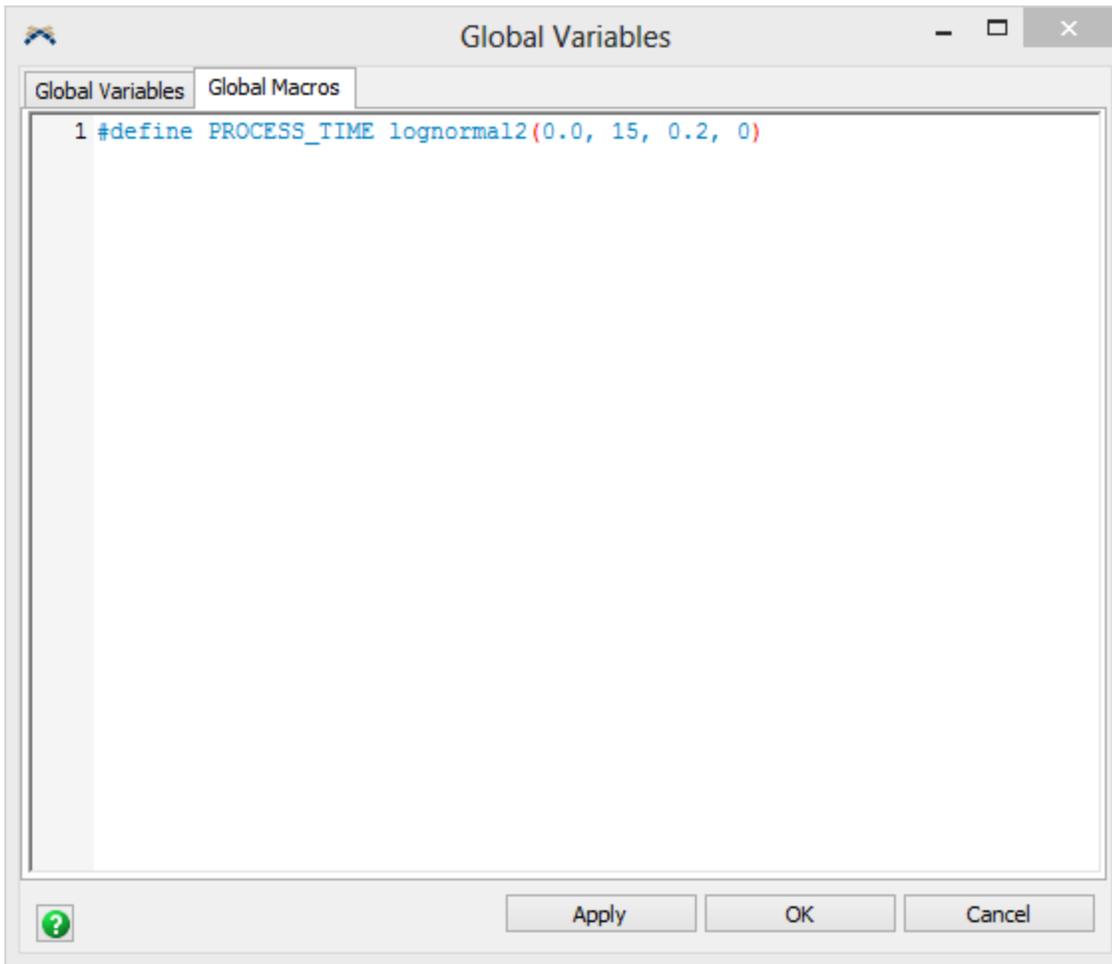


- Clique **+** e configure o **Variable Name** para Operador (Operator)
- Selecione o Tree Node em **Type**.
- Clique em **+** próximo ao campo **Value** e selecione o Operador como value e então clique em **Select**.



- Repita os próximos passos para criar outra variável para o *Forklift*.
- Próximo, clique sobre a guia **Global Macros** e entre com o seguinte código:

```
#define PROCESS_TIME lognormal2(0.0, 15, 0.2, 0)
```



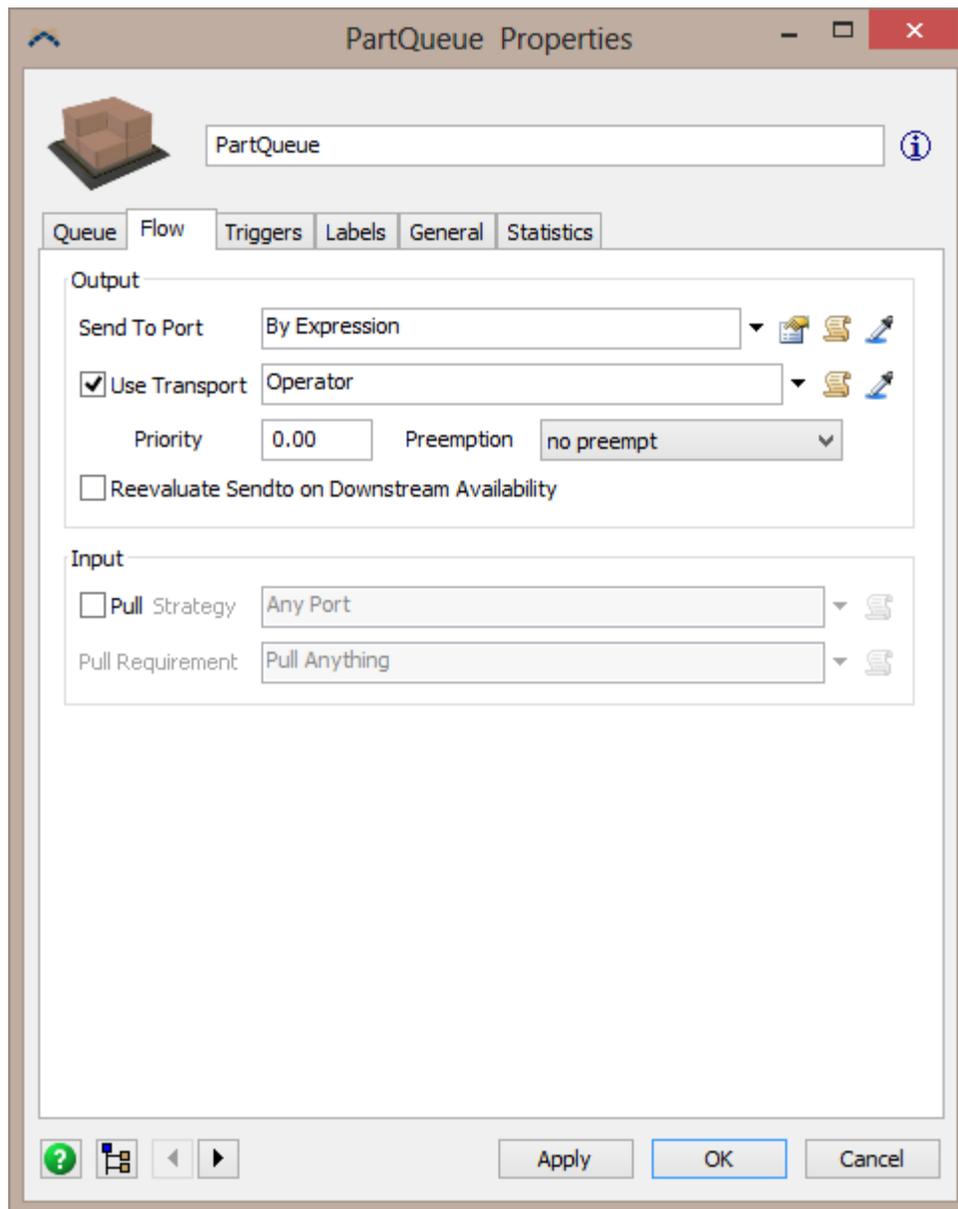
- Clique em **OK** para aplicar e feche a janela **Global Variables**.

Etapa 6: Configure as filas (Queues)

O *Processor1* irá processar todos os itens com *itemtype* igual a 1 e o *Processor2* processará todos os itens com *itemtype* igual a 2.

- Defina a variável **Maximum Content (Capacidade máxima)** do *PartQueue* para 25.
- Na guia **Flow**, selecione **By Expression** no campo **Send To Port**. Por padrão, esta opção tem *getitemtype(item)*.

Cheque a opção **Use Transport** e altere o texto *centerobject(current, 1)* para By Name através do pick list, e clique no sinal **+** para escolher o Operator e selecione.

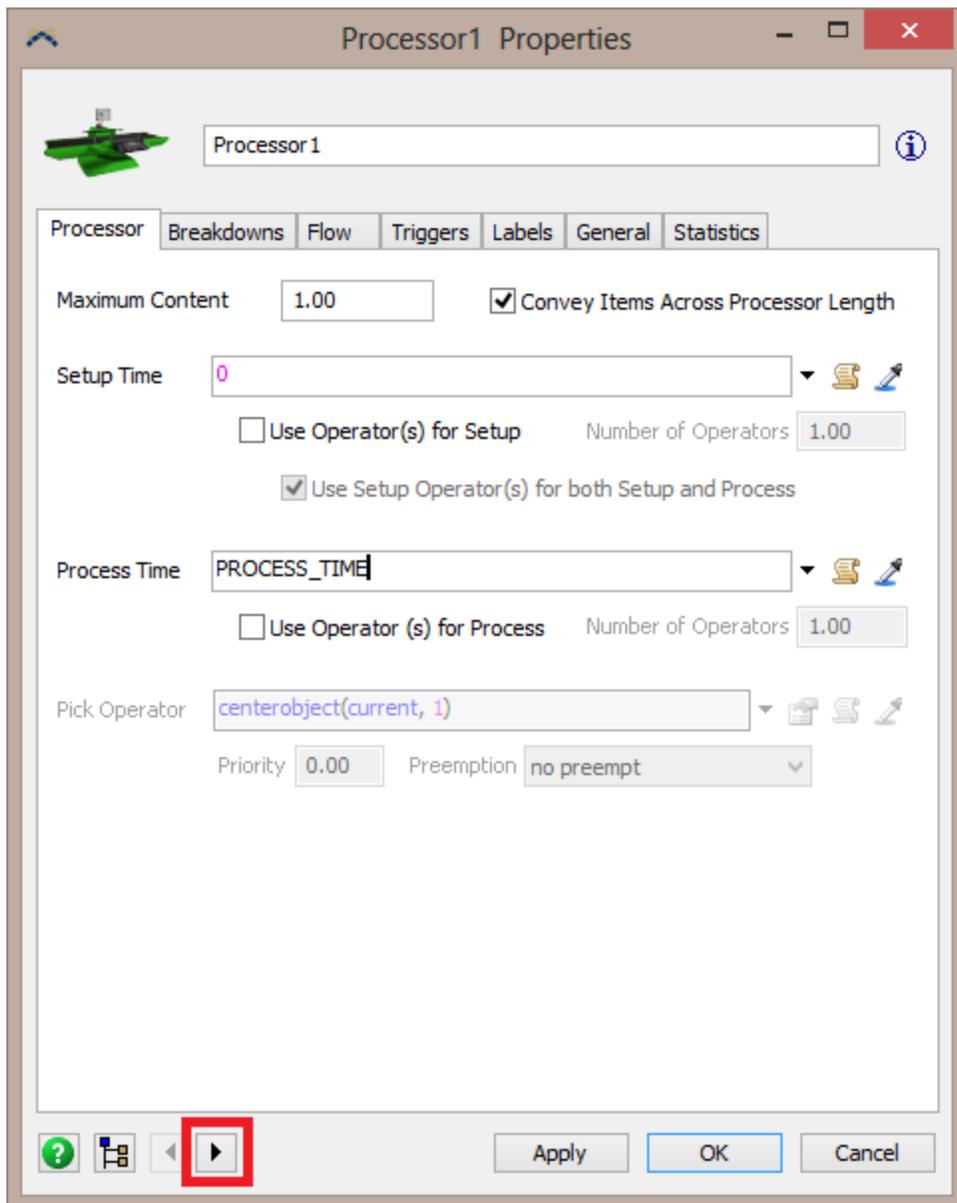


- Clique em **OK** para aplicar e fechar a janela de propriedades.
- Configure a capacidade máxima (**Maximum Content**) do Queue1 e Queue2 para 100.

Etapa 7: Configure os Processors

A Global Macro que será usada para definir os tempos de processos já foi definida na etapa anterior.

- Abra as propriedades do *Processor1* e digite PROCESS_TIME na caixa **Process Time**.



- Clique na **Seta para Direita** no canto inferior esquerdo nas propriedades do *Processor* para mover para o próximo processor.
- Defina o campo **Process Time** do *Processor2* para PROCESS_TIME.
- Clique em **OK** para aplicar e fechar a janela de propriedades.

Etapa 8: Configure a Global Table

Iremos agora criar uma Global Table para utiliza-la junto ao *Combiner*.

O Combiner é usado para agrupar diversos itens. O processo realizado é o seguinte:

- Primeiramente o Combiner receberá um único item pela porta de entrada número 1. Este item automaticamente se torna o “recipiente” que armazenará os outros itens.
- Uma vez que o primeiro item foi recebido, o Combiner inicia a coleta dos outros itens das portas restantes de acordo com a tabela “Components List” na guia Combiner.

Components List

	Target Quantity
From Input Port 2	2.00
From Input Port 3	3.00

- Assim que o lote estiver montado, o Combiner executa os campos "setup" e em seguida "process time" antes de enviar o item composto por vários outros itens para a próxima etapa.

Para este modelo, teremos diferentes composições para diferentes tipos de pallets, representando diferentes pedidos criados em uma indústria. Como conectamos a saída de pallets a entrada número 1 do Combiner, estes pallets podem definir qual a lista usada para a composição do "pedido".

Para isso, criaremos uma Global Table que será usada para armazenar uma lista com as diferentes composições das ordens de cada cliente (component list) para uso do combiner.

- Crie uma **Global Table** indo até a opção **Tools > Global Table > Add**.
- Nomeie a tabela como *PartsList*.
- Defina o número de linhas para 2 e o número de colunas para 6.
- Entre com os seguintes valores na sua tabela.

	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	3.00	5.00	4.00	2.00	5.00	4.00
Row 2	5.00	2.00	4.00	3.00	4.00	5.00

- Feche a janela da Global Table.

Etapa 9: Configure o Combiner

- Abra as propriedades do *Combiner* e clique na guia **Combiner**.
- Selecione **Convey Items Across Combiner Length**. Isso significa que os itens irão ser empilhados no pallet em cima do Combiner e depois se movimentarão juntos de uma vez sobre a extensão de todo o combiner.
- Clique na guia **Flow** e cheque a caixa **Use Transport**.
- Substitua *centerobject(current,1)* por *ByName* dos picklists, e clique no botão  para escolher o Forklift e clique em **Select**.

Queremos que o campo *Component List* se atualize cada vez que um pallet entrar pela porta número 1.

- Clique na guia **Triggers** e adicione uma nova operação no campo **OnEntry**.
- Selecione a opção *Update Combiner Component List* dentre as opções disponíveis.
- Substitua "*tablename*" para "*PartsList*".
- Clique em **OK** para aplicar as configurações e fechar a janela *Properties*.

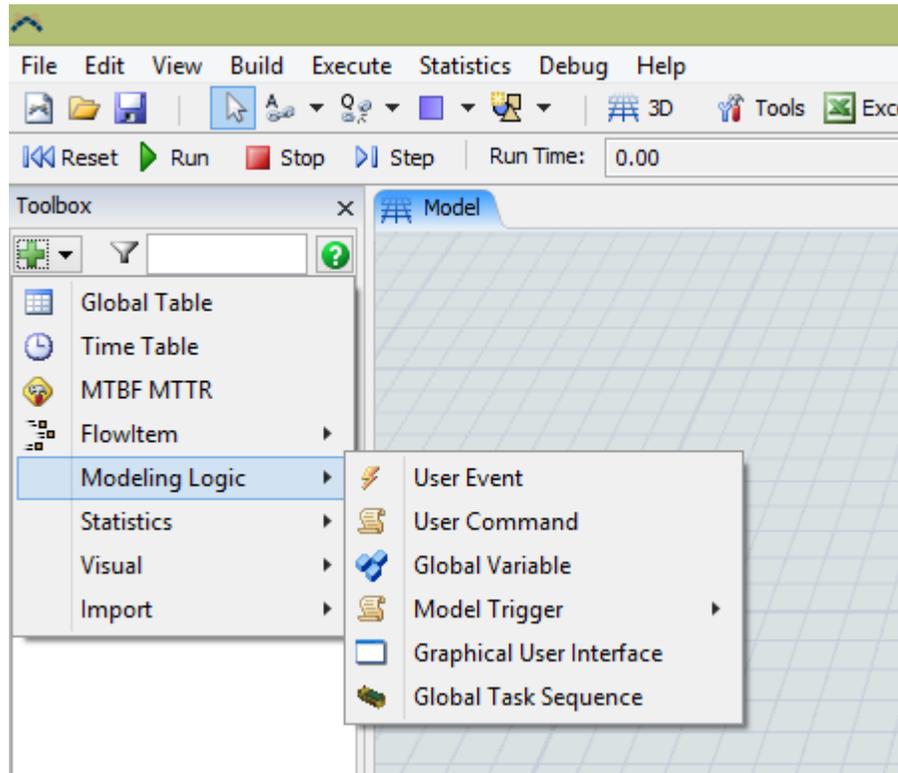
A opção *Update Combiner Component List* "puxa" os dados da Global Table de acordo com o itemtype do item que entra pela porta número 1 do combiner. Sendo assim, se o pedido for igual a 4, o combiner atualizará a tabela *Component List* de acordo com as duas linhas da coluna 4 da Global Table *PartsList*.

Etapa 10: Escreva Comandos Customizados

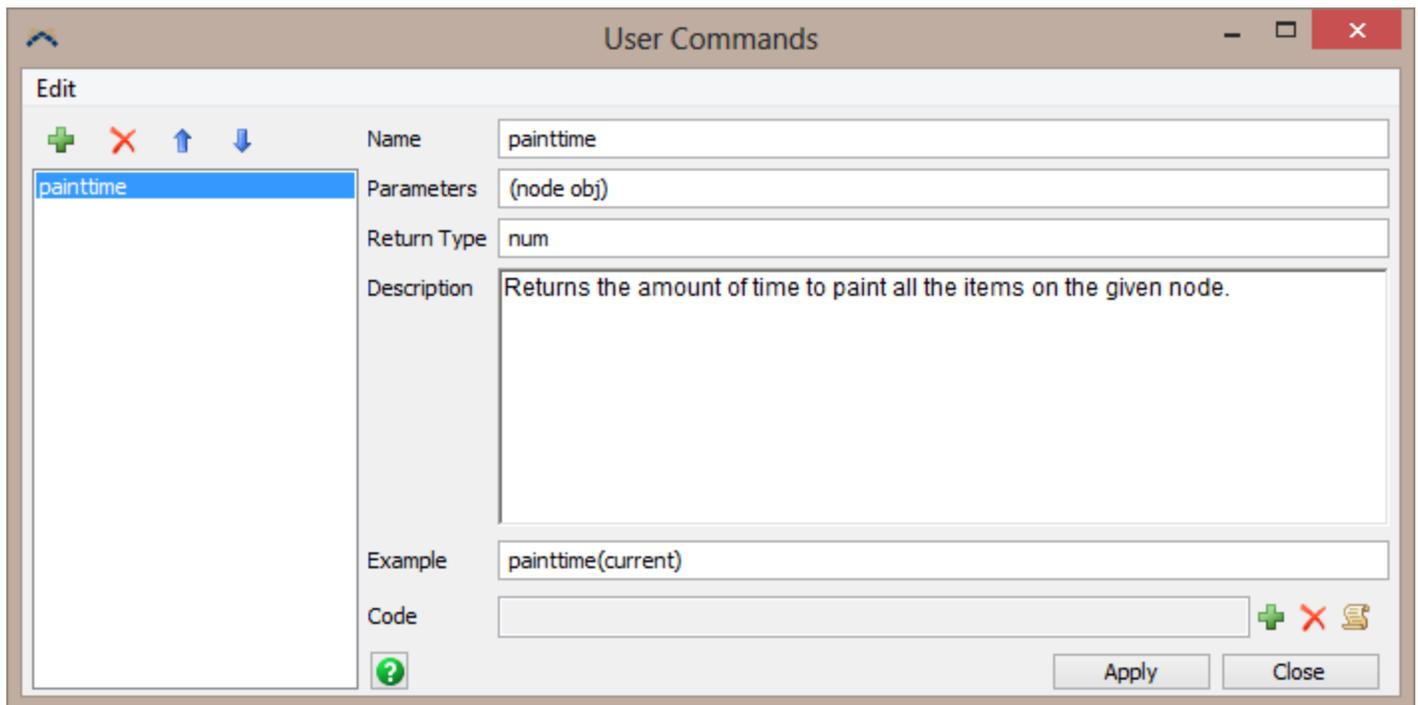
O FlexSim lhe permite escrever a sua própria lógica customizada e utiliza-la como uma função em qualquer lugar do modelo.

O primeiro comando que escreveremos retornará o tempo estimado de pintura dos itens que estão nos pallets. Este código pode ser escrito diretamente no campo "process time" dos processores *Painter1* e *Painter2*, mas caso tivéssemos diversos equipamentos de pinturas, seria muito mais conveniente escrever o código uma única vez e utiliza-lo em todos os objetos do que escrever diversas vezes.

- Abra a ferramenta **User Commands** conforme tela abaixo.



- Clique em **+** para criar um novo comando.
- Defina o campo **Name** para o nome *painttime*.
- Defina o campo **Parameters** para *(node obj)*.
- Defina o campo **Return Type** para *num*. **(informa que precisa retornar um número neste caso)**
- E a caixa **Description** para: *Retorna o tempo necessário para realizar a pintura de todos os itens de um determinado node.*
- Defina o campo **Example** para *painttime(current)*.



Para o código, queremos que o comando identifique quantos itens estão no pallet e passe ao processador o tempo total para pintura dos mesmos. Sendo assim, cada item com o itemtype igual a 1 demora 20 segundos para ser pintado e itens com itemtype igual a 2 demoram 14 segundos para ser pintado.

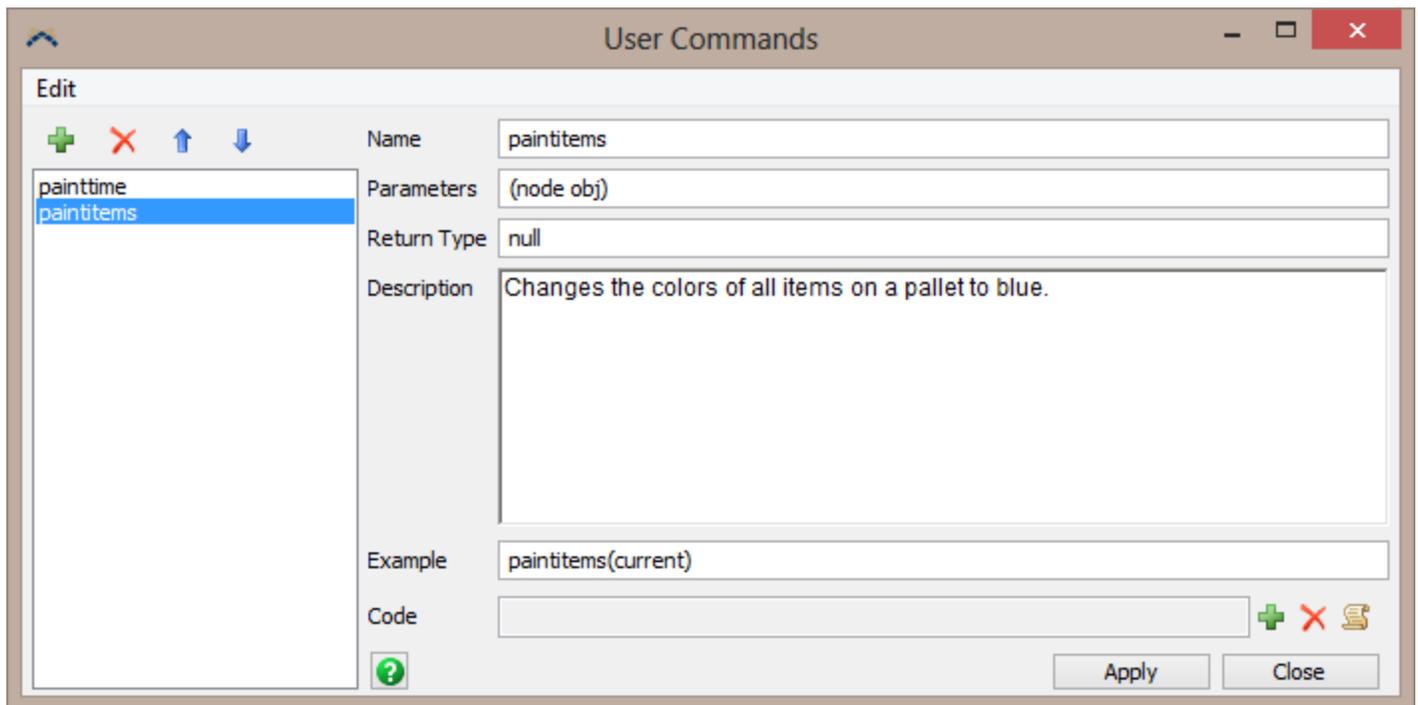
- Clique no botão  para abrir a janela de programação.
- Digite o seguinte código:

```
treenode object = parnode(1);
int painttime = 0;
for(int index = 1; index < content(first(object)); index++) {
    if(getitemtype(rank(object,index)) == 1) {
        painttime += exponential(0.0, 20.0, 0);
    } else {
        painttime += exponential(0.0, 14.0, 0);
    }
}
return painttime;
```

- Clique em **OK** para aplicar as configurações e fechar a janela.

Depois disso, criaremos um comando que mudará a cor de todos os itens que estão no pallet para azul após passarem pelos processos de pintura.

- Adicione um novo User Command e defina o seu **Name** para *paintitems*.
- Defina o campo **Parameters** para *(node obj)*.
- Defina o campo **Return Type** para *null*. **(informa que nada precisa retornar neste caso)**
- Defina a caixa **Description** para: *Define a cor de todos os itens que estão sobre o pallet para azul*
- Defina o campo **Example** para *paintitems(current)*.



Queremos que o código identifique todos os itens que estão em cima do pallet e os pinte de azul.

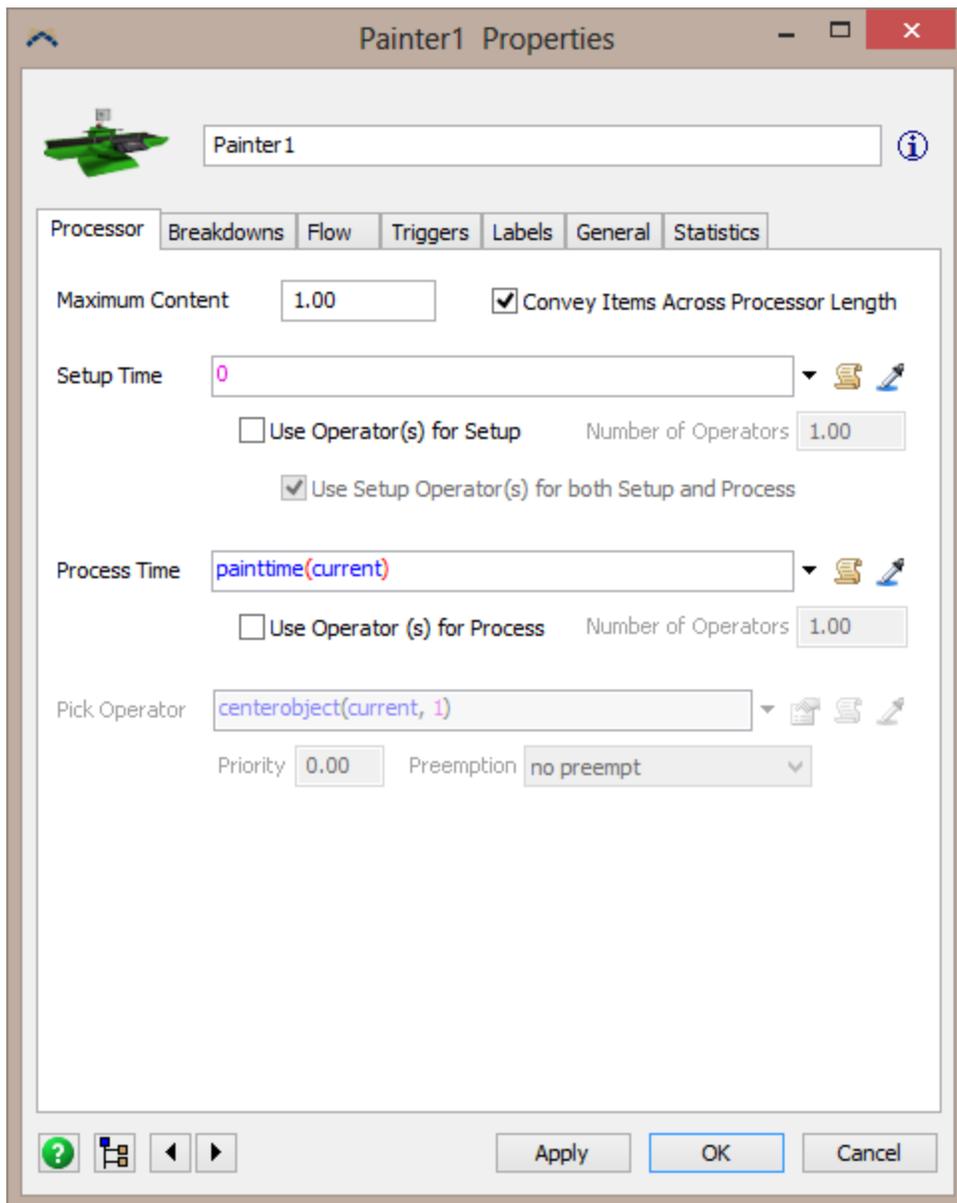
- Clique no botão  para abrir a janela de programação.
- Entre com o seguinte código:

```
treenode object = parnode(1);
for(int index = 1; index <= content(first(object)); index++) {
    colorblue(rank(first(object), index));
}
```

- Clique em **OK** para aplicar as configurações e fechar a janela.
- Clique em **Apply** e depois clique em **Close** para fechar a janela do **User Commands**.

Etapa 11: Configure os Painters

- Abra as propriedades do *Painter1*.
- Defina o campo **Process Time** para *painttime(current)*.



- Clique na guia **Flow** e cheque a opção **Use Transport**.
- Defina o transporte para *Forklift*.
- Clique na guia **Triggers** e clique no ícone  que está em frente a opção **OnProcessFinish** para editar o código presente neste campo.
- Adicione o seguinte comando na última linha disponível da janela:

```
paintitems(current);
```

- Clique em **OK** para aplicar as configurações e fechar a janela de códigos.
- Clique em **OK** para aplicar e fechar a janela Properties.

Faça o mesmo para o *Painter2*. Para copiar tudo isso de uma maneira mais rápida de um objeto para outro, veja a ferramenta Edit Selected Objects Utility.

Etapa 12: The Separator

Como atualmente o *Separator* se encontra configurado não será necessário fazer nenhuma alteração no mesmo. Por padrão, o *Separator* recebe os itens e separa o recipiente do conteúdo, enviando assim o recipiente (pallet) para a porta de saída número 1 e o conteúdo (as caixas) para a porta número 2. Caso seja de seu interesse alterar essas configurações, vá até a guia **Flow** do *Separator* e altere a opção do campo **Send To Port**.

Estamos agora prontos para rodar a simulação.

Etapa 13: Resete (Reset) e Rode (Run) o Modelo

- Lembre-se de clicar no botão  **Reset** para resetar os parâmetros do modelo para o estado inicial.
- Clique no botão  **Run** para iniciar a simulação.

Você deverá ver os itens chegando ao *Queue* e então sendo movidos para um dos *processors* através dos operadores. Itens vermelhos vão para o *Processor1* e itens verdes vão para o *Processor2*.

Os pallets serão colocados no *Combiner* e então aguardam até receber todos os componentes itens necessários do *Queue1* e do *Queue2*. A empilhadeira então move este pallet para os equipamentos de pintura *Painter1* e *Painter2*. Antes de deixar os *Painters*, todos os itens ficarão da cor azul. O pallet então irá para o *Separator* e fará a separação do pallets e dos produtos, enviando o pallet para o *Conveyor1* e os itens para o *Conveyor2* e posteriormente para o *Sink*.

Esta etapa complete o Exercício de Modelagem com Ferramentas Globais. Parabéns!!

Tutorial sobre User Events

1. Introdução
2. Construção do Modelo passo a passo

Introdução

Este tutorial irá ajudá-lo no entendimento do uso de User Events em seu modelo de simulação. Este exercício assume que você saiba usar variáveis globais. Se você não sabe, por favor veja o tutorial sobre o uso de ferramentas globais de simulação (Global Modeling Tools).

O que você irá aprender

- Como usar a ferramenta User Events.

Tempo aproximado para completar este exercício

Este exercício levará de 15-20 minutos para ser completado.

Overview do Modelo:

Neste modelo, nós iremos realocar os operadores toda vez que resetarmos o modelo, assim como configurar a taxa entre chegadas (inter-arrival) e o número de processadores a ser usado. A um determinado tempo no modelo, nós iremos alterar a taxa de chegada das peças e abrir equipamentos (processors) adicionais.

Construção do Modelo passo a passo

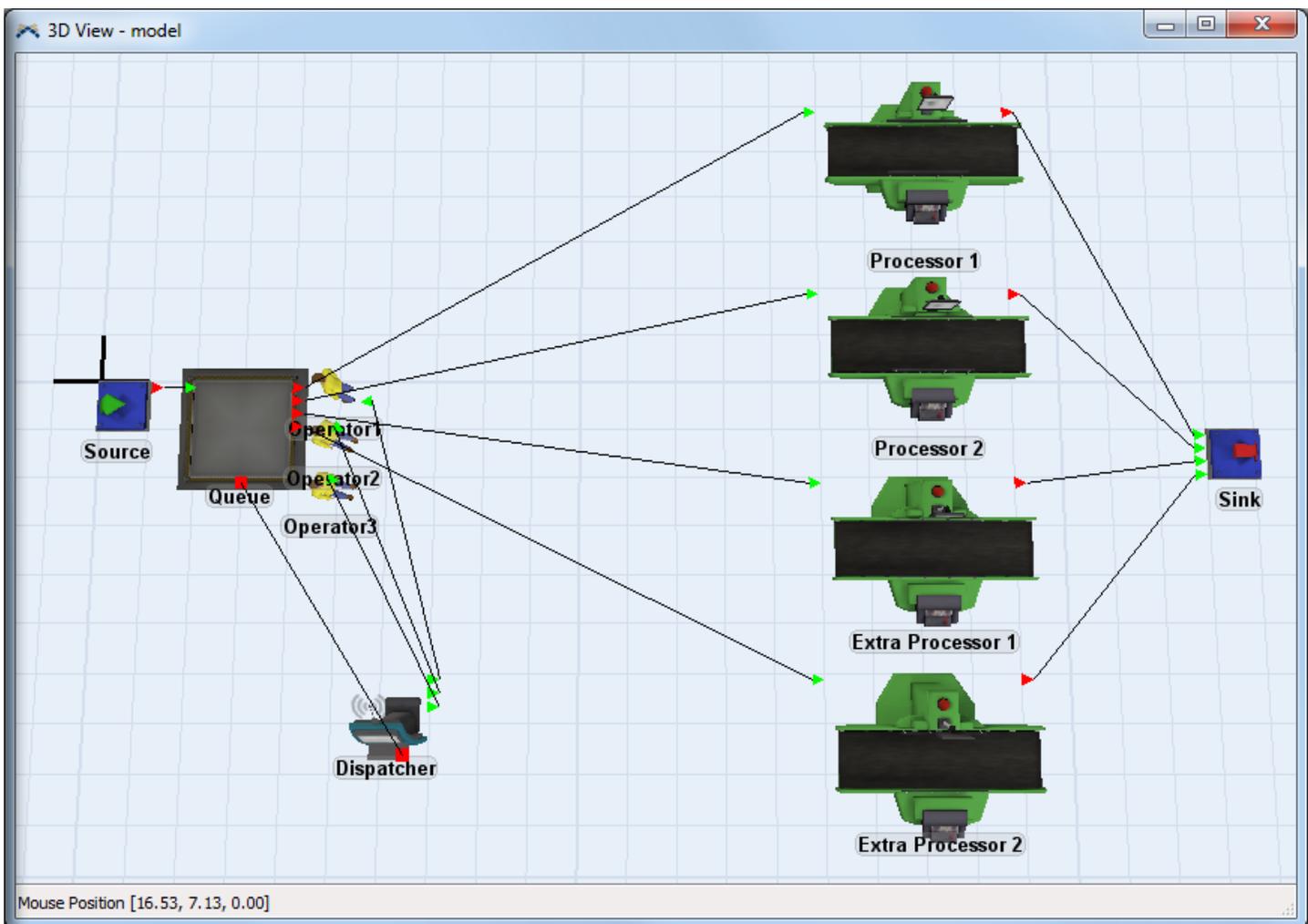
Construa o layout do modelo

Inicie um novo modelo ao clicar no botão  na barra de ferramentas. Clique em OK na janela de unidades do modelo, nós iremos usar a unidade padrão para nosso modelo.

Se a qualquer momento você encontrar dificuldades enquanto constrói este modelo, um tutorial completo e funcional poderá ser encontrada em <http://www.flexsim.com/tutorials>.

Etapa1: Crie os Objetos

- Arraste os objetos de sua biblioteca (Library Icon Grid) para a superfície de simulação 3D para criar seu modelo conforme criado agora.
- Renomeie os objetos como mostrado abaixo.



Conecte todos os objetos conforme informado abaixo:

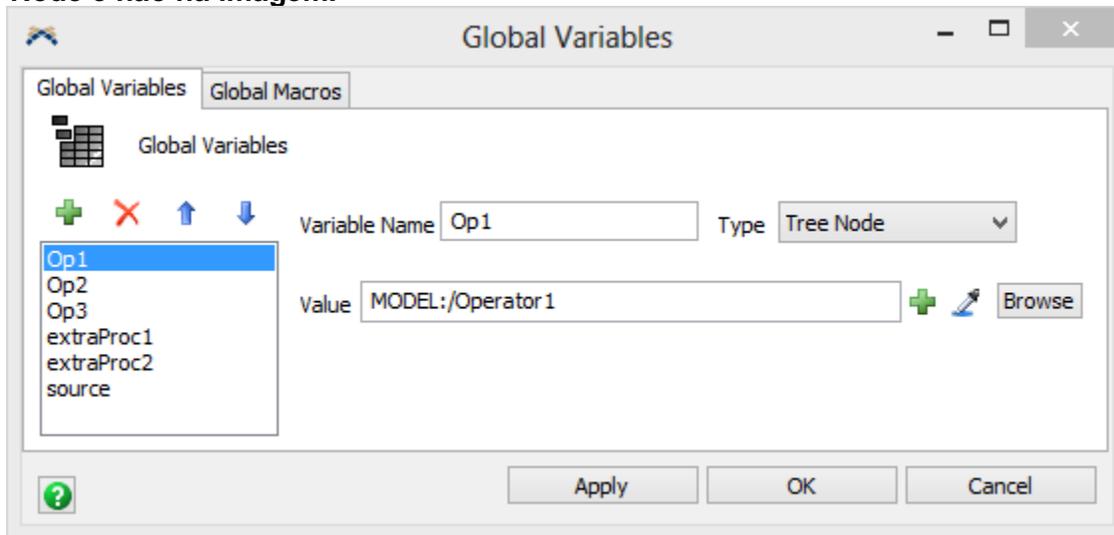
- Conecte o *Source* ao *Queue*.
- Conecte o *Queue* ao *Processor 1*, *Processor 2*, *Extra Processor 1* e *Extra Processor 2*.
- Conecte o *Processor 1*, *Processor 2*, *Extra Processor 1* e *Extra Processor 2* com o *Sink*.
- Conecte o *Queue* ao *Dispatcher* com conexão de porta central (tecla S)
- Conecte o *Dispatcher* ao *Operator1*, *Operator2* e *Operator3* com a conexão padrão (tecla A).

Etapa 2: Configure as variáveis globais (Global Variables)

Aqui nós iremos criar as variáveis globais para objetos que nós iremos acessar em nosso User Event. Alternativamente, objetos podem ser acessados pelo comando node:

```
Treenode Op1 = node ("/Operator1", model());
```

- Abra a janela **Global Variables** (Tools > Global Variables).
- Crie uma variável global para cada operador, para os dois processadores extra, e para o source. Cada um deles será do tipo Tree Node, nomeado conforme abaixo: **CUIDADO: Busque os objetos no Tree Node e não na imagem.**



Etapa 3: Configure o Source e a fila (Queue)

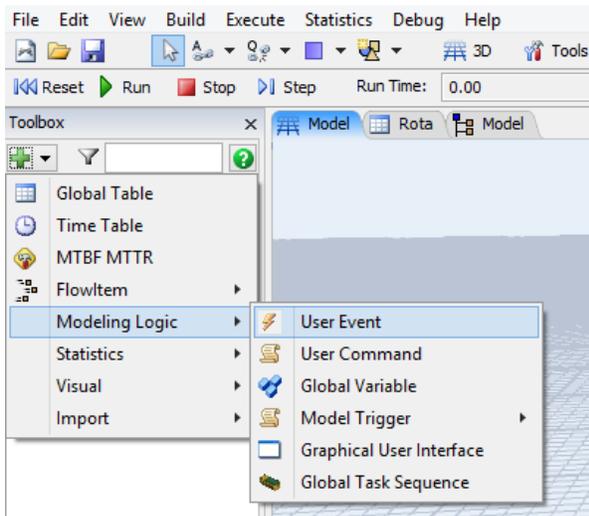
- Abra a janela de propriedade do Source.
- Vá para a guia **Labels**.
- Crie uma number label chamada "arrivalTime", deixe a configurado em 0.
- Vá para a guia Source.
- Para o **Inter-arrival time** entre com *getlabelnum(current, "arrivalTime")*.
- Clique em Ok para aplicar e feche a janela de propriedades.

- Abra a janela de propriedades do Queue.
- Vá para a guia **Flow**.
- Marque **Use Transport**.

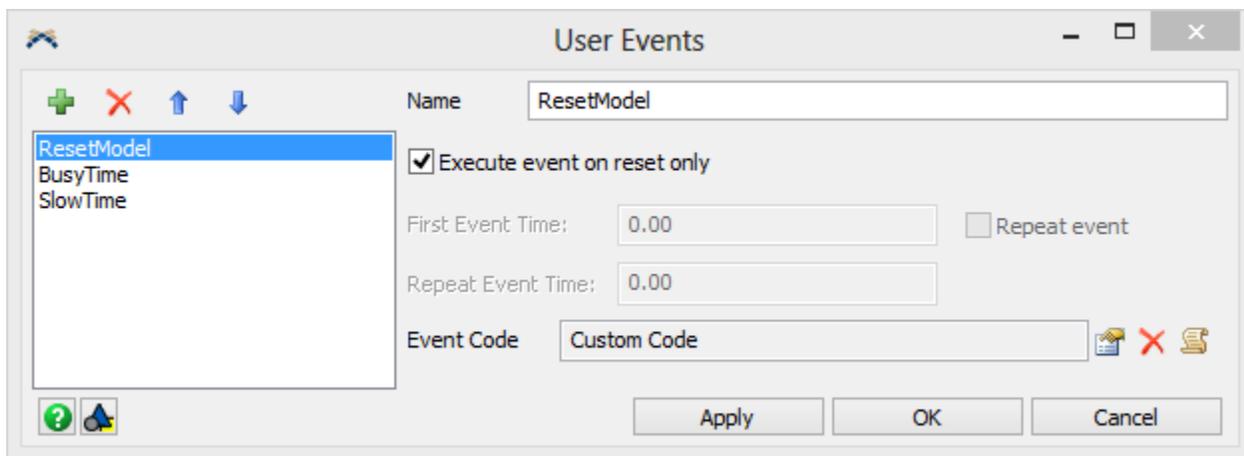
Etapa 4: Crie o User Events

Nós iremos criar três user events. Um que dispara quando o modelo reseta e dois para alterar a "ocupação" do modelo.

- Abra a janela **User Events** conforme tela abaixo.



- Adicione três novos eventos e nomeie eles "*ResetModel*", "*BusyTime*" e "*SlowTime*".



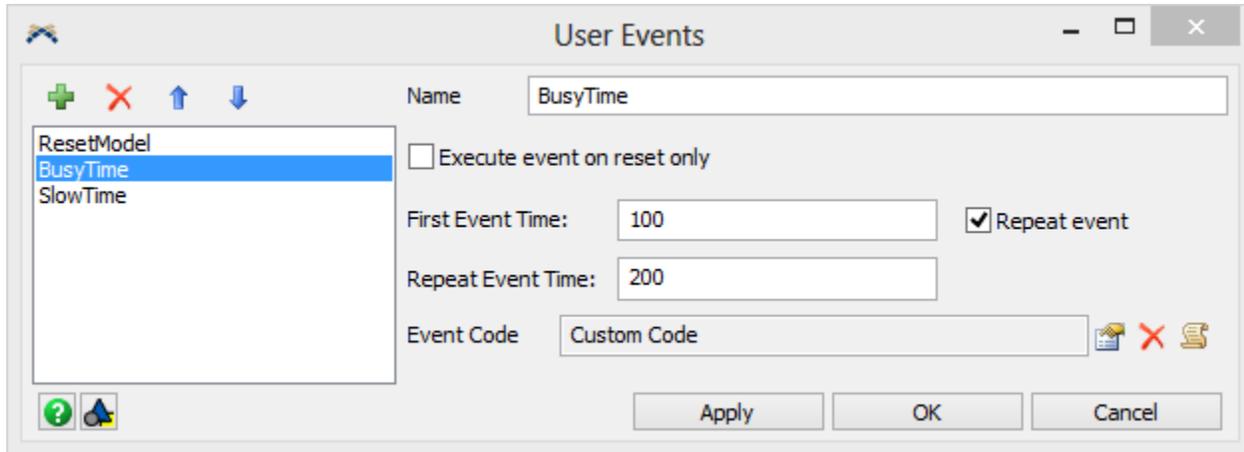
- Com o *ResetModel* selecionado, marque **Execute event on reset only**. Toda vez que você pressionar o botão Reset, o código será executado.
- Clique no botão de edição de código  e escreva o seguinte código:

```
setloc(Op1, 5, 0, 0);
setloc(Op2, 5, -1, 0);
setloc(Op3, 5, -2, 0);
setlabelnum(source, "arrivalTime", 10);
```

```
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 setloc(Op1, 5, 0, 0);
5 setloc(Op2, 5, -1, 0);
6 setloc(Op3, 5, -2, 0);
7
8 setlabelnum(source, "arrivalTime", 10);
```

Isto irá configurar a localização de cada operador para próximo ao queue (se o seu queue não está no mesmo lugar que o meu [x:2.00, y:0.00, z:0.00], configure a localização você mesmo de seu queue e entre com os valores apropriados de x,y, e z). Este user event irá também configurar label do source para 10, então, o intervalo entre chegadas (Inter-arrival time) irá alterar-se de acordo.

- Com BusyTime selecionado, entre 100 para **First Event Time** e 200 para **Repeat Event Time**. Cada 200 segundos o modelo ficará como busy time. Marque a caixa **Repeat Event**.



Clique no botão de edição de código  e escreva o seguinte código:

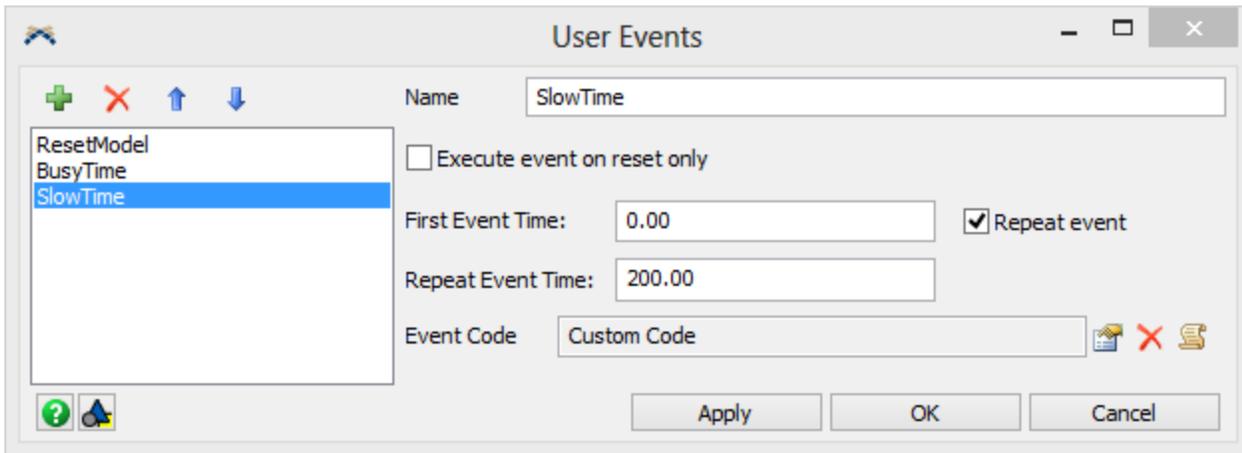
```
setlabelnum(source, "arrivalTime", 5);
openinput(extraProc1);
openinput(extraProc2);

msg("Busy Time", "It's Busy Time!");
```

```
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 setlabelnum(source, "arrivalTime", 5);
5
6 openinput(extraProc1);
7 openinput(extraProc2);
8
9 msg("Busy Time", "It's Busy Time!");
```

Isto irá configurar o tempo entre chegadas para 5, então as peças são criadas duas vezes mais rápidas. Os processadores extra irão abrir e portanto podem ser usados durante o tempo ocupado (busy time). Você receberá uma mensagem falando que você está no tempo ocupado.

Com o SlowTime selecionado, entre 0 para **First Event Time** e 200 para **Repeat Event Time**. A cada 200 segundos o modelo irá correr em tempo lento, iniciando no tempo 0. Selecione a caixa **Repeat Event**.



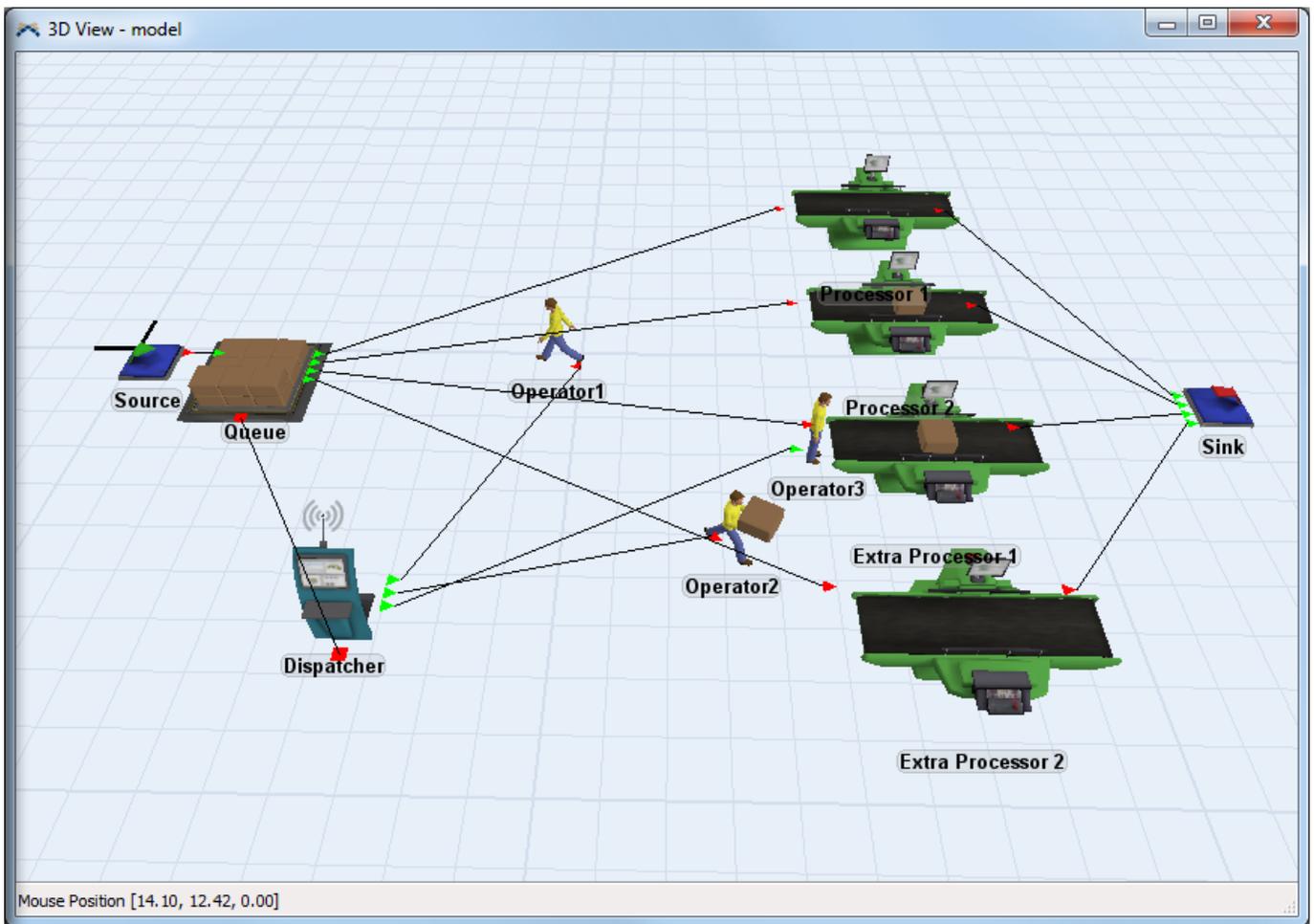
- Clique no botão de edição de código  e escreva o seguinte código:

```
closeinput (extraProc1);
closeinput (extraProc2);
setlabelnum (source, "arrivalTime", 10);
```

```
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 closeinput (extraProc1);
5 closeinput (extraProc2);
6
7 setlabelnum (source, "arrivalTime", 10);
```

Isto irá configurar o tempo entre chegadas (inter-arrival time) para 10 segundos, partes podem chegar mais lentamente. Os processadores extra irão fechar durante o tempo de lentidão.

- Pressione OK na janela **User Events** para fechar e aplicar as mudanças.
- Resete e rode seu modelo. Note que os operadores são recolocados para o mesmo local e alternam entre ocupados e tempos lentos.



Isto completa o tutorial User Events. Parabéns!

Tutorial de Time Tables

1. Introdução
2. Construção do modelo passo a passo.

Introdução

Neste tutorial nós iremos introduzir a você a ferramenta Time Table. A Time Table pode ser usada para especificar tempos quando um recurso fixo (Fixed Resource) ou um executor de tarefa (TaskExecutor) são programados para estar inativos. Isto pode ser atribuído em razão de quebra, manutenção, reparo e etc...do equipamento.

O que você irá aprender

- Como criar TimeTables and referenciar os membros.
- Como usar a TimeTable para especificar os tempos inoperantes para os seus equipamentos (processors) e operadores (operators)

Tempo aproximado para completar este exercício

Este exercício deve levar aproximadamente de 15-20 minutos para ser completado.

Overview do Modelo

Neste modelo nós teremos vários operadores realizando algumas tarefas. Uma TimeTable será criada para especificar quando os operadores estão inoperantes. Outra Time Table irá programar a manutenção para os equipamentos (processors).

Construção do Modelo passo a passo

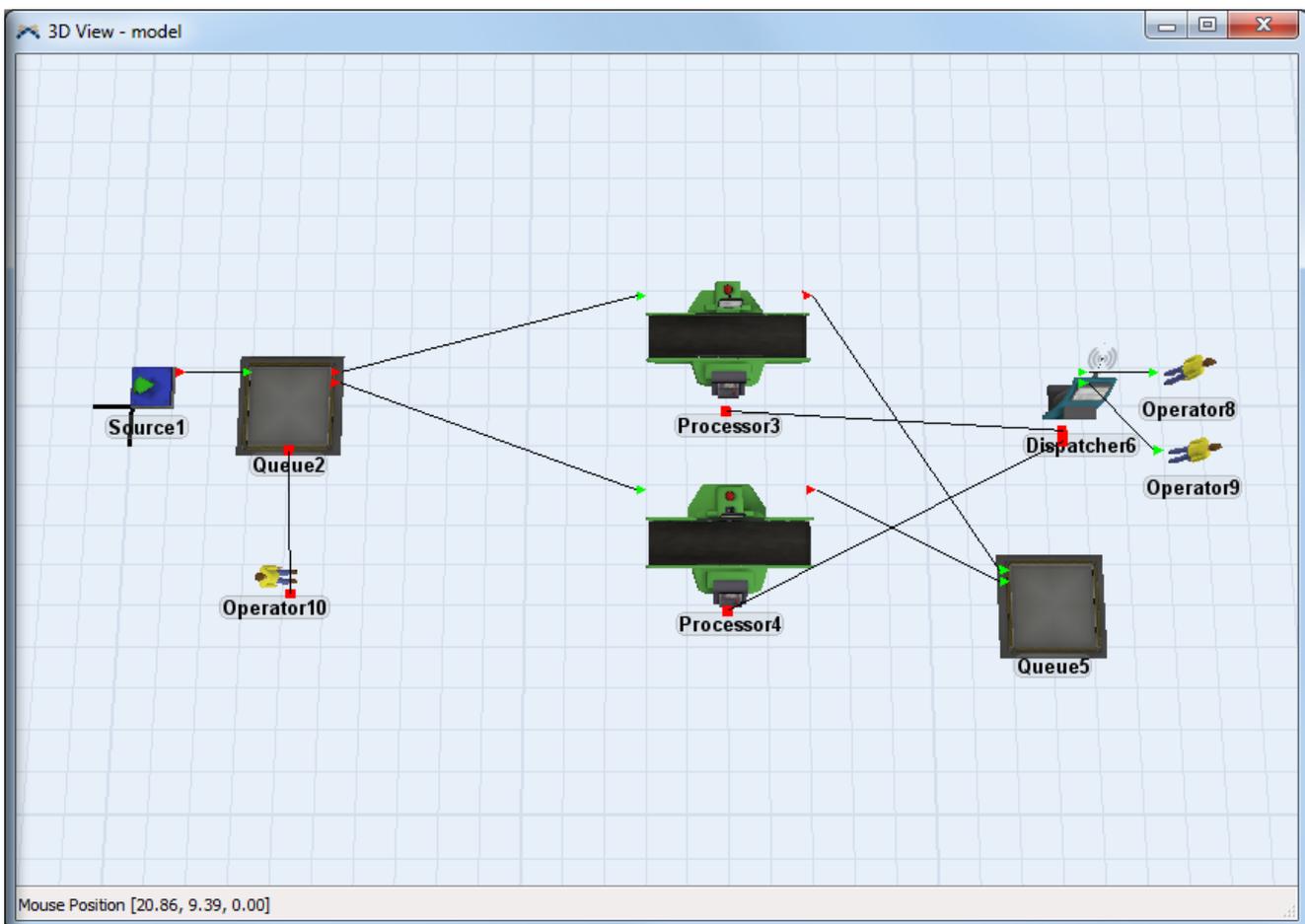
Construindo o Modelo de TimeTable

Inicie um novo modelo ao clicar no botão  na barra de ferramentas. Clique OK na janela de unidades do modelo (Model Units window), nós iremos usar as unidades padrões para nosso modelo.

Se a qualquer momento você encontrar dificuldades enquanto desenvolve este modelo, um tutorial completo e funcional pode ser encontrado em <http://www.flexsim.com/tutorials>

Etapa 1: Crie os objetos

- Arraste objetos de sua Library Icon Grid (biblioteca) para dentro da superfície de simulação 3D para criar um modelo conforme mostrado abaixo.



Conecte todos os objetos conforme mostrado abaixo:

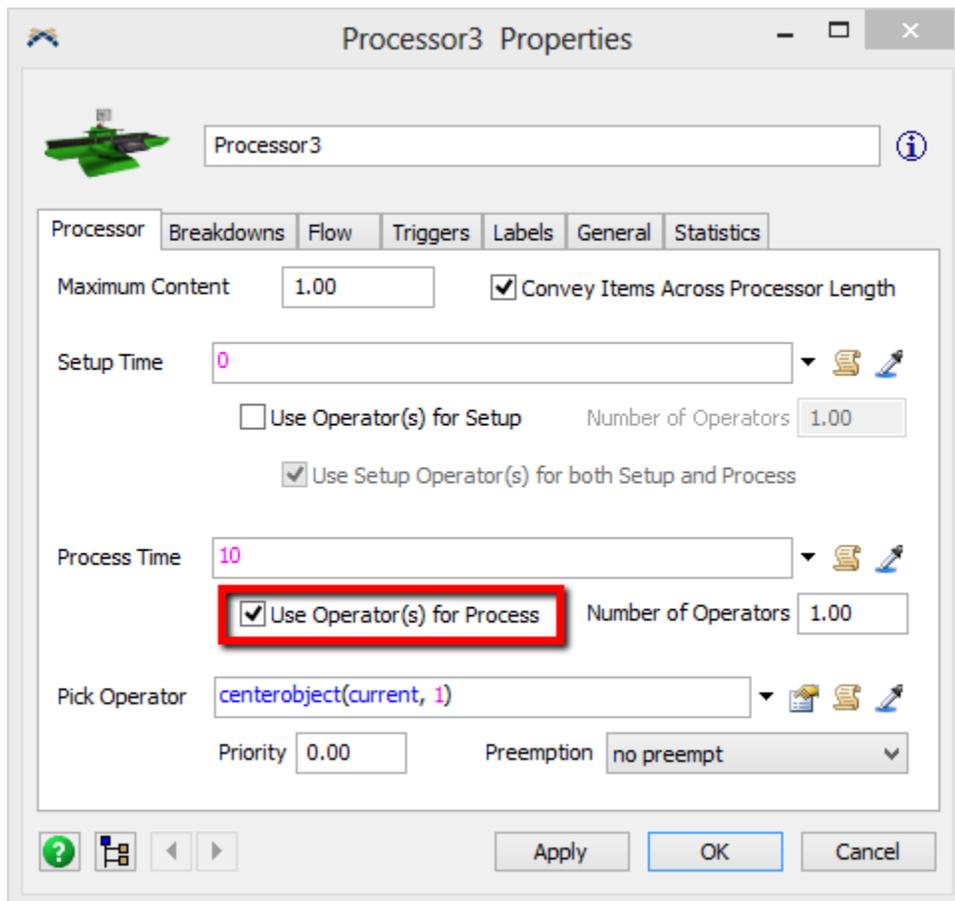
- Conecte o *Source1* ao *Queue2*.
- Conecte o *Queue2* ao *Processor3* e *Processor4*.
- Conecte o *Processor3* e *Processor4* ao *Queue5*.
- Conecte o *Dispatcher6* ao *Operator8* e *Operator9*.
- Conecte o *Queue2* ao *Operator10* com a conexão de porta central (tecla S).
- Conecte o *Processor3* e o *Processor4* ao *Dispatcher6* com a conexão de porta central (tecla S).

Etapa 2: Configure o Queue e os Equipamentos (Processors)

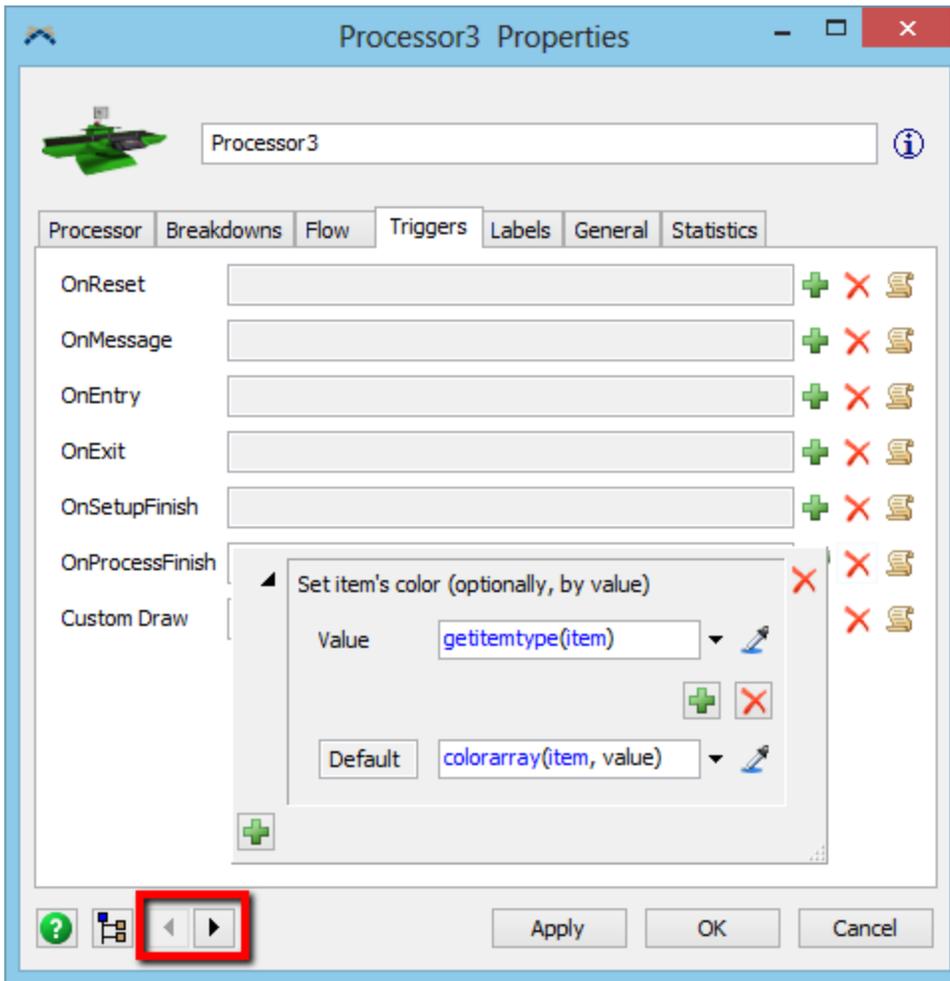
Um operador irá transportar os flowitems da fila (Queue2) para os dois equipamentos (processors). Os outros dois operadores serão usados para processar flowitems nos dois equipamentos (processors) e transportar os flowitems do equipamento (processor) para a fila (Queue5).

- Clique sobre a fila (*Queue2*) para abrir suas propriedades na janela Quick Properties.
- Na seção **Flow**, marque **Use Transport** e deixe-o usando a opção centerobject padrão.

- Abra a janela de propriedades do *Processor3*.
- Na guia **Processor**, marque **Use Operator(s) for Process** e deixe-o usando a opção centerobject padrão



- Vá para a guia **Flow**
- Marque **Use Transport** e deixe-o usando a opção centerobject padrão
- Acesse a guia **Triggers**
- Clique sobre o botão **+** na trigger **OnProcessFinish**
- Selecione a opção *Set Color* e deixe-o com as opções padrões



Nota: Você pode alternar rapidamente entre os equipamentos (processors) (ou outros similares objetos) ao clicar nas setas a esquerda ou a direita na parte inferior da janela de propriedades.

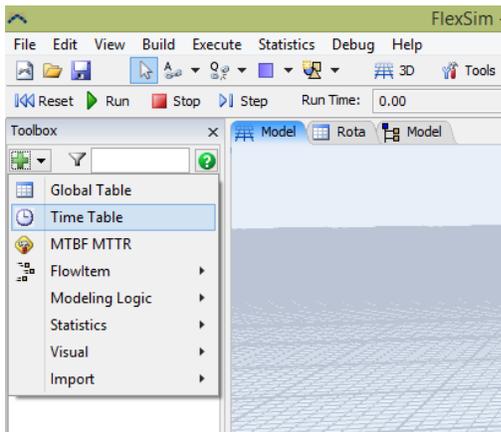
- Repita as últimas etapas de configuração para o Processor4
- Em **OK** para aplicar as mudanças e feche a janela de propriedades.

Resete e rode o modelo para assegurar-se que os operadores estão levando as caixas do Queue2 para os equipamentos (processors), processando as caixas e então levando elas para o Queue5. As caixas devem também alterar de cor após o processamento.

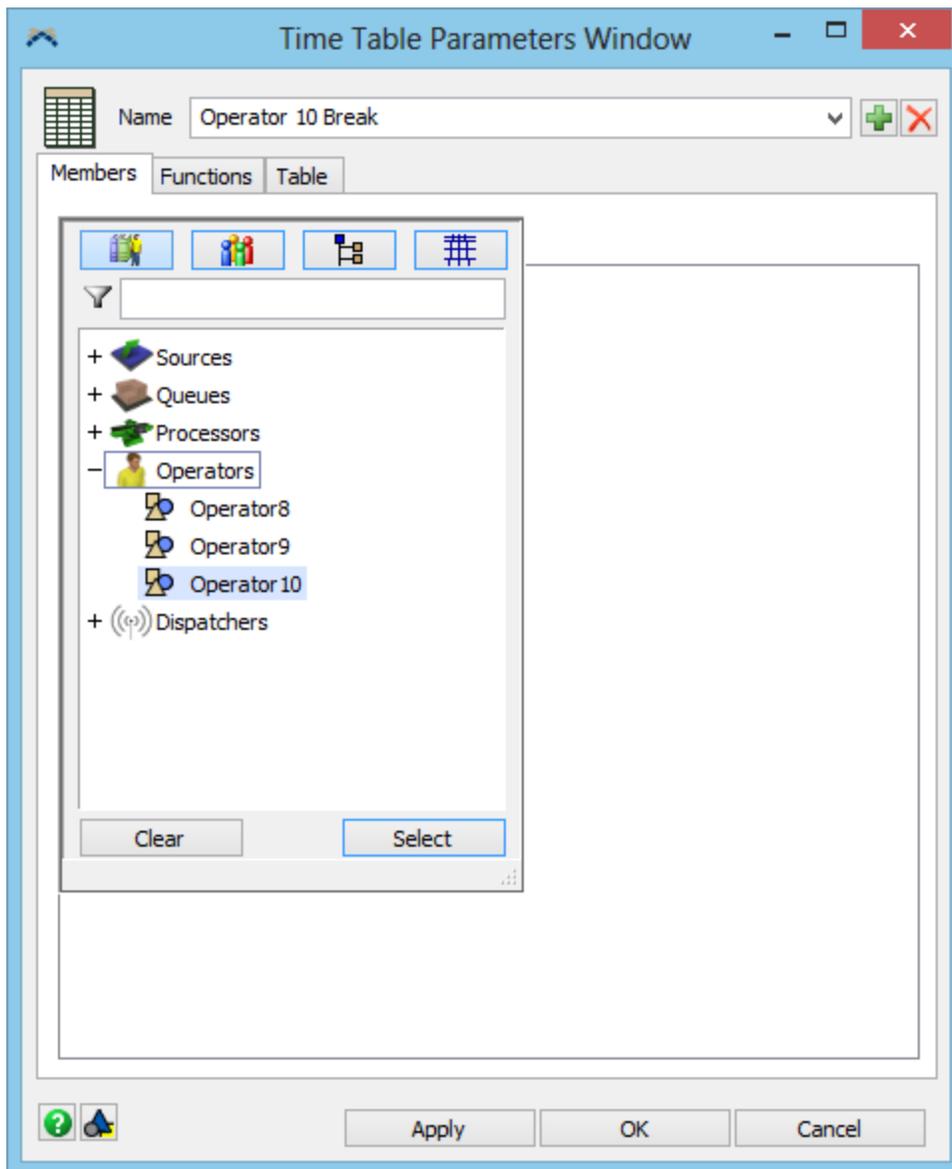
Etapa 3: Crie uma Time Table

Nós iremos agora criar uma Time Table para o *Operator10*.

- Abra a janela da **TimeTable** conforme tela abaixo.



- Renomeie a **TimeTable** para o *Operator 10 como Break*.
- Vá para a guia “Members” e clique no botão **+** para adicionar um membro. Destaque o *Operator10* e clique em **Select**.
- Vá para a guia Table e na linha 1 da tabela, configure o **Time** para 200, configure **State** para 12, e configure **Duration** para 30.
- Configure o **Repeat** para Custom e altere o valor para 200. Isto irá fazer com que o operador pare a cada 200 segundos.
- Vá para a guia Functions e na pick list para **Down Function** selecione Travel to Location, Delay Until Down Time Complete. Altere somente o **Location** para 2, -8, 0. Estas são as coordenadas de x, y, e z, respectivamente, que o operador irá enquanto estiver inoperante.



- Na pick list para **Resume Function** selecione *Do Nothing*.
- Clique em **OK** para aplicar e feche a janela TimeTable.

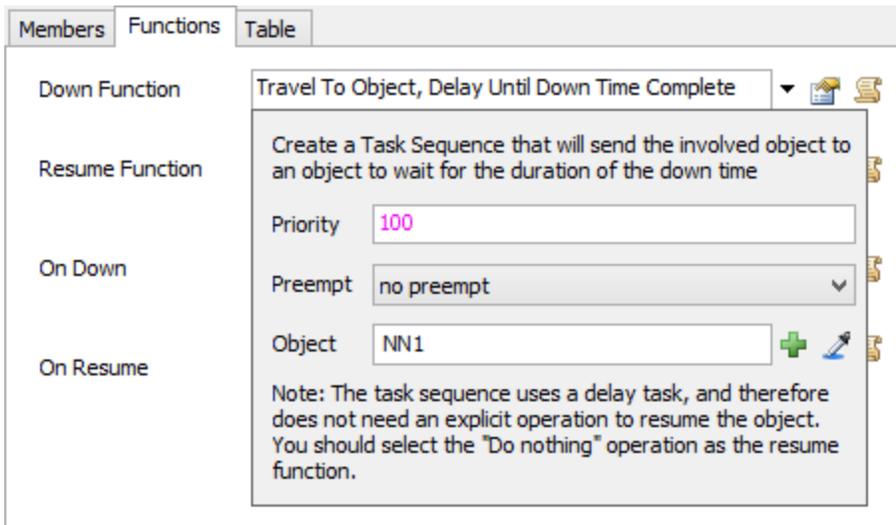
Resete e rode seu modelo. Quando você roda o modelo, a cada 200 segundos você verá que o operador se afasta da área de trabalho por 30 segundos e então retorna ao trabalho.

Nota: Se o operador está no meio da tarefa, ele irá completar a tarefa antes de iniciar seu descanso ou parada.

Etapa 4: Atualize a TimeTable

Nós podemos também fazer com que o operador pare em um objeto específico ao invés de configurar as coordenadas.

- Crie um Network Node e coloque-o distante de sua área de trabalho.
- Volte para a sua TimeTable do *Operator 10*.
- Na pick list para o **Down Function**, selecione *Travel to Object, Delay Until Down Time Complete*. Altere o **Destination Name** para NN1.



- Deixe o pick list para **Resume Function** como *Do Nothing*.
- Clique em **OK** para aplicar e feche a janela de propriedades.
- Resete e rode seu modelo e note que o operador irá para o *NN1*.

ATENÇÃO: A duração da parada do operador será o tempo que ele gasta em um determinado local. Se ele leva 10 segundos para caminhar até lá, ele irá parar por 30 segundos uma vez que chegou ao local. Adicionando 10 segundos para sua jornada de retorno, ele terá um total de 50 segundos entre a finalização de sua última tarefa e início de uma nova.

Etapa 5: Programando a Manutenção para o Processor.

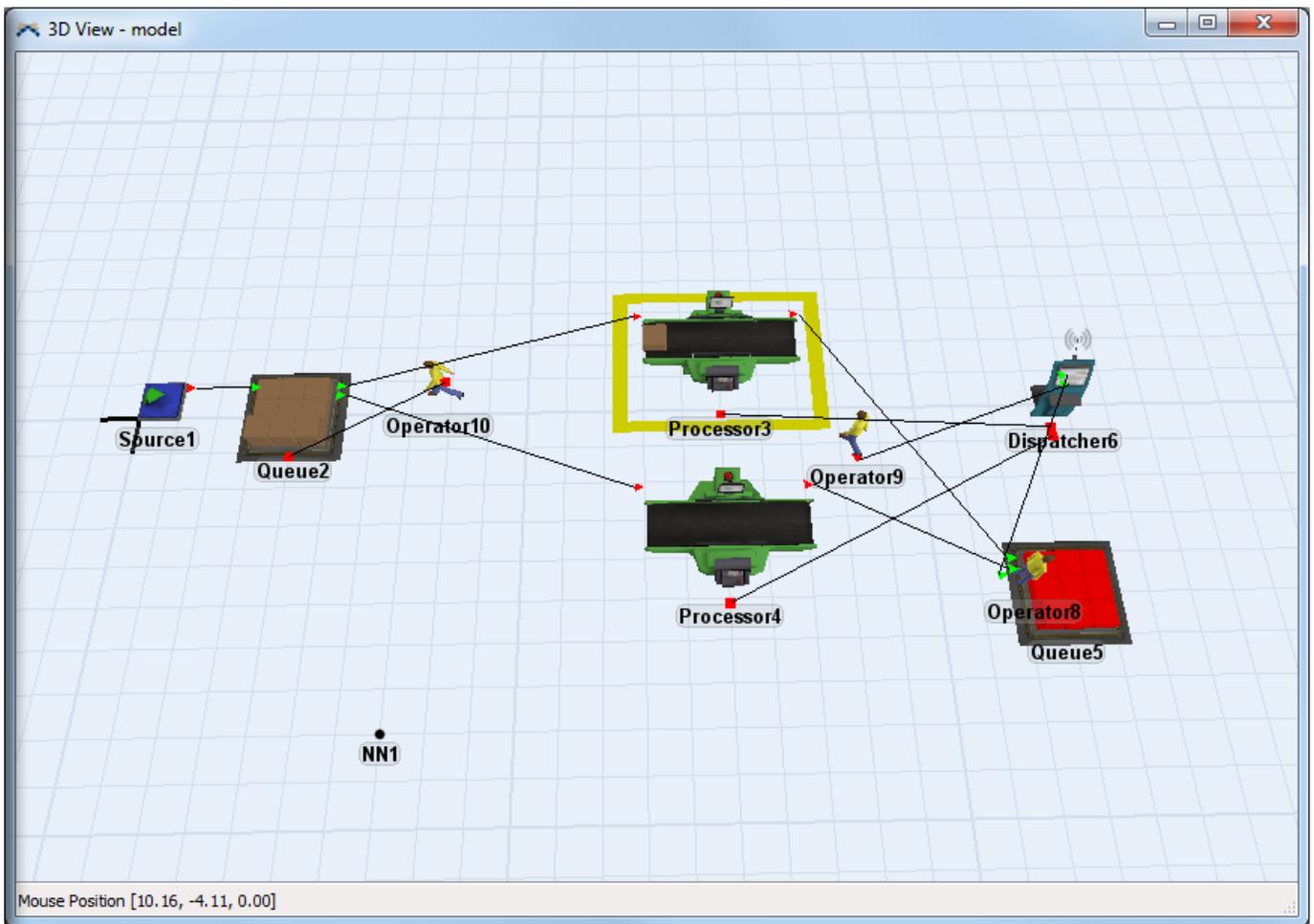
Próximo, nós iremos dar ao **Processor3** uma programação de parada para manutenção.

- Adicione outra **TimeTable**.
- Renomeie a TimeTable para *Processor Down Time* ou Parada programada do processor.
- Adicione o membro *Processor3*.
- Na guia Table, em Row 1 (Linha 1) da tabela, configure o **Time** como *200*, o **State** como *12* e configure o **Duration** para *100*.
- Configure o **Repeat Time** para Custom, *300*. Isto irá fazer com que o processor (equipamento) fique parado por 300 segundos após a manutenção inicial.
- Na pick list para **Down Function**, selecione *Stop Input*.
- Na pick list para **Resume Function**, selecione *Resume Input*.

Parando e colocando em funcionamento as portas de entrada dos equipamentos (objetos) significa que, os equipamentos irão continuar processando qualquer item que estiver dentro do equipamento naquele instante, mas não irá receber mais itens até que o tempo de parada seja cumprido. Se você deixar a TimeTable como Stop/Resume Object em sua configuração, os equipamentos irão parar com ou sem itens dentro dele e não irão retornar a atividade até que o tempo de inatividade se complete.

- Clique em **OK** para aplicar e feche a janela de TimeTable.

Resete e rode seu modelo. Seu modelo deve parecer-se com algo conforme descrito Guiaixo:



Isto completa o tutorial sobre Time Table. Parabéns!

Tutorial sobre Kinematics

1. Introdução
2. Construção do modelo passo a passo

Introdução

Este tutorial irá ensinar como usar o Kinematics em seu modelo. Kinematics permite você realizar movimentos simultâneos com um objeto simples. Isto é melhor visualizado ao olhar a forma como o objeto Crane se movimenta.

O que você irá aprender

- Como criar e adicionar o Kinematics para um objeto fixo (fixed resource).
- Como movimentar um objeto usando o Kinematics.

Tempo aproximado para completar este exercício.

Este exercício deve levar de 30-45 minutos para ser completo.

Overview do Modelo

Neste modelo nós iremos tratar o processor como se ele fosse uma centrífuga e ele irá trepidar no momento em que processa flowitems. [Clique aqui](#) para um tutorial passo a passo.

Construção do Modelo passo a passo

Construindo o modelo do Kinematics

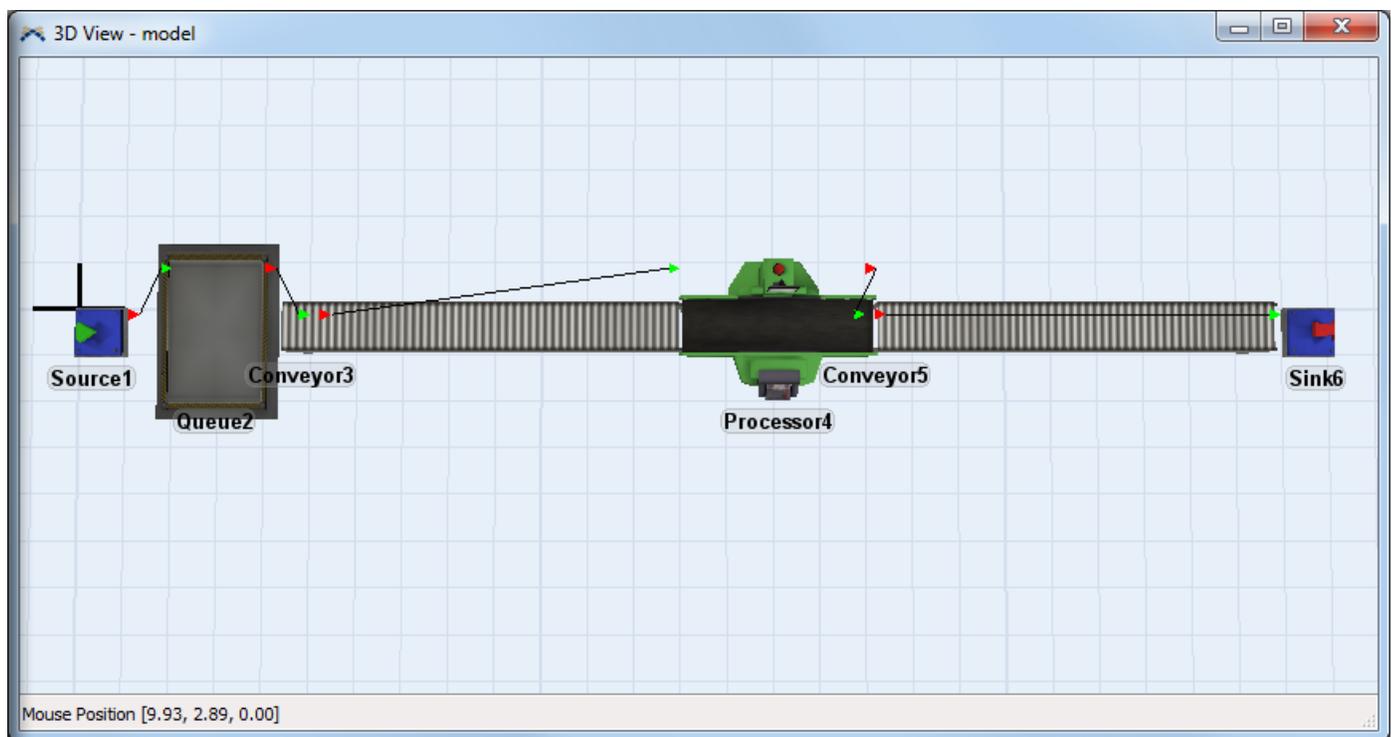
Inicie um novo modelo ao clicar no botão  na barra de ferramentas. Clique Ok na janela de unidades do modelo, nós iremos usar as unidades padrões para o nosso modelo.

Se a qualquer momento você encontrar dificuldades enquanto desenvolve este modelo, um tutorial funcional e completo pode ser encontrado em <http://www.flexsim.com/tutorials>

Parte 1: Kinematics Básico

Etapa 1: Crie os objetos

- Arraste os objetos da sua biblioteca (Library Icon Grid) para dentro da superfície de simulação 3D para criar o modelo mostrado abaixo.



Conecte todos os objetos conforme descrito abaixo:

- Conecte o *Source1* com o *Queue2*, o *Queue 2* com *Conveyor3*, o *Conveyor3* com o *Processor4*, o *Processor4* com o *Conveyor5* e finalmente, o *Conveyor5* com o *Sink6*

Deixe-os nas seguintes posições (x, y, z) – que pode ser configurado na guia General dos objetos:

Source: (-5, 1, 0)
Queue2: (-1, 1, 0)
Conveyor3: (3, 0, 0)
Processor4: (12, 1, 0)
Conveyor5: (17, 0, 0)
Sink: (29, 1, 0)

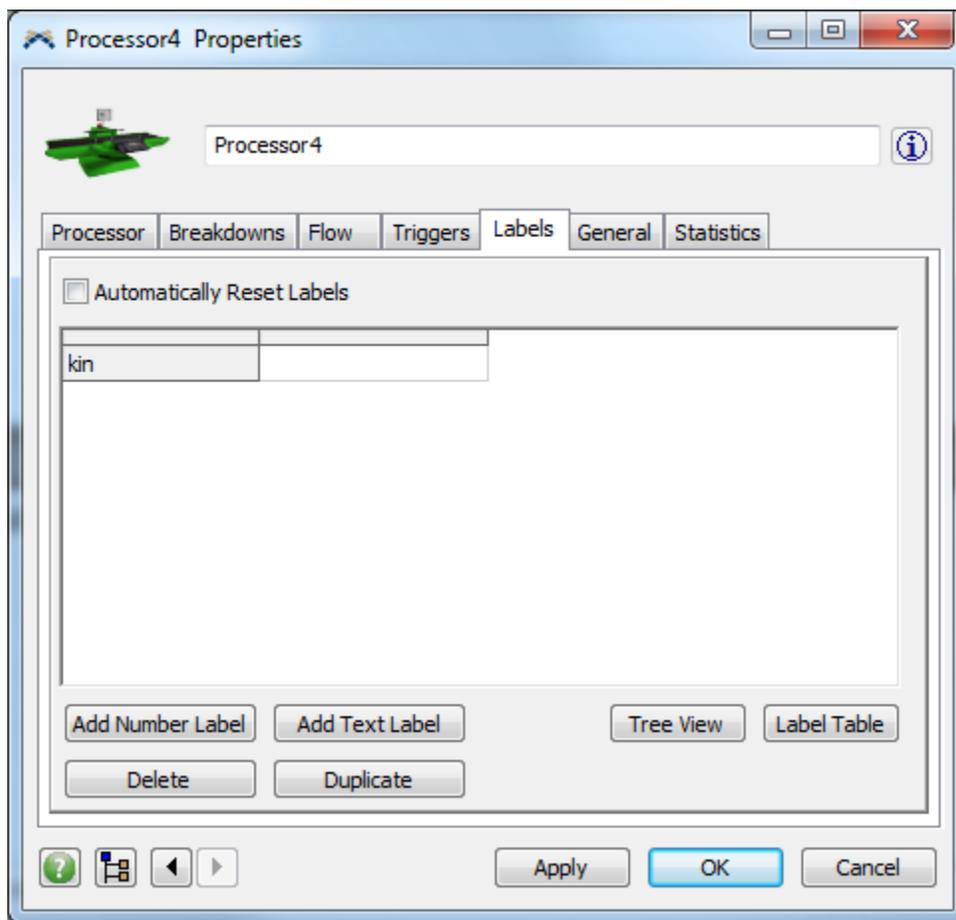
Etapa 2: Adicione as Labels do Kinematic

Todos os kinematics necessitam de um único node para a informação ser armazenada. A forma mais simples de fazer isso é criando uma text label dedicada para o kinematics.

Alternativamente, um node pode ser adicionado para as variáveis dos objetos, neste caso, todos os comandos label serão substituídos por *getvarnode*.

- Abra a janela de propriedades para o Processor.
- Vá para guia **Labels**.
- Adicione um text label ao clicar em **Add Text Label** e renomeie a label “kin”.

ATENÇÃO: Não tente deletar o node kinematics de dentro da janela de propriedades. Use as seguintes instruções para alterar os parâmetros de seu kinematic. Deletando-o manualmente pode danificar o programa.

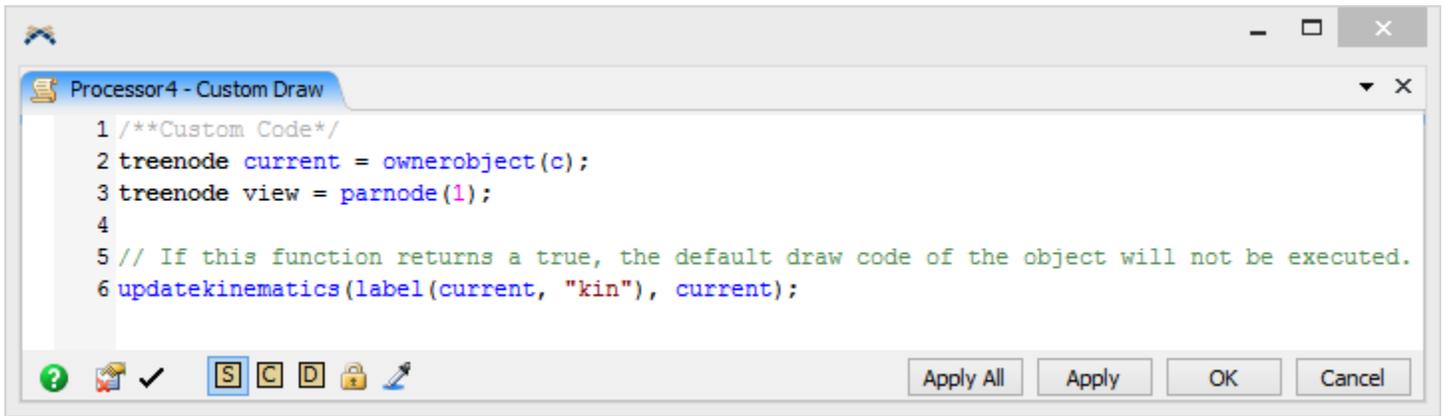


Etapa 3: Adicione Custom Draw Code

O kinematics necessitará ser atualizado continuamente enquanto o modelo roda.

- Vá para a guia **Triggers**
- Clique no botão de edição de código  próximo a trigger **Custom Draw**
- Entre com a seguinte linha de código:

```
updatekinematics(label(current, "kin"), current);
```



- Clique em **OK** para aplicar e feche a janela de edição de códigos.

Etapa 4: Atualize o Kinematics quando o processo termina

O kinematics necessitará de uma atualização final no momento em que estiver completo. Isto irá assegurar que, independente da taxa de frame (framerate) de sua visualização 3D (o qual define a frequência com que a trigger Custom Draw é acionada), quando você tiver uma visualização em 3D aberta, o kinematic irá provavelmente ser atualizado para a sua posição/rotação de estado final.

- Na guia **Triggers**, clique no botão de edição de código  próximo a trigger **OnProcessFinish**
- Copie o mesmo comando *updatekinematics* usado na trigger **Custom Draw**:

```
updatekinematics(label(current, "kin"), current);
```

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 updatekinematics(label(current, "kin"), current);
```

- Clique em **OK** para aplicar e feche a janela de edição de códigos.

Etapa 5: Adicione o código OnReset

Faz-se necessário que o objeto retorne para sua posição original quando você resetar o modelo.

- Na guia **Triggers**, clique no botão de edição do código  próximo a trigger **OnReset**.
- Use o comando *initkinematics* com o node sendo a text label que você criou recentemente. Acesse a guia **General** da janela de propriedades e verifique a posição e rotação do objeto em x, y, e z. Configure os correspondentes valores no comando *initkinematics*. (Os valores de rotação para o seu equipamento (processor) será 0 por padrão). Configure os dois últimos parâmetros como 0.

```
initkinematics(label(current, "kin"), 12, 1, 0, 0, 0, 0, 0, 0);
```

Nota: Os dois últimos parâmetros indicará a gestão da rotação e coordenadas de localização. Quando o rotation management (gestão de rotação) é configurado como 1, seu objeto irá rotacionar de forma que a frente (a direção x da posição) dele fique virado para a direção da viagem. Se as coordenadas de localização estiverem configuradas como 1, ele usará as coordenadas dos objeto contenedor ao invés do modelo propriamente dito.

```
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 initkinematics(label(current, "kin"), 12, 1, 0,0,0,0,0,0);
```

- Clique em **OK** para aplicar e fechar a janela de edição do código.

Etapa 6: Atualize o Kinematics quando o setup finaliza

Você vai querer que o seu objeto retorne a sua posição de estado inicial no início do kinematic. Nós iremos agora adicionar a informação kinematic que irá movimentar o objeto.

- Na guia **Triggers**, clique no botão de edição do código  próximo a trigger **OnSetupFinish**.
- Copie o comando *initkinematics* com os mesmos parâmetros da trigger **OnReset**
- Entre com a seguinte linha de código adicional, para que a tela de Custom Code da trigger **OnSetupFinish** se pareça conforme a tela abaixo:

```
addkinematic(label(current, "kin"), 0, 0, 3240, 360, 90, 180, 0, 0, time(), 2);
```

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 initkinematics(label(current, "kin"), 12, 1, 0,0,0,0,0,0);
6 addkinematic(label(current, "kin"), 0, 0, 3240,360, 90, 180, 0, 0, time(), 2);
```

O comando *addkinematic* configura os parâmetros x, y, e z, para 0, 0, e 3240, respectivamente. Isto será a movimentação de rotação para então rotacionar ao redor do eixo z em 3240 graus (9 vezes). A meta de velocidade (ou velocidade máxima) será configurada para 360 graus/segundo, com uma aceleração de 90 graus/segundo/segundo. A velocidade inicial e final será 0. A velocidade inicial será o tempo que o comando é chamado, então nós usamos o comando *time()*. Por último, nós queremos que isso seja a movimentação de rotação, então nós configuramos o parâmetro para 2 ou KINEMATIC_ROTATE (para movimentação de translação nós configuraremos isto para 1 ou KINEMATIC_TRAVEL).

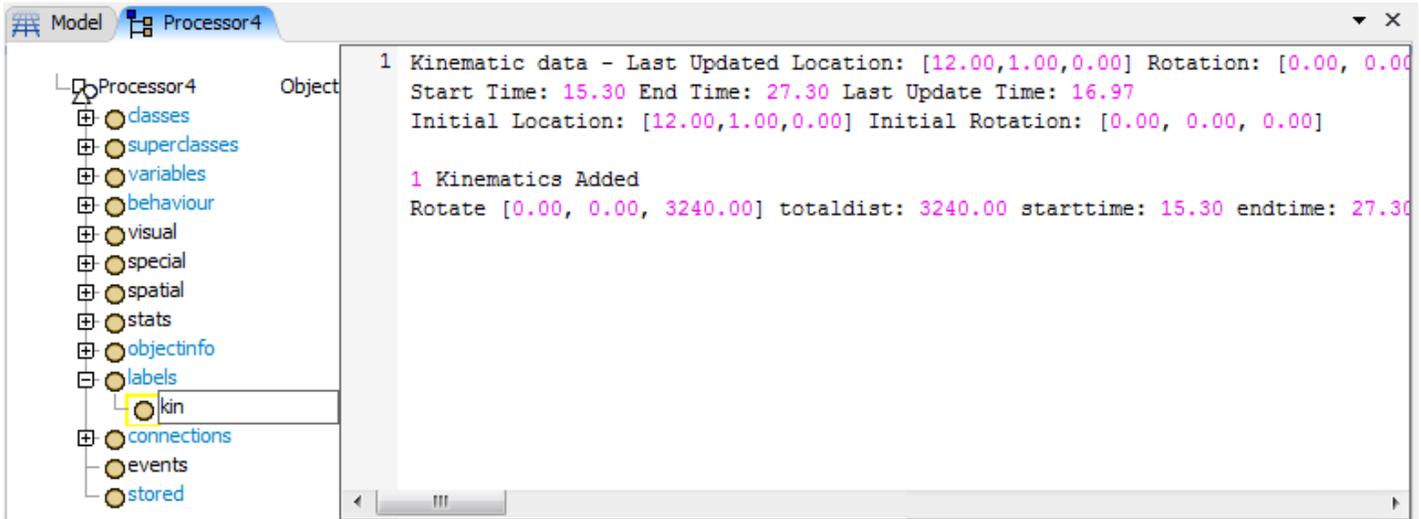
- Clique em **OK** para aplicar e fechar a janela de propriedades.

Resete e rode seu modelo e você deve ver o processor acelerar, rotacionar e desacelerar para então parar. Você pode verificar que se o próximo flowitem começar a ser processado antes do seu kinematics ser feito, ele vai instantaneamente redefinir sua posição para coincidir com os parâmetros *initkinematics*. Nossa próxima etapa será coincidir o tempo de processamento com o tempo que leva para o seu kinematic finalizar.

Etapa 7: Veja as informações do Kinematic

Informação importante sobre o seu kinematic serão armazenados na label "kin".

- Rode o modelo até o flowitem entrar no processor e pare o modelo, NÃO RESETE.
- Clique com o botão direito sobre o *Processor* e clique sobre o botão **Explore Tree**.
- Expanda a árvore do processor, então expanda o node das "labels" dentro da árvore. Clique sobre a label "kin" e você verá as informações para o kinematic conforme a figura abaixo:



- Arraste a barra para direita até que você possa visualizar o starttime (tempo inicial) e endtime (tempo final). A diferença entre os dois é o tempo que leva para o kinematic ser completado. Você pode ver que a diferença é de 12 segundos.
- Abra a janela de propriedades do Processor.
- Na guia **Processor**, altere o tempo de processamento (**Process Time**) para 12.
- Opcional: Adicionando um **Setup Time** de um ou dois segundos, permitirá nosso item se movimentar para fora do processor antes do processor começar a girar novamente.

Quando você rodar o modelo, você deverá ver o equipamento finalizando o processamento quando a rotação terminar. O kinematics permite a você configurar sua simulação em termos de velocidade, aceleração do seu equipamento assim como dar a você a parte visual.

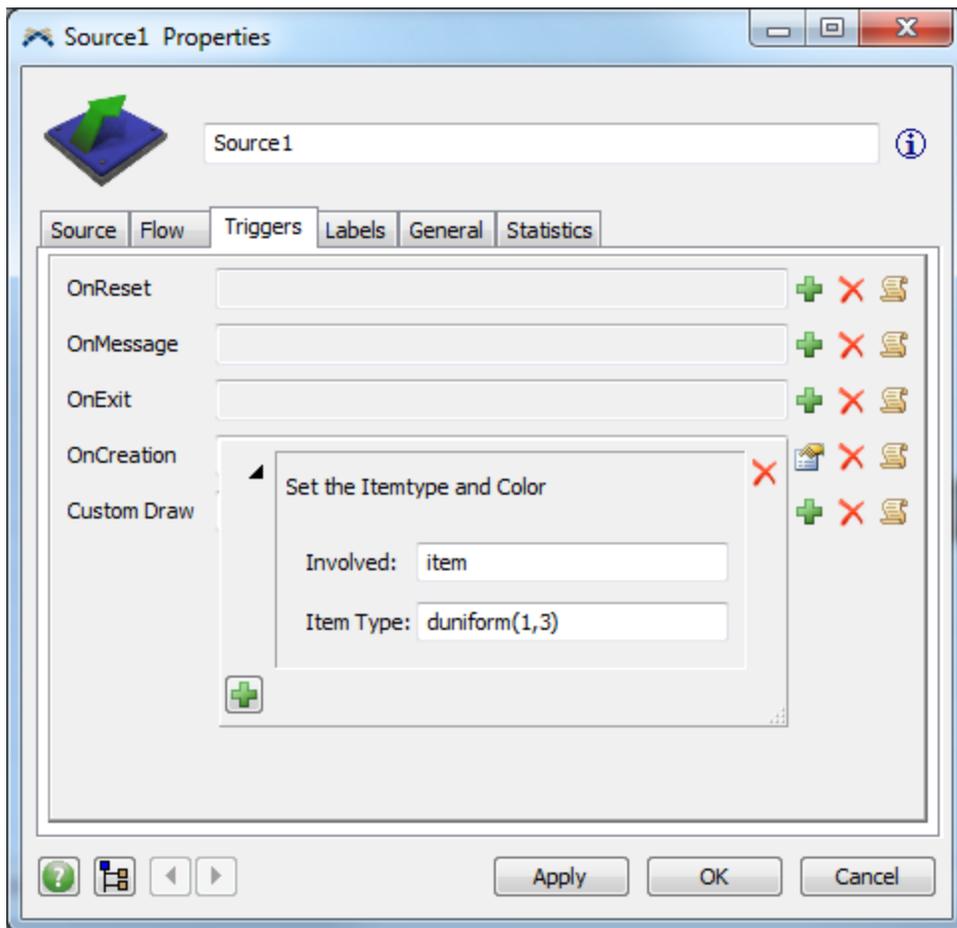
Kinematics pode ser usado para multiplicar movimentos simultâneos para um objeto simples. Tente adicionar mais comandos *addkinematics* para a trigger **OnSetupFinish**.

Parte 2: Atualize o Kinematics dinamicamente

Nesta parte do tutorial, nós teremos a rotação do processor em diferentes velocidades de acordo com o tipo de item.

Etapa 8: Crie tipos de itens múltiplos (Multiple Itemtypes)

- Abra a janela de propriedades do Source.
- Vá para a guia **Triggers**
- Clique no botão **+** próximo a trigger **OnCreation**.
- Selecione a opção *Set Itemtype and Color* da pick list
- Deixe os valores como padrão.

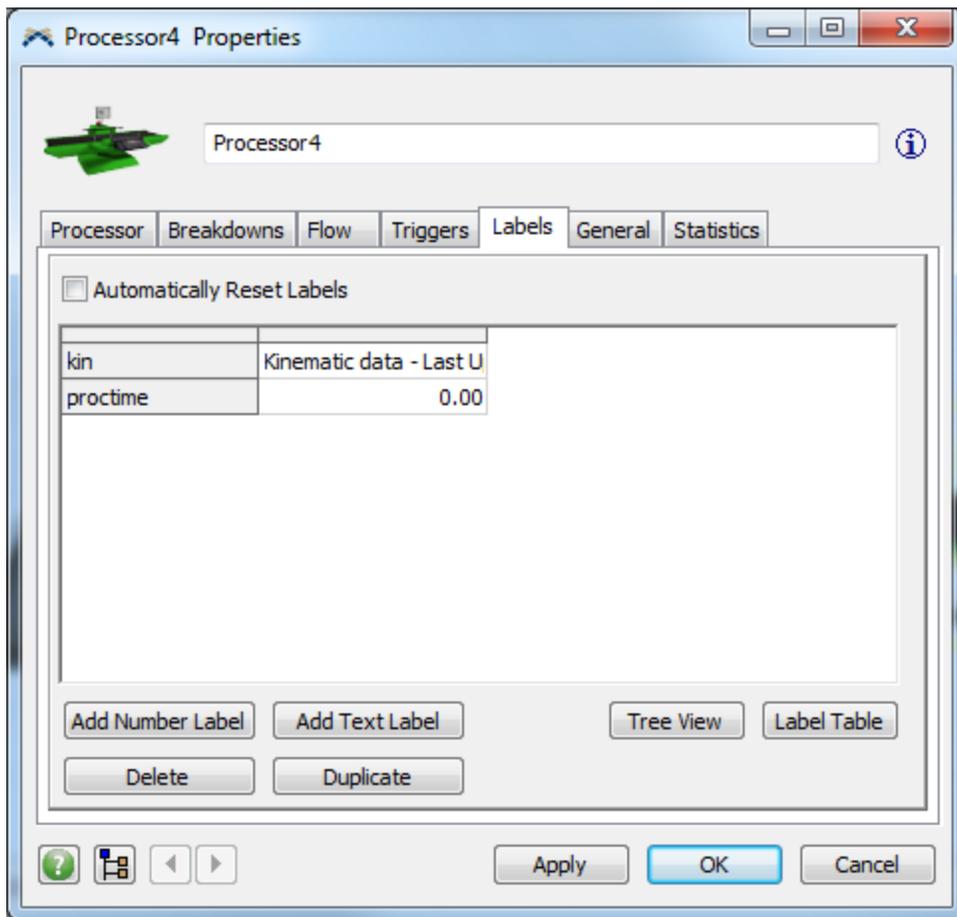


- Clique em **OK** para aplicar e fechar a janela de propriedades

Etapa 9: Processe Cada tipo de item (Itemtype) diferentemente

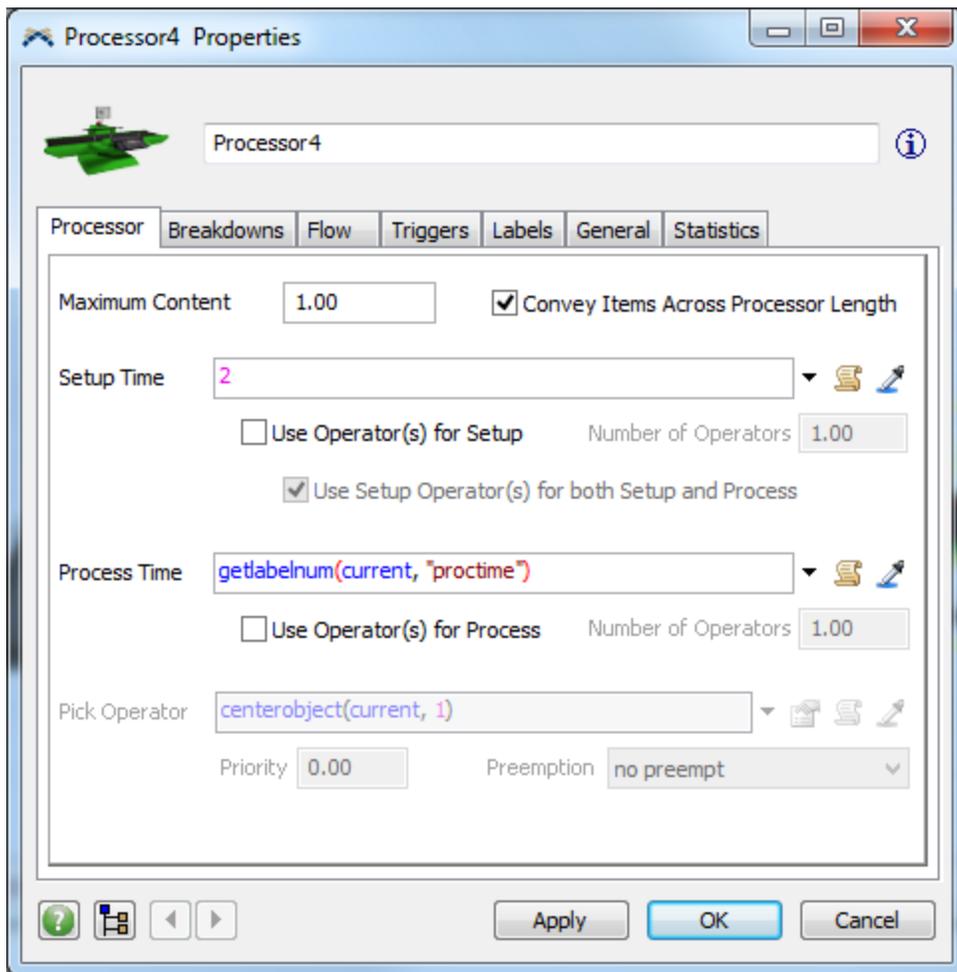
Cada tipo de item (item type) será processado a diferentes velocidades e além do mais requer diferentes tempos de processamento. Visto que isto irá alterar dinamicamente enquanto o modelo roda, nós iremos criar uma number label para manter o rastreamento da informação.

- Abra a janela de propriedades do Processor.
- Vá para a guia **Labels**
- Clique em **Add Number Label** e renomeie a label como *“proctime”*.



- Vá para a guia **Processor**
- Entre com o seguinte comando no campo **Process Time**:

```
getlabelnum(current, "proctime")
```



Etapa 10: Crie os Kinematics Dinâmicos

Nós precisamos customizar o kinematics para que ele mude para cada tipo de item (item type).

- Vá para guia **Triggers**.
- Clique no botão de edição de código  e acesse a trigger **OnSetupFinish**
- Entre com o seguinte código:

```
initkinematics(label(current, "kin"), 13, 1, 0, 0, 0, 0, 0, 0);
int z;
int speed;

int type = getitemtype(item);
switch (type)
{
    case 1:
        z = 1080;
        speed = 360;
        break;
```

```

    case 2:
    z = 3240;
    speed = 360;
    break;

    case 3:
    z = 3240;
    speed = 180;
    break;
}

```

```
addkinematic(label(current, "kin"), 0, 0, z, speed, 90, 180, 0, 0, time(), 2);
```

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 initkinematics(label(current, "kin"), 13, 1, 0, 0, 0, 0, 0, 0);
6
7 int z;
8 int speed;
9
10 int type = getitemtype(item);
11
12 switch (type)
13 {
14     case 1:
15         z = 1080;
16         speed = 360;
17         break;
18
19     case 2:
20         z = 3240;
21         speed = 360;
22         break;
23     |
24     case 3:
25         z = 3240;
26         speed = 180;
27         break;
28 }
29
30 addkinematic(label(current, "kin"), 0, 0, z, speed, 90, 180, 0, 0, time(), 2);

```

O comando *initkinematics* deve ser o mesmo que o da Parte 1. O comando *addkinematics* é colocado dentro de uma referência “e se” que verifica o tipo de item para cada item conforme o setup é finalizado. Tente brincar com os valores de rotação de z assim como com os valores de velocidade máxima, aceleração e desaceleração (maxspeed, acceleration e deceleration) do equipamento (processor).

Etapa 11: Atualize o tempo de processamento

Nós iremos agora conseguir a informação kinematic do nosso objeto e usá-lo para alterar o tempo de processamento do equipamento (processor).

- Ainda falando sobre o código customizado (custom code) para a trigger **OnSetupFinish** adicione o seguinte código:

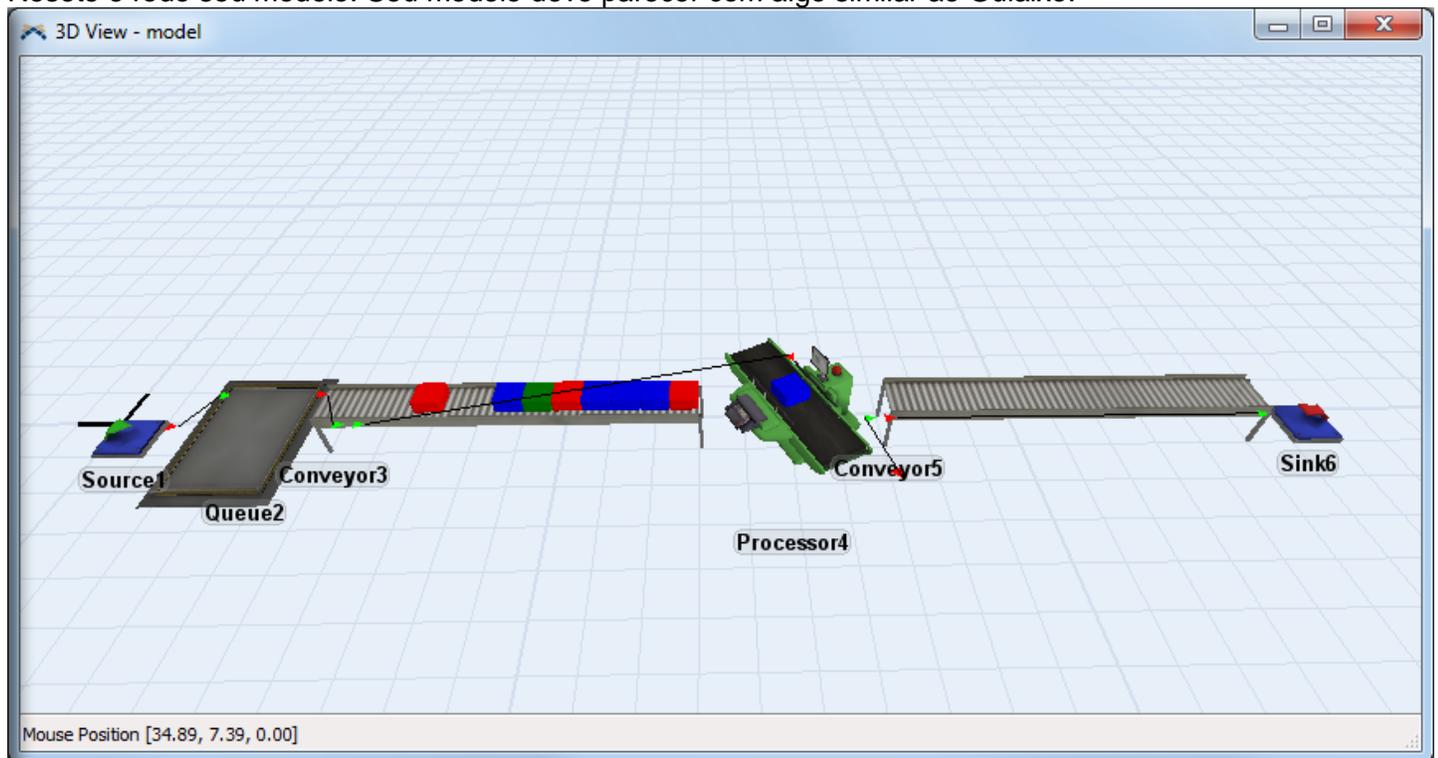
```
double endtime = getkinematics(label(current, "kin"), KINEMATIC_ENDTIME);  
double starttime = getkinematics(label(current, "kin"), KINEMATIC_STARTTIME);  
double proctime = endtime - starttime;  
setlabelnum(current, "proctime", proctime);
```

Se você tiver mais de um kinematic de cada vez, você pode configurar qual kinematic está puxando a informação e de onde, e ainda conseguir informações a distâncias específicas ou tempos de viagem no meio da ação kinematic. Veja o manual para mais detalhes.

O comando *getkinematics* pega a informação da última atualização realizada para o kinematic. A informação disponível que pode ser pega está descrita na seção Kinematics do manual de ajuda.

- Clique **Ok** para aplicar e fechar a janela de edição do código.
- Clique **OK** para aplicar e fechar a janela de propriedades do objeto.

Resete e rode seu modelo. Seu modelo deve parecer com algo similar ao Guiaixo:



Isto completa o tutorial sobre o Kinematics. Parabéns!

Sequência de Tarefas - Exercício 1

1. Introdução
2. Construção do modelo passo a passo

Introdução

Neste tutorial, você vai aprender a construir uma sequência de tarefas básicas a partir do zero. O operador vai pegar o flowitem de um queue, levá-lo a uma mesa para inspecionar o item, em seguida, pegar o item e levar para um processador. Escrevendo sua própria sequência de tarefas permite que você aloque e use um operador para toda a tarefa. Este tutorial pressupõe um conhecimento sólido de interação básica com o software, e vai pedir para você escrever algumas linhas de programação.

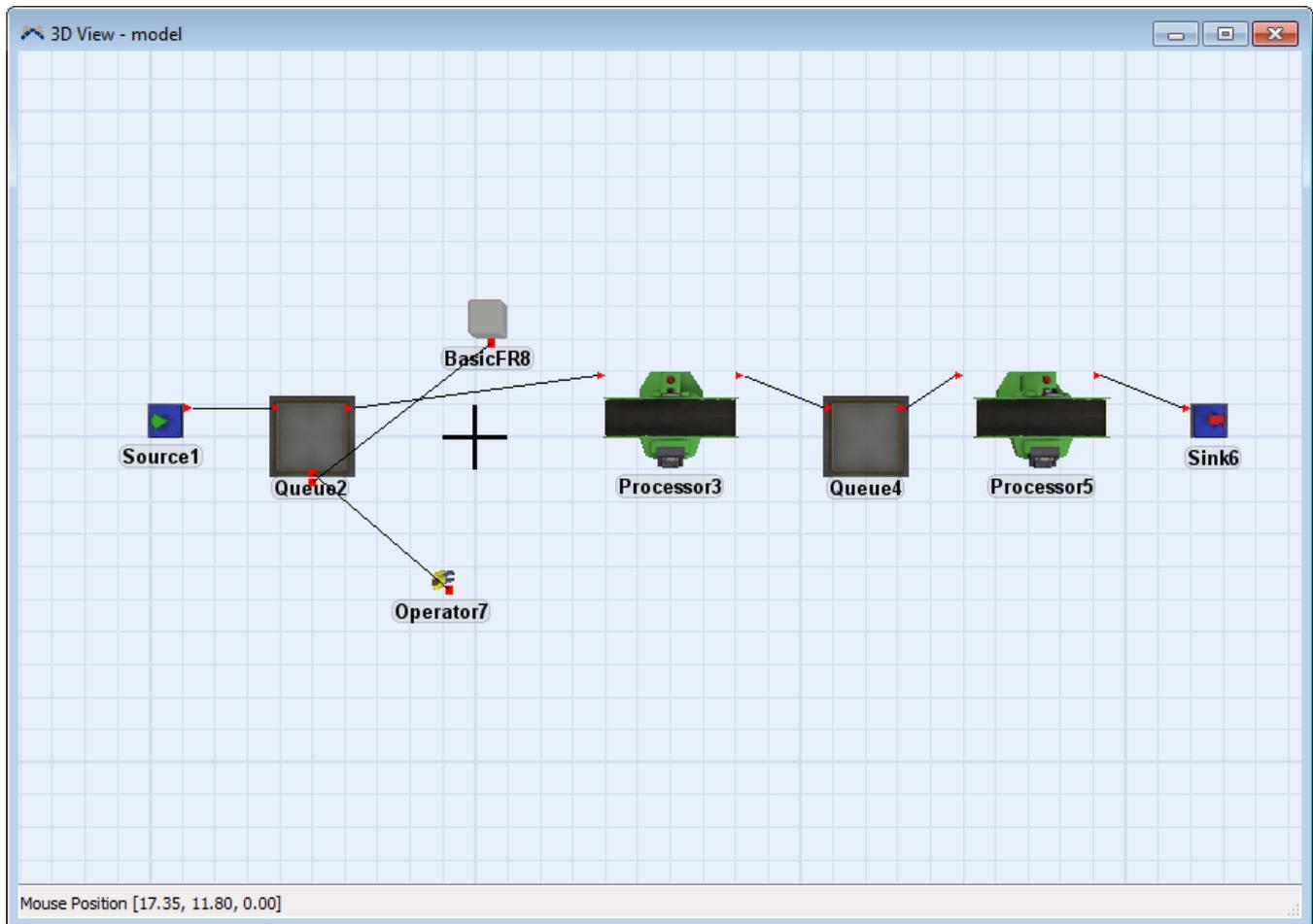
Se a qualquer momento você encontrar alguma dificuldade durante a construção deste modelo, um tutorial completo e funcional pode ser encontrado em <http://www.flexsim.com/tutorials>

Para mais informações sobre TaskType e seus parâmetros, veja a página sobre Task Types.

Construção do tutorial passo a passo

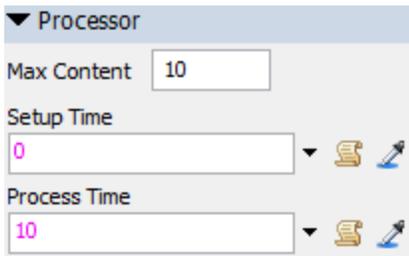
Etapa 1: Configure o modelo

- Crie um **Source**, um **Queue**, um **BasicFR**, um **Processor**, um **Queue**, um **Processor**, um **Operator** e um **Sink**, e os posicione conforme a figura mostrado abaixo. O BasicFR atuará como a mesa de teste, mas não terá nenhuma lógica aplicada nele. Ele está ali apenas para dar ao operador um lugar para viajar. Um Visual Tool poderá também ser usada, ou qualquer outro Fixed Resources.
- Conecte os objetos como na figura mostrado abaixo, esteja seguro de conectar o Operador e o BasicFR na porta central do primeiro Queue nesta ordem

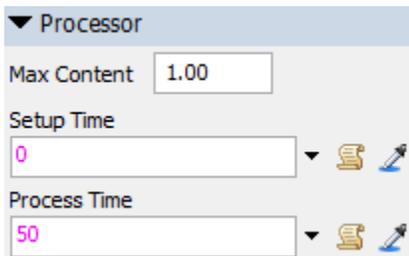


Etapa 2: Edite os Objetos

- Clique sobre o primeiro Processor para abrir suas propriedades na janela Quick Properties.
- Na seção **Processor**, altere a capacidade máxima (**Max Content**) para **10**.



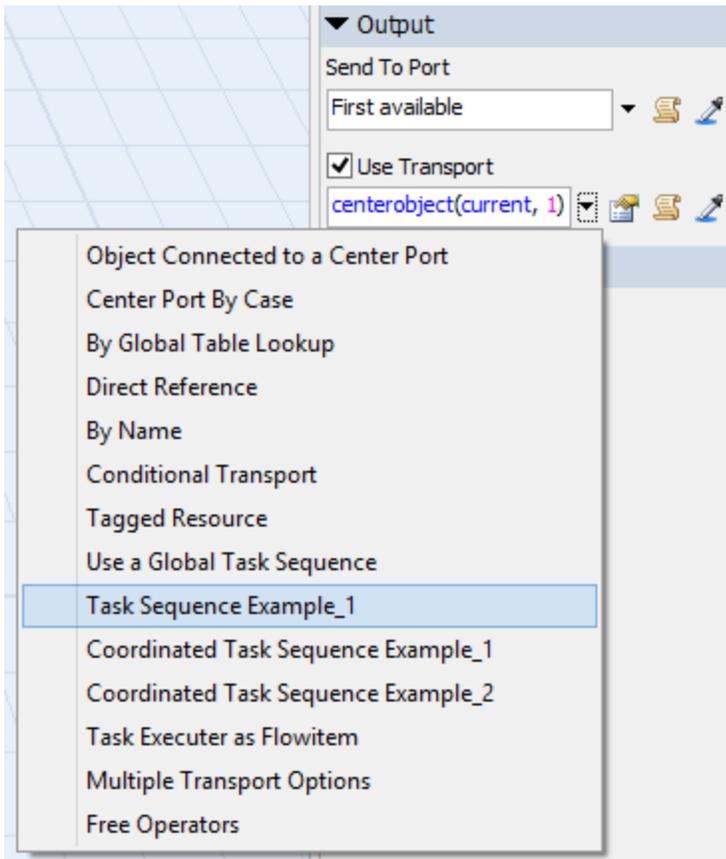
- Clique sobre o primeiro Processor para abrir suas propriedades na janela Quick Properties.
- Na seção **Processor**, altere o tempo de processamento **Process Time** para **50**.



Etapa 3: Escreva a sequência de tarefas (Task Sequence)

Para facilitar as coisas, você vai usar o exemplo básico Task Sequence para iniciar a sequência de tarefas. De lá, você pode alterar e adicionar o que quiser para atender às suas necessidades.

- Dê um duplo clique no primeiro Queue para abrir sua Properties window, então clique na guia Flow. Na guia **Flow**, selecione a caixa **Use Transport**, e selecione **Task Sequence Example_1** da listagem. Este exemplo de task sequence por padrão fornece a mesma funcionalidade referenciada para um operador. O Operador viaja para o objeto atual, pega o item, viaja para o objeto subsequente e descarrega o item. Iremos alterar apenas alguns detalhes.



- Do lado direito da listagem de opções **clique no botão Code Edit**  para abrir o editor de códigos. Neste modelo, o Operador deve fazer a seguinte tarefa, e nada mais. Então, você irá escrever os códigos de forma que se pareça muito com os códigos das linhas da figura abaixo, não importando neste momento os números das linhas. As demais linhas não precisam ser escritas nesta altura. Não se atente ao fato do número da linha em seu modelo ficar exatamente como no guia abaixo. A sequência de numeração da linha poderá ser diferente neste momento

```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4 /**Task Sequence Example 1*/
5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecuter, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this fiel
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executer
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
18
19 inserttask(ts,TASKTYPE_TRAVEL,current,NULL);
20 inserttask(ts,TASKTYPE_FRLOAD,item,current,port);
21
22 inserttask(ts,TASKTYPE_TRAVEL,outobject(current,port),NULL);
23 inserttask(ts,TASKTYPE_FRUNLOAD,item,outobject(current,port),opipno(current,port));
24
25 dispatchtasksequence(ts);
26 // return a 0 so this object will know that you made your own tasksequence and it doesn't n
27 //to make the standard tasksequence automatically
28 return 0;

```

Após um operador pegar o item, nós queremos que ele viaje até um BasicFR, e aguarde por 10 segundos antes de viajar para o Processor seguinte.

- Agora na linha vazia 21, digite o seguinte:
`inserttask(ts, TASKTYPE_TRAVEL, centerobject(current,2), NULL);`
 Pressione a tecla **Enter** para ir para a próxima linha.

```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4 /**Task Sequence Example 1*/
5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecuter, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this fiel
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executer
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
18
19 inserttask(ts,TASKTYPE_TRAVEL,current,NULL);
20 inserttask(ts,TASKTYPE_FRLOAD,item,current,port);
21 inserttask(ts,TASKTYPE_TRAVEL,centerobject(current,2),NULL);
22 |
23 inserttask(ts,TASKTYPE_TRAVEL,outobject(current,port),NULL);
24 inserttask(ts,TASKTYPE_FRUNLOAD,item,outobject(current,port),opipno(current,port));
25
26 dispatchtasksequence(ts);
27 // return a 0 so this object will know that you made your own tasksequence and it doesn't n
28 //to make the standard tasksequence automaticlv

```

- Na linha 22, digite o seguinte:
`inserttask(ts, TASKTYPE_DELAY, NULL, NULL, 10, STATE_BUSY);`
 Clique no botão **OK** para fechar a janela de código.

```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4 /**Task Sequence Example 1*/
5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecuter, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this fiel
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executer
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
18
19 inserttask(ts,TASKTYPE_TRAVEL,current,NULL);
20 inserttask(ts,TASKTYPE_FRLOAD,item,current,port);
21 inserttask(ts,TASKTYPE_TRAVEL,centerobject(current,2),NULL);
22 inserttask(ts,TASKTYPE_DELAY,NULL,NULL,10,STATE_BUSY);
23 inserttask(ts,TASKTYPE_TRAVEL,outobject(current,port),NULL);
24 inserttask(ts,TASKTYPE_FRUNLOAD,item,outobject(current,port),opipno(current,port));
25
26 dispatchtasksequence(ts);
27 // return a 0 so this object will know that you made your own tasksequence and it doesn't n
28 //to make the standard tasksequence automatically

```

- Clique no botão **OK** da janela de propriedades para fechá-la.

Etapa 4: Resete e Execute o Modelo

- **Resete e Rode o Modelo.** O operador deve viajar para o Queue, pegar um item, viajar para o BasicFR, esperar por 10 segundos, viajar para o Processor, e descarregar ou soltar o item.
- **Salve o modelo.** Você precisará usá-lo nas próximas etapas.

Sequência de Tarefas - Exercício 2

1. Introdução
2. Construção do modelo passo a passo

Introdução

Neste tutorial, você irá utilizar o modelo desenvolvido no tutorial 1 (Custom Task Sequence). O operador irá agora ficar no processor para processar o item antes dele iniciar a nova sequência de tarefas. Este tutorial assume um conhecimento básico sólido de interação com o software e iremos solicitar que você escreva algumas linhas de código básico.

Se em alguma momento você encontrar alguma dificuldade enquanto estiver construindo este modelo, um tutorial funcional completo pode ser encontrado em <http://www.flexsim.com/tutorials>

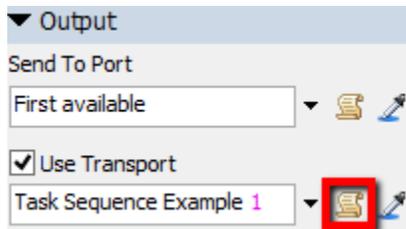
Construção do modelo passo a passo

Etapa 1: Carregue o modelo

Se você ainda não o fez, carregue o modelo do Exercício 1 de sequência de tarefas.

Etapa 2: Adicione as tarefas

- Dê um duplo clique no primeiro Queue para abrir a janela de propriedades, e clique na guia **Flow**.
- Clique no botão **Code Edit**  que fica do lado direito da listagem do **Use Transport** para abrir o editor de código



- Adicione uma nova linha após a Line 24. Na linha 25, digite o seguinte:
`inserttask(ts, TASKTYPE_UTILIZE, item, outobject(current, 1), STATE_UTILIZE);`

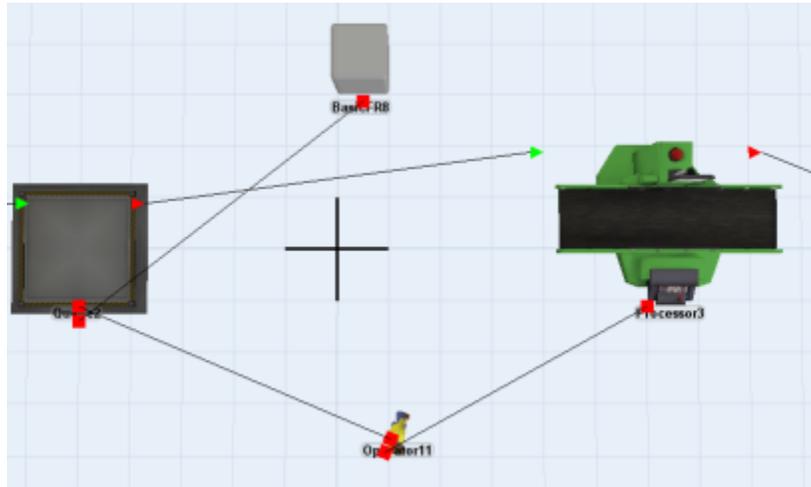
```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4 /**Task Sequence Example 1*/
5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecuter, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this fiel
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executer
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
18
19 inserttask(ts,TASKTYPE_TRAVEL,current,NULL);
20 inserttask(ts,TASKTYPE_FRLOAD,item,current,port);
21 inserttask(ts,TASKTYPE_TRAVEL,centerobject(current,2),NULL);
22 inserttask(ts,TASKTYPE_DELAY,NULL,NULL,10,STATE_BUSY);
23 inserttask(ts,TASKTYPE_TRAVEL,outobject(current,port),NULL);
24 inserttask(ts,TASKTYPE_FRUNLOAD,item,outobject(current,port),opipno(current,port));
25 inserttask(ts,TASKTYPE_UTILIZE,item,outobject(current,1),STATE_UTILIZE);
26 |
27 dispatchtasksequence(ts);
28 // return a 0 so this object will know that you made your own tasksequence and it doesn't n
```

- Clique em **OK** para fechar a janela de edição de código.

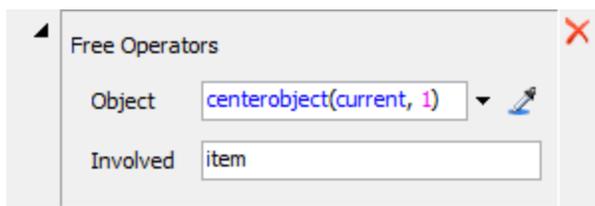
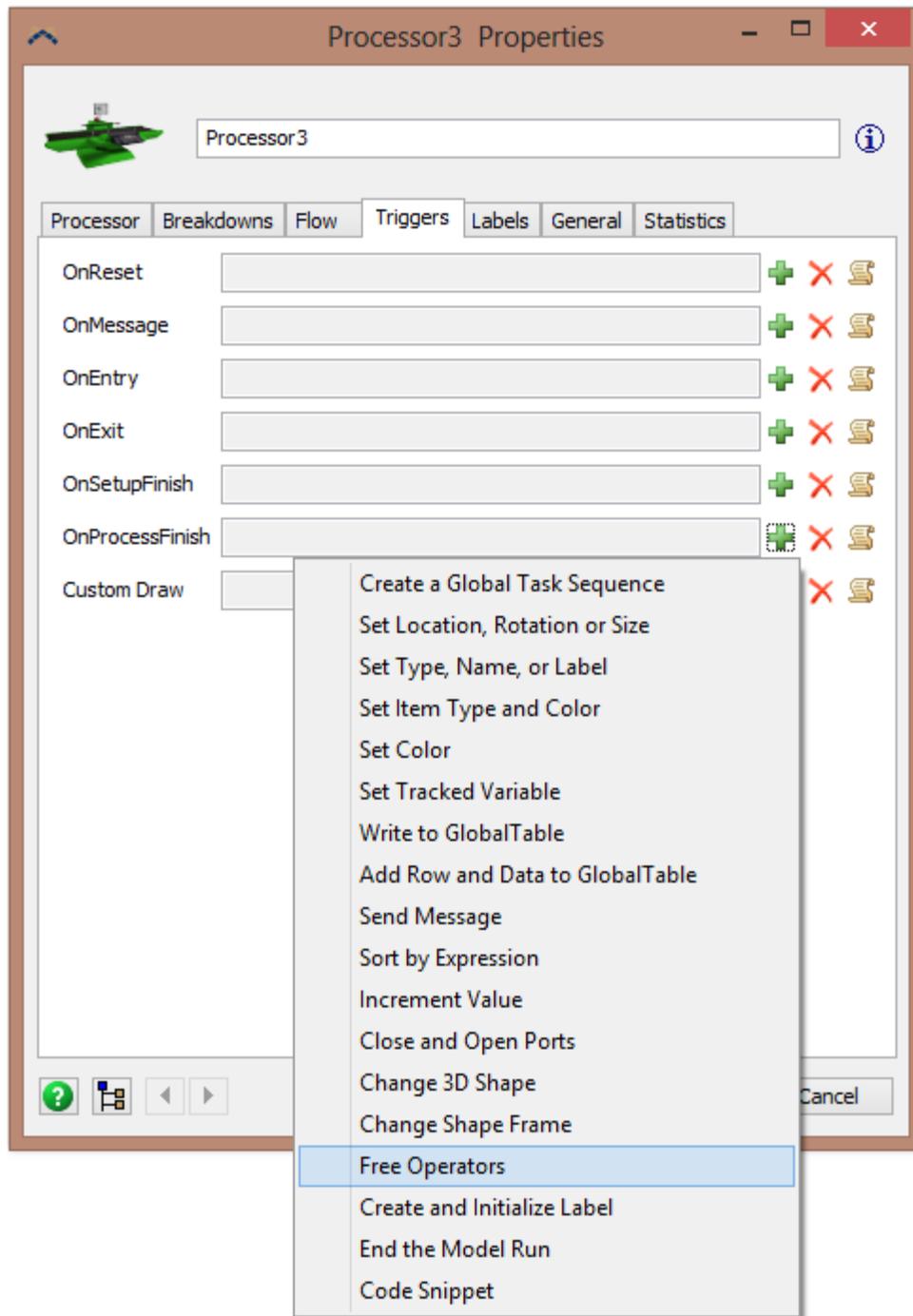
Se você rodar o model agora, você irá notar que o operador permanece no processador após largar o item no processador para ser processado. Isto se deve ao fato de que nada está determinando que o operador saia do local e vá para um outro objeto. O melhor local para dar este comando ao operador é no OnProcessFinish do Processor que está utilizando o operador.

Etapa 3: Edite o processador para liberar o operador

- Conecte o operador ao primeiro Processor com uma porta de conexão central(S)



- Dé um duplo clique no Processor para abrir a janela de propriedades, então clique na guia **Triggers**. Na trigger do **OnProcessFinish** selecione **Free Operators**. O parâmetro da trigger padrão irá funcionar para este modelo. Para este caso estamos falando do comando `TASKTYPE_UTILIZE` que nós usamos no exercício anterior. Para que o operador possa ser liberado do processador para executar uma próxima tarefa, os objetos devem coincidir com um ponto, neste caso, item. Siga o que foi feito na tela abaixo.



- Clique no botão **OK** da janela de propriedades para fechá-la.

Etapa 4: Resete e Execute o Modelo

- **Resete e Rode o Modelo.** O operador deve caminhar até o Queue, pegar um item, caminhar até o BasicFR, aguardar por 10 segundos, ir até o Processor, descarregar o item, e aguardar no processor durante o process time
- **Salve o modelo.** O próximo tutorial utilizará o que fizemos até aqui.

Sequência de Tarefas - Exercício 3

1. Introdução
2. Construção do modelo passo a passo

Introdução

Neste tutorial, você irá utilizar o modelo desenvolvido no tutorial 1 (Custom Task Sequence). O operador irá agora pegar o item do Processor e carregá-lo para o segundo Queue. Este tutorial assume um conhecimento básico sólido de interação com o software e iremos solicitar que você escreva algumas linhas de código básico.

Se em alguma momento você encontrar alguma dificuldade enquanto estiver construindo este modelo, um tutorial funcional complete pode ser encontrado em <http://www.flexsim.com/tutorials>

Para mais informações sobre TaskType e seus parâmetros, veja a página sobre Task Types.

Construção do Modelo passo a passo

Etapa 1: Carregue o Modelo

Se você tem, abra o modelo de Task Sequence 2 do exercício anterior.

Etapa 2: Delete o que encontra-se na Trigger OnProcessFinish do primeiro processor

Tendo em vista que você adicionará uma sequência de tarefas, uma mudança precisa ser feita na forma como o Operator é liberado a partir da tarefa Utilize. O comando freeoperator () precisa ser movido para o campo Request Transport, para o modelo trabalhar corretamente. Se você precisar de mais informações e com uma explicação em detalhe, isso será fornecido no final deste tutorial.

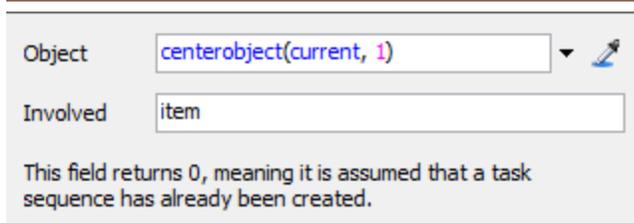
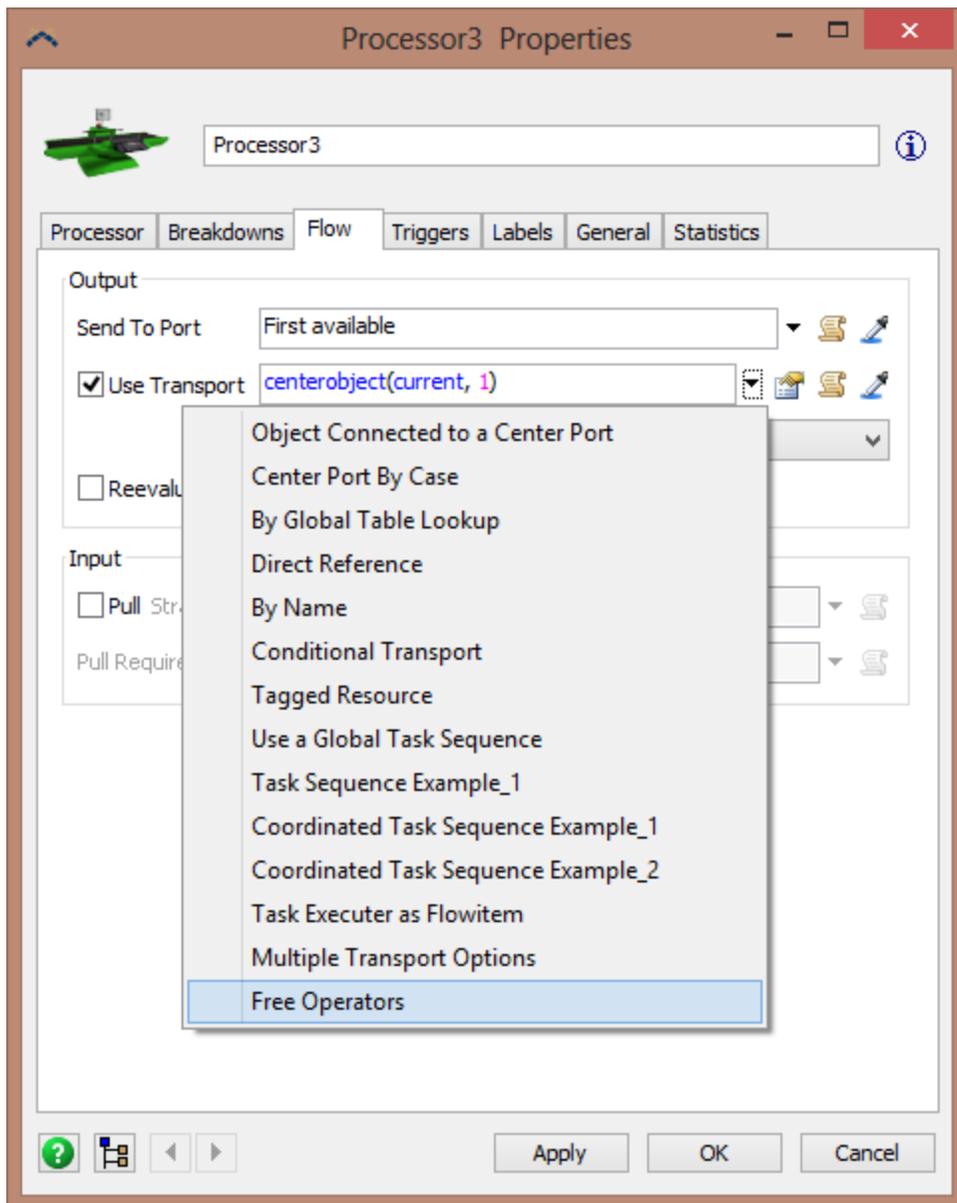
- Dê um duplo clique no primeiro Processor para abrir a janela de propriedades, e clique na guia **Triggers**.
- Na trigger **OnProcessFinish**, clique no botão  e clique no botão  para remover a função desta trigger.
- Não feche a janela de propriedades ainda.

Etapa 3: Escreva a lógica de fluxo

Agora você terá de liberar o operador no campo Request Transport From. Além disso, uma vez que o próximo conjunto de tarefas que você irá adicionar substituirá a lógica de fluxo do processador, especificamente o Request Transport From, você precisa dizer ao processador que ele não precisa criar uma sequência de tarefas.

Nota: Sempre que você escrever uma sequência de tarefas que controla a saída de um objeto de algum lugar diferente de Request Transport From nas propriedades do objeto a sequência de tarefas será afetada, e você DEVE retornar 0 na lógica Request Transport From do objeto, caso contrário, haverá sérios problemas. Neste exemplo, você irá escrever a sequência de tarefas no Queue, mas as próximas tarefas que você adicionar afetarão a lógica natural de transporte dos Processors, apesar de que a sequência de tarefa (task sequence) no Queue, precisará retornar 0 na lógica Processor Request Transport From. Você também precisa fazer isso se você estivesse escrevendo uma sequência de tarefas em um campo Trigger, em vez de fazer no campo Request Transport From.

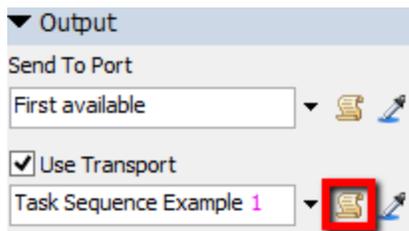
- Dê um duplo clique no primeiro Processor para abrir a janela de propriedades, e clique na guia **Flow**.
- Check a caixa **Use Transport**, então escolha **Free Operators** de listagem drop down. Os parâmetros padrões devem funcionar bem para este modelo. Note que uma observação está informando o retorno de 0 para você, ou seja, você não tem que se preocupar em escrever nada mais.



- Clique no botão **OK** para fechar a janela de propriedades.

Etapa 3: Escreva o restante da sequência de tarefas

- Clique sobre o primeiro Queue para abrir sua janela de propriedades Quick Properties.
- Na seção **Flow**, clique no botão **Code Edit**  ao lado da picklist **Use Transport** para abrir a janela de edição de código.



- Você precisará criar uma variável local para que você possa mais facilmente referenciar o segundo Queue na sua sequência de tarefa. Na linha 17, digite o seguinte:
`treenode downQueue = outobject(outobject(current, 1), 1);`

```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4 /**Task Sequence Example 1*/
5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecuter, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this field.
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executer
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode downQueue = outobject(outobject(current,1),1);
18 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
19
20 inserttask(ts, TASKTYPE_TRAVEL, current, NULL);
21 inserttask(ts, TASKTYPE_FRLOAD, item, current, port);
22 inserttask(ts, TASKTYPE_TRAVEL, centerobject(current,2), NULL);
23 inserttask(ts, TASKTYPE_DELAY, NULL, NULL, 10, STATE_BUSY);
24 inserttask(ts, TASKTYPE_TRAVEL, outobject(current, port), NULL);
25 inserttask(ts, TASKTYPE_FRUNLOAD, item, outobject(current, port), opipno(current, port));
26 inserttask(ts, TASKTYPE_UTILIZE, item, outobject(current, 1), STATE_UTILIZE);
27
28 dispatchtasksequence(ts);

```

- Começando na linha 27, digite o seguinte:
`inserttask(ts, TASKTYPE_FRLOAD, item, outobject(current, 1));`
`inserttask(ts, TASKTYPE_TRAVEL, downQueue, NULL);`
`inserttask(ts, TASKTYPE_FRUNLOAD, item, downQueue, 1);`

```

5 /**Creates a standard task sequence manually.*/
6 /**If this "Request Transport From" field returns a 0 rather than a valid pointer
7 to either a dispatcher or taskexecutor, then no call is made, and it is assumed
8 that the user will dispatch their own tasksequence.
9
10 This example shows the code that is required to create the exact same tasksequence
11 that is normally created automatically and dispatched to the object referenced by this field
12
13 treenode dispatcher = centerobject(current,1); // the dispatcher or task executor
14 double priority = getvarnum(current,"transportpriority"); // read the Priority value on the
15 int preempting = getvarnum(current,"preempttransport"); // read the Preemption mode on the
16
17 treenode downQueue = outobject(outobject(current,1),1);
18 treenode ts = createemptytasksequence(dispatcher,priority,preempting);
19
20 inserttask(ts,TASKTYPE_TRAVEL,current,NULL);
21 inserttask(ts,TASKTYPE_FRLOAD,item,current,port);
22 inserttask(ts,TASKTYPE_TRAVEL,centerobject(current,2),NULL);
23 inserttask(ts,TASKTYPE_DELAY,NULL,NULL,10,STATE_BUSY);
24 inserttask(ts,TASKTYPE_TRAVEL,outobject(current,port),NULL);
25 inserttask(ts,TASKTYPE_FRUNLOAD,item,outobject(current,port),opipno(current,port));
26 inserttask(ts,TASKTYPE_UTILIZE,item,outobject(current,1),STATE_UTILIZE);
27 inserttask(ts,TASKTYPE_FRLOAD,item,outobject(current,1));
28 inserttask(ts,TASKTYPE_TRAVEL,downQueue,NULL);
29 inserttask(ts,TASKTYPE_FRUNLOAD,item,downQueue,1);
30
31 dispatchtasksequence(ts);
32 // return a 0 so this object will know that you made your own tasksequence and it doesn't n

```

- Clique no botão **OK** na janela de códigos e de propriedades para fechá-las.

Nota: A razão pela qual o comando freeoperators () precisava ser alterado a partir do OnProcessFinish é devido ao fato de que uma sequência de tarefas escrita tem a capacidade de substituir a lógica interna dos objetos. Se o queue que está logo após ao processor não estiver disponível quando o processor terminar o processamento, então o processor irá liberar o operador para carregar / descarregar uma parte ou o item no Queue antes que esteja realmente disponível, o que pode fazer com que o Queue fique lotado de itens, e pode causar alguns problemas. Assim, movendo os freeoperators () para o campo Request Transport From, o operador só será liberado para continuar com sua tarefa de carregar/descarregar somente quando o Queue após o processor, estiver pronto para receber essa parte. A solução perfeita para este problema seria ter dois utilizes tasks, liberando o operador, tanto no OnProcessFinish, e no Request Transport From. Desta forma, o operador estará livre após o Processor finalizar a tarefa, no caso de você precisar dele para outra tarefa, mas ele não vai mover o item até que o objeto a jusante (Queue logo após o Processor) estiver disponível. Veja as informações sobre Fixed Resource nas páginas correspondentes.

Etapa 4: Resete e Execute o Modelo.

- **Resete e Rode o Modelo.** O Operador deve viajar para o Queue, pegar o item, ir até o Basic FR, aguardar por, viajar para o processor, largar o item, aguardar no processor durante o tempo de processamento, pegar o item, ir até o próximo Queue e largar o item.
- **Salve o modelo.**

Tutorial sobre SQL

1. Introdução
2. Construção do modelo passo a passo

Introdução

Este tutorial irá auxiliar você no entendimento de como conectar o FlexSim aos banco de dados SQL.

Nota: Esta orientação do tutorial não é para iniciantes. Assuma-se que você tenha um conhecimento básico sobre o FlexSim e é familiar com banco de dados SQL. Usaremos phpMyAdmin para o SQL durante este guia.

O que você irá aprender

- Enviar informações para os banco de dados SQL do Flexsim
- Ler dados de um banco de dados SQL para uso dinâmico de dentro do FlexSim

Tempo aproximado para completar este exercício

Este exercício deve ser completado em aproximadamente de 15-20 minutos.

Overview do modelo

Neste modelo, nós iremos pegar os dados de uma tabela de dentro de um banco de dados SQL para determinar os tempos de processamento. Nós iremos também enviar informações para outra tabela no mesmo banco de dados sobre os staytimes dos flowitem.

Clique aqui para um tutorial passo a passo.

Construção do modelo passo a passo

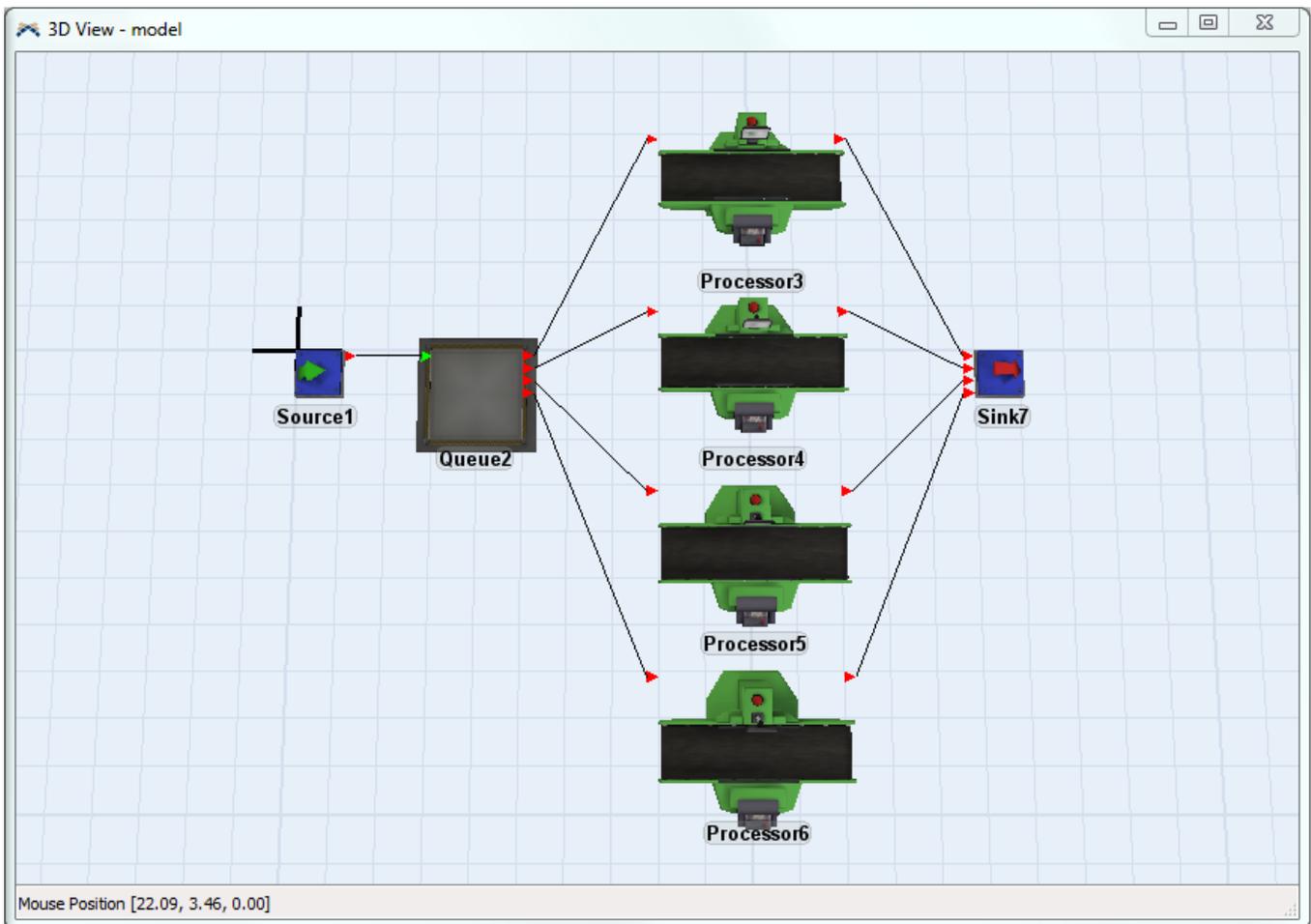
Construa um modelo SQL

Inicie um novo modelo ao clicar no botão  na barra de ferramentas. Clique Ok na janela de unidades do modelo, nós iremos usar as unidades padrões para o nosso modelo.

Se a qualquer momento você encontrar dificuldades enquanto desenvolve este modelo, um tutorial funcional e completo pode ser encontrado em <http://www.flexsim.com/tutorials>

Etapa 1: Crie os objetos

- Arraste os objetos de sua biblioteca (Library Icon Grid) para dentro da superfície de simulação 3D para criar o modelo conforme mostrado abaixo.



Conecte todos os objetos conforme informado Guiaixo:

- Conecte o *Source1* ao *Queue2*.
- Conecte o *Queue2* ao *Processor3*, *Processor4*, *Processor5* e *Processor6*.
- Conecte o *Processor3*, *Processor4*, *Processor5* e *Processor6* ao *Sink7*.

Etapa 2: Configure o banco de dados SQL

Nós iremos criar um banco de dados que poderá ler e enviar dados para o seu modelo no Flexsim. Primeiro, nós iremos criar uma tabela que armazena os tempos de processamento para cada equipamento (processor) baseado no tipo de item (itemtype).

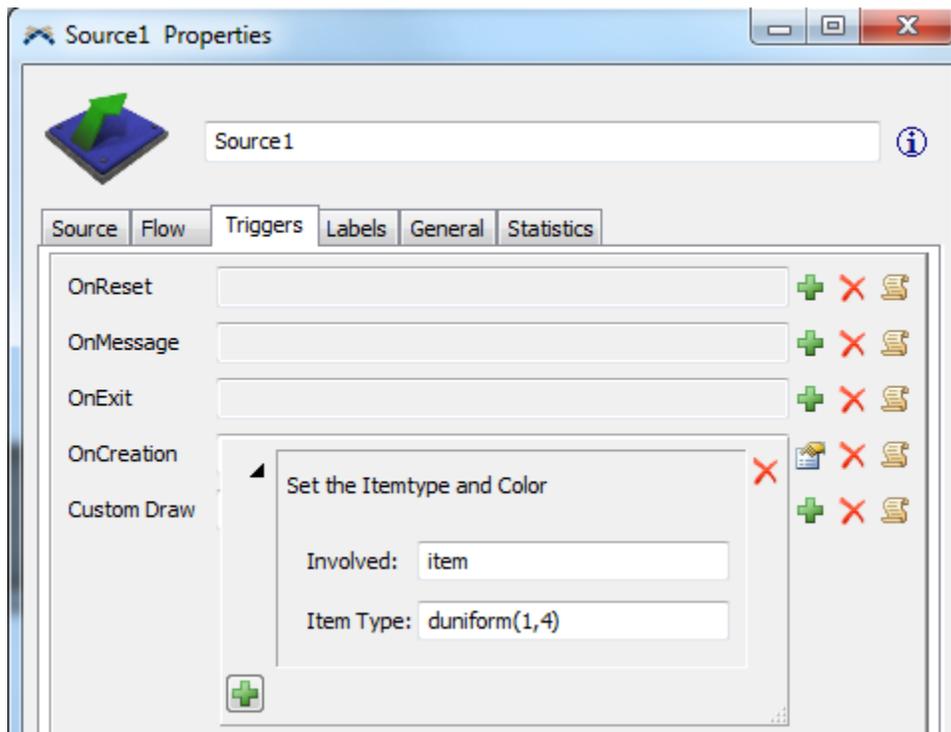
- Crie um banco de dados SQL chamado "flexsimdata"
- Crie uma tabela dentro de "flexsimdata" chamada "processtimes" com 4 colunas chamada "Proc1", "Proc2", "Proc3", e "Proc4". Cada uma destas colunas devem conter **FLOAT** number data.
- Preenche as colunas conforme mostrado na imagem seguinte:

Proc1	Proc2	Proc3	Proc4
3.2	3.4	3.6	3.8
4.2	4.4	4.6	4.8
5.2	5.4	5.6	5.8
6.2	6.4	6.6	6.8

Etapa 3: Configure o Source

Nós iremos criar 4 tipos de itens diferentes no Source.

- Altere o **Inter-Arrivaltime** do source para *exponential(0, 3, 0)*
- Vá para a guia **Triggers**.
- Clique no botão **+** para a trigger **OnCreation**.
- Selecione **Set ItemType and Color** das opções de picklist.
- Altere o **Item Type** para *duniform(1, 4)*



- Clique em **OK** para aplicar e fechar a janela de propriedades.

Etapa 4: Configure os processors

Cada um dos 4 processors irão acessar o banco de dados SQL que nós recentemente criamos para configurar seu tempo de processamento.

- Abra a janela de propriedades dos *Processors*.
- Clique no botão de edição de códigos  próximo ao *Process Time*.
- Entre com o seguinte código:

```
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3 treenode item = parnode(1);
4
5 dbopen("flexsimdata", "processtimes", 1);
6
7 double proctime = dbgettextenum(getitemtype(item), 1);
8
9 dbclose();
10
11 return proctime;
```

- O comando **dbopen()** irá acessar a base de dados e abri-la para leitura e escrita. O primeiro parâmetro no nome do banco de dados conforme nomeado em seu Windows ODBC Data Source Administrator (não necessariamente o mesmo mostrado em phpMyAdmin). O segundo parâmetro neste caso é a tabela que você quer ler ou enviar dados (escrever). O terceiro parâmetro alterna entre o modo de tabela (table mode) (1) e o SQL mode. Nós usamos 1 neste caso porque nós não estamos usando consultas diretas SQL, nós estamos declarando uma tabela para trabalhar dentro de um banco de dados
 - Nós usamos um comando especial chamado **dbgettextenum()** para pegar as informações de uma tabela no banco de dados. Em nosso caso, nós queremos que a linha contenha o tipo de item (itemtype), e nós queremos que cada coluna represente cada equipamento (processor). Nós iremos alterar o 1, 2, 3 e 4 para os outros processor.
 - Nós precisamos usar o **dbclose()** para fechar o banco de dados para que outros banco de dados possam ser acessados mais tarde.
- Clique em **OK** para aplicar e fechar a janela de propriedades.

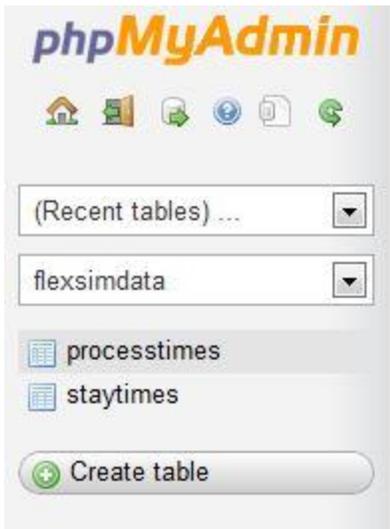
Escreva o mesmo código nos outros três processors, substituindo cada segundo parâmetro dos equipamentos (processors) consecutivos em **dbgettextenum** por 2, 3 e 4 respectivamente.

Resete e rode o modelo. Se você olhar na estatística staytime, você notará que ele corresponde aos valores no banco de dados *flexsimdata* na tabela *processtimes*.

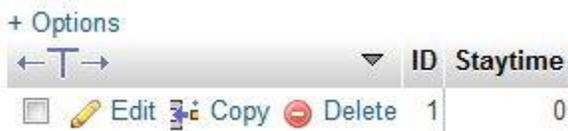
Etapa 5: Adicione a tabela dos Staytimes ao banco de dados.

Nós iremos agora adicionar outra tabela em nosso banco de dados SQL para armazenar os tempos de permanência de todos os itens fluindo em nosso modelo.

- Crie uma tabela dentro de "*flexsimdata*" chamada "*staytimes*" com 2 colunas chamada "*ID*" e "*Staytime*". A primeira coluna deve conter dados inteiros (**integer**) e deve ser auto incrementar e a segunda coluna deve conter dados numéricos **FLOAT**.



- Crie uma linha inicial conforme mostrado na seguinte figura com $ID = 1$ e $Staytime = 0$.



Etapa 6: Registrar no banco de dados SQL

Cada vez que um item entra no Sink, nós iremos registrar o tempo do item no Sistema (staytime) e registrar o valor para o banco de dados SQL.

- Abra a janela de propriedades do Sink.
- Vá para a guia **Triggers**.
- Clique no botão de edição  para a trigger **OnEntry**.
- Entre com o seguinte código:

```
string staytime = numtostring(time() - get(stats_creation_time(item)), 2, 3);
string query = concat("INSERT INTO `flexsimdata`.`staytimes` (`Staytime`)VALUES ('",
staytime, "');");
string altquery = concat("UPDATE staytimes SET Staytime =", staytime, " WHERE ID =",
numtostring(getinput(current)));

dbopen("flexsimdata", "SELECT * FROM staytimes", 0);
int rows = dbgetnumrows();
if (getinput(current) < rows)
{
    dbsqlquery(altquery);
}
else {
    dbsqlquery(query);
}
dbclose();
```

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 string staytime = numtostring(time() - get(stats_creationtime(item)),2, 3);
7
8 string query = concat("INSERT INTO `flexsimdata`.`staytimes` (`Staytime`)VALUES ('", staytime, "');");
9 string altquery = concat("UPDATE staytimes SET Staytime =", staytime, " WHERE ID =", numtostring(getinp
10
11 dbopen("flexsimdata", "SELECT * FROM staytimes", 0);
12 int rows = dbgetnumrows();
13 if (getinput(current) < rows)
14 {
15     dbsqlquery(altquery);
16 }
17 else
18 {
19     dbsqlquery(query);
20 }
21 dbclose();
22

```

Uma vez que conseguirmos o tempo de permanência registrado dentro da variável chamada *staytime*, nós fazemos uma query para adicionar os dados dentro da coluna **Staytime** da tabela *staytimes*. O query alternativo é feito para o caso em que a tabela já contenha os dados mas precisa ser sobrescrito. Portanto, temos a referência forçando a usar uma consulta alternativa enquanto houver linhas já existentes.

- Clique em **OK** para aplicar e fechar a janela de propriedades.

Resete e rode seu modelo. Você notará que a table *staytimes* de sua base dados está preenchida com os dados capturados do Sink, similar a seguinte imagem abaixo.

+ Options

	ID	Staytime
<input type="checkbox"/> Edit Copy Delete	1	4.8
<input type="checkbox"/> Edit Copy Delete	2	6.6
<input type="checkbox"/> Edit Copy Delete	3	6.8
<input type="checkbox"/> Edit Copy Delete	4	6.4
<input type="checkbox"/> Edit Copy Delete	5	3.6
<input type="checkbox"/> Edit Copy Delete	6	4.8
<input type="checkbox"/> Edit Copy Delete	7	3.2
<input type="checkbox"/> Edit Copy Delete	8	4.6
<input type="checkbox"/> Edit Copy Delete	9	6.4
<input type="checkbox"/> Edit Copy Delete	10	6.8

Isto completa o tutorial SQL! Parabéns.

Tutorial de Objetos Fluídos

1. Introdução
2. Construção do modelo passo a passo

Introdução

Este exercício irá introduzir a maior parte dos objetos fluidos do Flexsim. Você vai aprender como eles interagem entre si e como incluí-los em um modelo com os objetos discretos. A construção de um modelo com os objetos fluidos envolve mais informações e exige mais atenção aos detalhes que um modelo com os objetos discretos. Por esse motivo, é recomendável que você se sinta confortável na construção de modelos com os outros objetos antes de começar a aprender sobre os objetos de Fluidos.

O que você irá aprender

- Como modelar fluidos materiais com o FlexSim
- Como converter flowitems em fluido material
- Como transferir e estocar fluido material
- Como usar o registro de nível de fluídos em um tanque para controlar o fluxo de material
- Como misturar fluidos materiais juntos
- Como converter fluido material em flowitems

Novos Objetos

Neste exercício, iremos introduzir os objetos FluidTicker, ItemToFluid, FluidPipe, FluidTank, FluidMixer, FluidProcessor e FluidToItem.

Tempo aproximado para completar este exercício

Este exercício deve levar de 30-45 minutos para ser completada.

Overview do Modelo de Fluídos

Em nosso modelo para fluidos, teremos um operador carregando caixas de dois tipos diferentes de materiais para o modelo. Estas caixas serão convertidos em dois fluidos, que será transportada por tubulações para dois tanques. A partir dos tanques, o fluido é enviado para um misturador único que irá misturar os dois produtos em um novo produto. Este produto é enviado por meio de FluidProcessor, e, em seguida, convertido em flowitems que são transportadas por um Conveyor para o Sink. O fluido, neste modelo será medido em litros, e o tempo será em segundos.

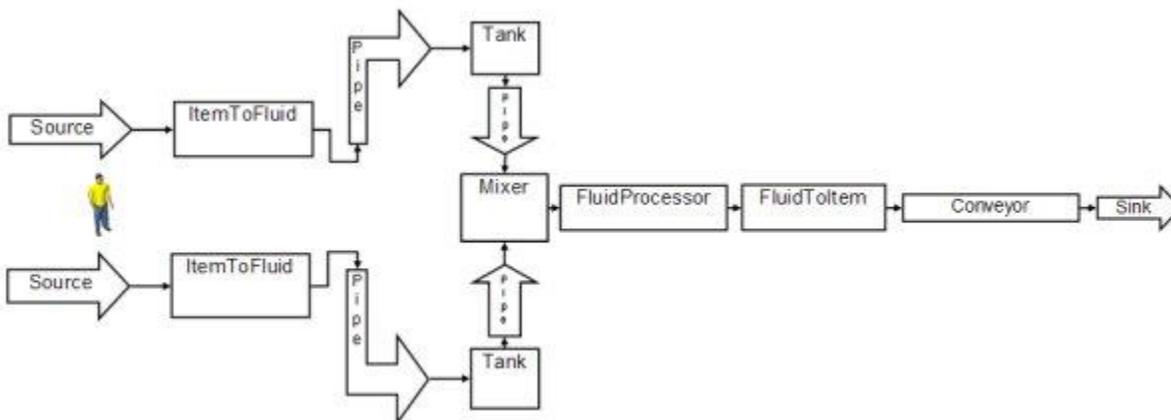


Figura 4-1 Diagrama do modelo de fluído

Dados do Modelo Fluído

Flowitem arrival rate: exponencial(0,10) segundos
Maximum Content of ItemToFluid: 20 galões
Fluid Units per Discrete Unit (ItemToFluid): 10 galões por flowitem
Maximum Content of Pipe leading to Tank: 20 galões
Transfer Rate (ItemToFluid to Tank): 2 galões por segundos
Tank Low Mark: 1 galão
Tank High Mark: 45 galões
Maximum Content of Pipe leading to Mixer: 10 galões
Transfer Rate (Tank to FluidToItem): 1 galão por segundos

Mixer Steps:

Step 1: Material1, sem tempo de atraso
Step 2: Material2, 10 segundos de atraso
Mixer Recipe:
Material1: 10 gallons, step 1
Material2: 20 gallons, step 2
Maximum Content of FluidToItem: 10 galões
Fluid Units per Discrete Unit (FluidToItem): 10 galões por flowitem

Novos conceitos

Terminologias do FlexSim

Antes de você começar este modelo, seria útil entender algumas terminologias básicas do sistema de fluidos do FlexSim.

Fluido: Qualquer material que não é facilmente e eficientemente modelado como um flowitem discreto. Normalmente, os materiais que são medidos por peso ou volume é difícil de modelar com flowitems. Isto é porque frequentemente parte de uma unidade (por exemplo, metade de um galão) pode ser movida por si só. Não existe nenhum mecanismo fácil para mover metade de um flowitem. Fluido material podem representar também objetos que são tão numerosos que flowitems torna-se impraticáveis. Por exemplo, milhares de garrafas em uma linha de enchimento irão deixar o modelo devagar em relação a um que usa um flowitem para cada garrafa. Em vez disso, os objetos de fluidos podem ser usados para modelar estas garrafas sem a sobrecarga que vem com os flowitems.

Fluido Objetos: Os onze objetos que são projetados para lidar com materiais fluidos. Nove deles não podem interagir com objetos discretos da Flexsim, mas dois deles são projetados para funcionar como uma interface entre os objetos de fluido e os objetos discretos. Mais informações podem ser encontradas aqui

Tick: Os objetos fluidos enviam e recebem materiais em certos intervalos. Esses intervalos são chamados de "ticks". No final de cada tick, os objetos fluidos calculam quanto materiais foram enviados e recebidos durante esse período.

Tick time: O comprimento de cada tick. O modelador pode definir esse valor para um valor que seja apropriado para seu modelo. Uma marca de tempo mais curto pode tornar o modelo mais preciso, mas também pode torná-lo mais lento. Um valor mais longo será um modelo mais rápido, mas o custo é uma perda de precisão. Cabe a cada modelador escolher o melhor trade-off de velocidade e precisão para o seu modelo.

Rate: A velocidade máxima no qual um material entra ou deixa um objeto. Geralmente, o fluido objeto têm uma taxa de entrada e uma de saída que estão separados um do outro. Em alguns objetos, a taxa no qual materiais entram vai afetar a taxa em que ele sai. Para esses objetos, ao modelador não é dado a oportunidade de editar a taxa de saída. A taxa real na qual os materiais entram ou saem é baseada em vários fatores: a taxa de saída do objeto a montante, a taxa de entrada do objeto a jusante, a quantidade

de materiais disponíveis ser enviado ao próximo equipamento e quantidade de espaço disponível no objeto subsequente.

Object Rate: Esta é a taxa máxima em que os materiais podem entrar ou sair de um objeto através de todas as portas de entrada e saída em conjunto. Os objetos normalmente têm uma taxa separada para as portas de entrada e as portas de saída. Se, no final de cada tick, o objeto calcula a quantidade de materiais que tenham sido enviados ou recebidos e se objeto atingiu a taxa máxima, não mais itens serão enviados ou recebidos para esse tick, mesmo se existirem portas que ainda não enviam ou recebem item.

Port Rate: Esta é a taxa máxima em que os materiais podem entrar ou sair de um única porta sobre o objeto. Os objetos têm diferentes taxas para portas de entrada e saída. Este valor único aplica-se a todas as portas de entrada ou saída. Ele não pode ser alterado para afetar as portas individuais.

Port Scale Factor: Este é um número que é usada para alterar a taxa para cada porta individualmente. Há uma escala de fator disponível para cada porta de entrada e saída. O valor para cada porta é multiplicado pela taxa de máxima da porta para encontrar a taxa máxima atual para essa porta.

Construção do Modelo passo a passo

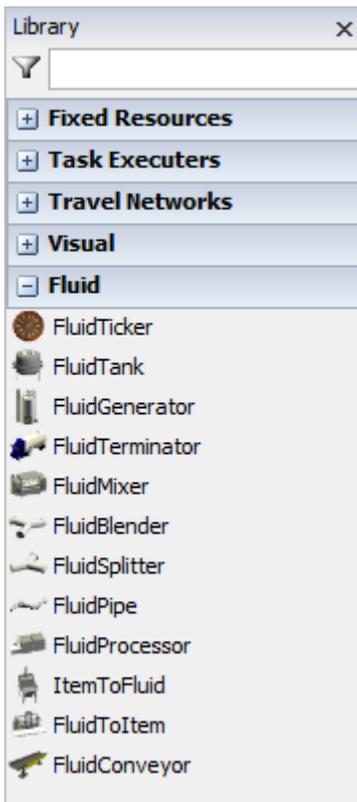
Construindo o modelo de fluídos

Para iniciar a construção de modelos de fluídos, você precisará começar um novo modelo. Não inicie com os modelos que você salvou de exercícios anteriores. Espera-se que você saiba como criar objetos, conectar portas e usar as propriedades do GUI.

Se a qualquer momento você encontra dificuldades enquanto constrói este modelo, um tutorial com funcionalidades completa pode ser encontra em <http://www.flexsim.com/tutorials>

Etapa 1: Layout do modelo e conexões

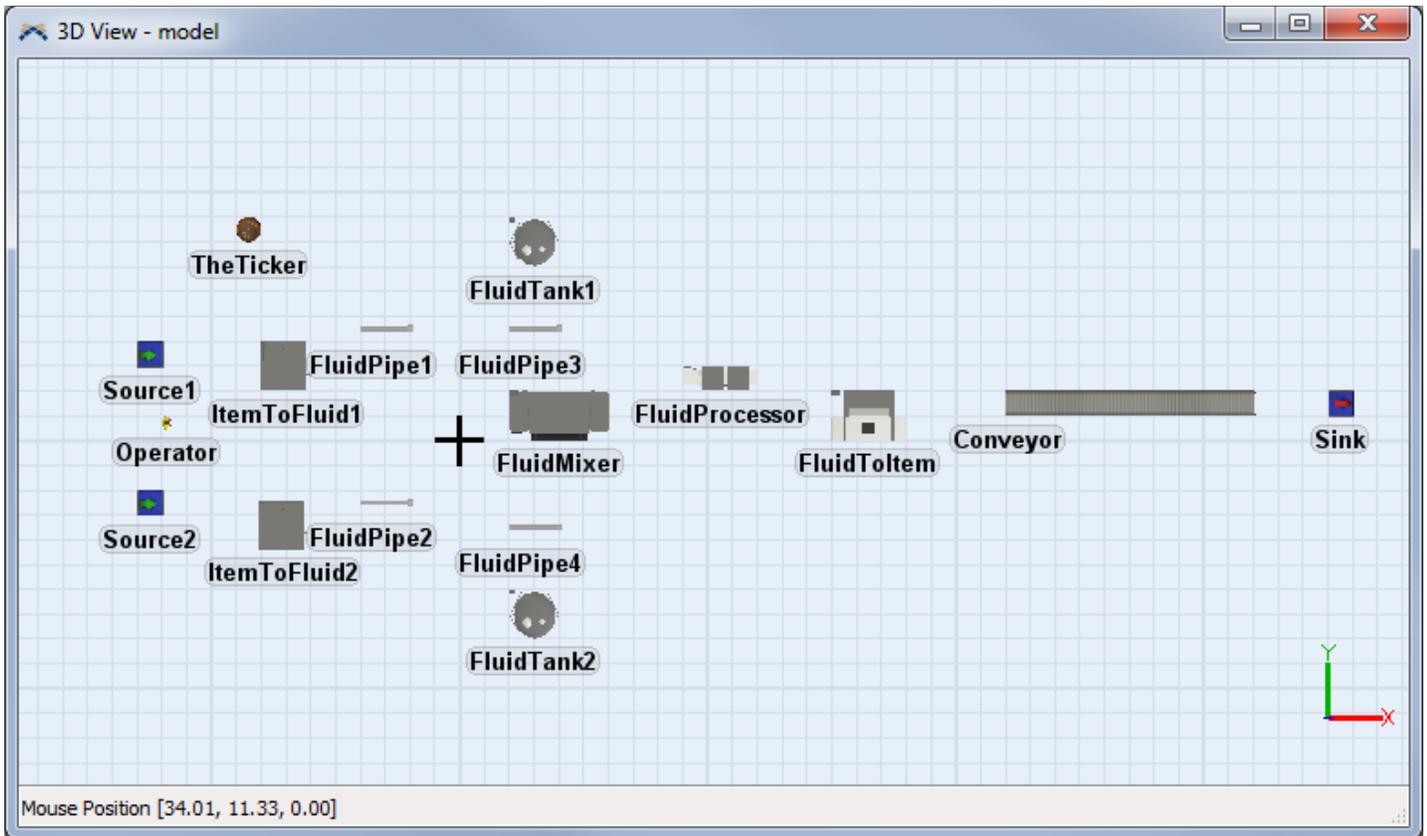
Para iniciar este exercício, arraste os seguintes objetos da biblioteca. Estes objetos podem ser encontrados na biblioteca ao clicar no botão + próximo a “Fluid” encontrado próximo ao botão da janela da biblioteca:



Nota: Quando você cria o primeiro Fluido Objeto, um Ticker é automaticamente criado e colocado na posição (0,0). Você pode movimentar em qualquer posição dentro do modelo, mas não o delete. Precisa-se dele para o Fluido Objeto funcionar corretamente.

- 2 **Sources** com nome *Source1* e *Source2* (Discrete Objects).
- 1 **Operator** com nome *Operator* (Discrete Objects).
- 2 **ItemToFluids** com nome *ItemToFluid1* e *ItemToFluid2* (Fluid Objects).
- 2 **FluidPipes** com nome *FluidPipe1* e *FluidPipe2* (Fluid Objects).
- 2 **FluidTanks** com nome *FluidTank1* e *FluidTank1* (Fluid Objects).
- 2 + **FluidPipes** com nome *FluidPipe3* e *FluidPipe4* (Fluid Objects).
- 1 **FluidMixer** com nome *FluidMixer* (Fluid Objects).
- 1 **FluidProcessor** com nome *FluidProcessor* (Fluid Objects).
- 1 **FluidToItem** com nome *FluidToItem* (Fluid Objects).
- 1 **Conveyor** com nome *Conveyor* (Discrete Objects).

- 1 **Sink** com nome *Sink* (Discrete Objects).
- Arranje os objetos como mostrado abaixo.

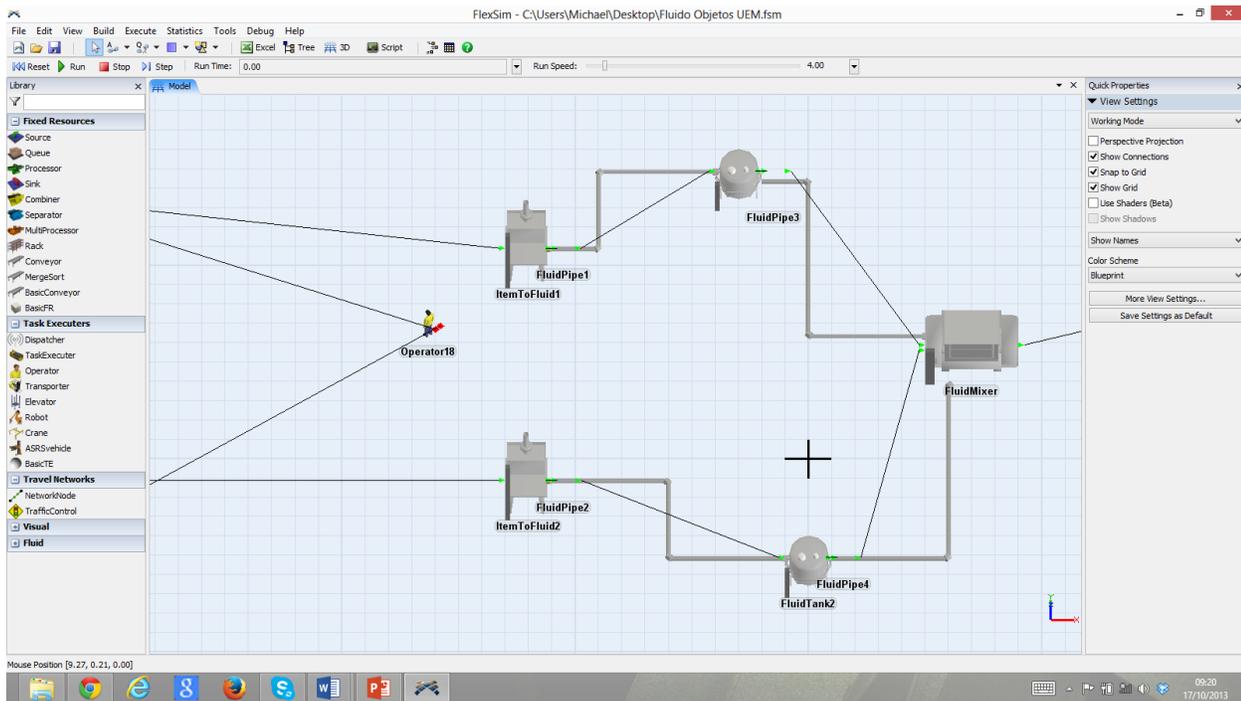


Uma vez que os objetos foram criados e posicionados onde você quiser em seu modelo, eles precisam ser conectados. Você usará as mesmas teclas usadas para conectar os objetos discretos, para conectar os objetos fluidos: A tecla A cria conexões de entrada/saída e a tecla S cria uma conexão de porta central.

- Conecte os objetos para que exista uma linha de processamento
 - do *Source1* para *ItemToFluid1*
 - do *ItemToFluid1* para *FluidPipe1*
 - do *FluidPipe1* para *FluidTank1*
 - do *FluidTank1* para *FluidPipe3*
 - do *FluidPipe3* para *FluidMixer*
 - do *FluidMixer* para *FluidProcessor*
 - do *FluidProcessor* para *FluidToltem*
 - do *FluidToltem* para *Conveyor*
 - do *Conveyor* para *Sink*

nesta ordem, como mostrado logo abaixo.

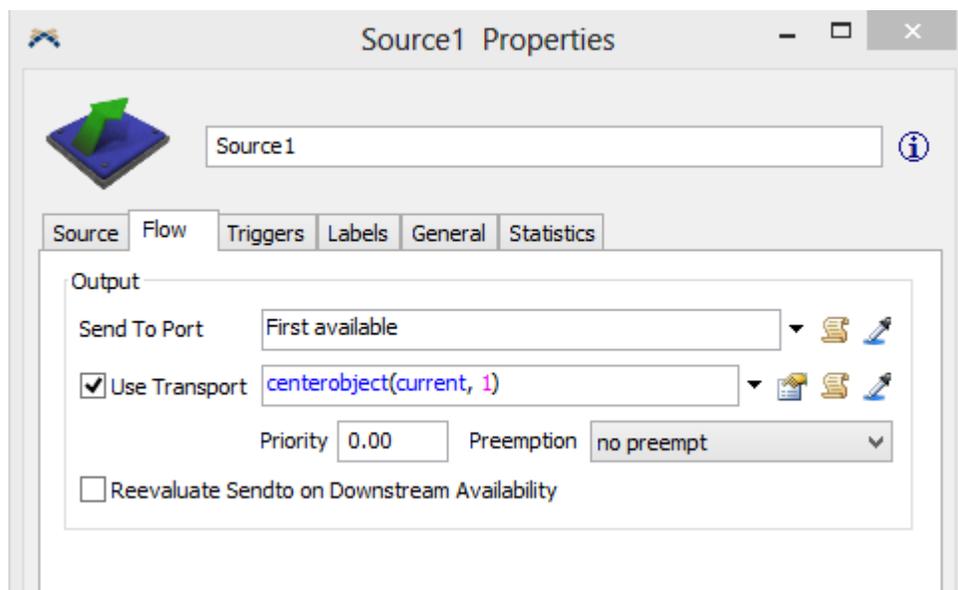
- Crie uma linha paralela da *Source2* para o *FluidMixer*
- Ambos sources precisam chamar *Operator* para transportar o flowitem para *ItemToFluid*, então uma conexão de porta central precisa ser feita de cada um dos Sources, Source 1 e Source 2 para o *Operator*.



Etapa 2: Configure os Sources

O padrão inter-arrival time para os Source irá trabalhar bem para este modelo. Eles precisam apenas chamar um Operator para transportar o flowitems que eles criaram para o objeto ItemToFluid.

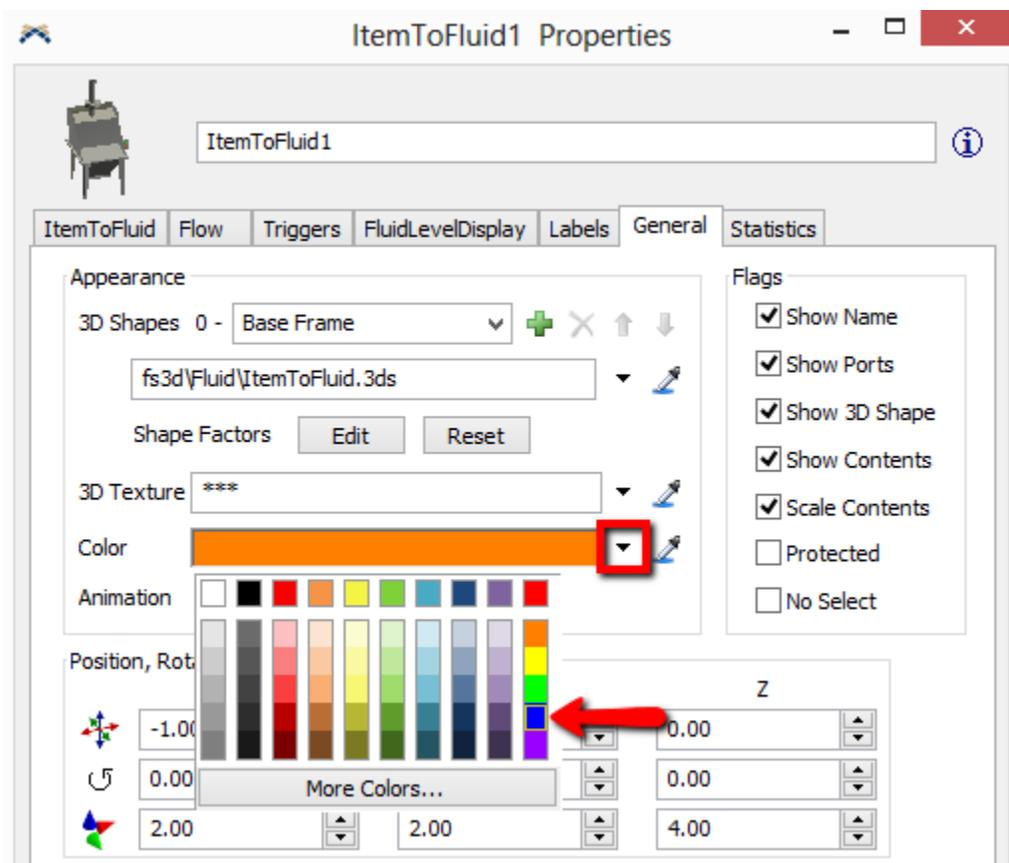
- Dê um duplo clique no *Source1* para abrir sua janela de propriedades. Clique na guia **Flow**, e check **Use Transport**.
- Clique **OK** para aplicar as mudanças e feche a janela **Properties**.
- Repita este passo para o *Source2*.



Etapa 3: Configure as cores dos objetos

Quando os objetos são criados, eles terão cores diferentes, dependendo de suas classes. As vezes se torna útil colorir os objetos baseado no tipo de material que eles irão processar. Neste modelo, existem duas linhas de processamento que são cada uma composta de um *ItemToFluid*, Pipe, Tank e outra Pipe. Usando as propriedades do GUI, nós iremos mudar a cor destes quatro objetos em uma linha para azul e os objetos da outra linha para vermelho. Isto irá fazer com que algumas partes destes objetos mudem de cor conforme o modelo está rodando.

- Dê um duplo clique em *ItemToFluid1* para abrir a janela **Properties**, e clique na guia **General**.
- Clique no botão ▾ localizado próximo ao campo **Color**. O popup de cor irá aparecer. Selecione a cor azul dentre as opções, e então clique em **OK**.
- Clique em **OK** para aplicar as mudanças e feche a janela **Properties**.
- Repita este passo para o resto dos objetos mencionados acima, colorindo a outra linha de processamento para vermelho ao invés de azul.

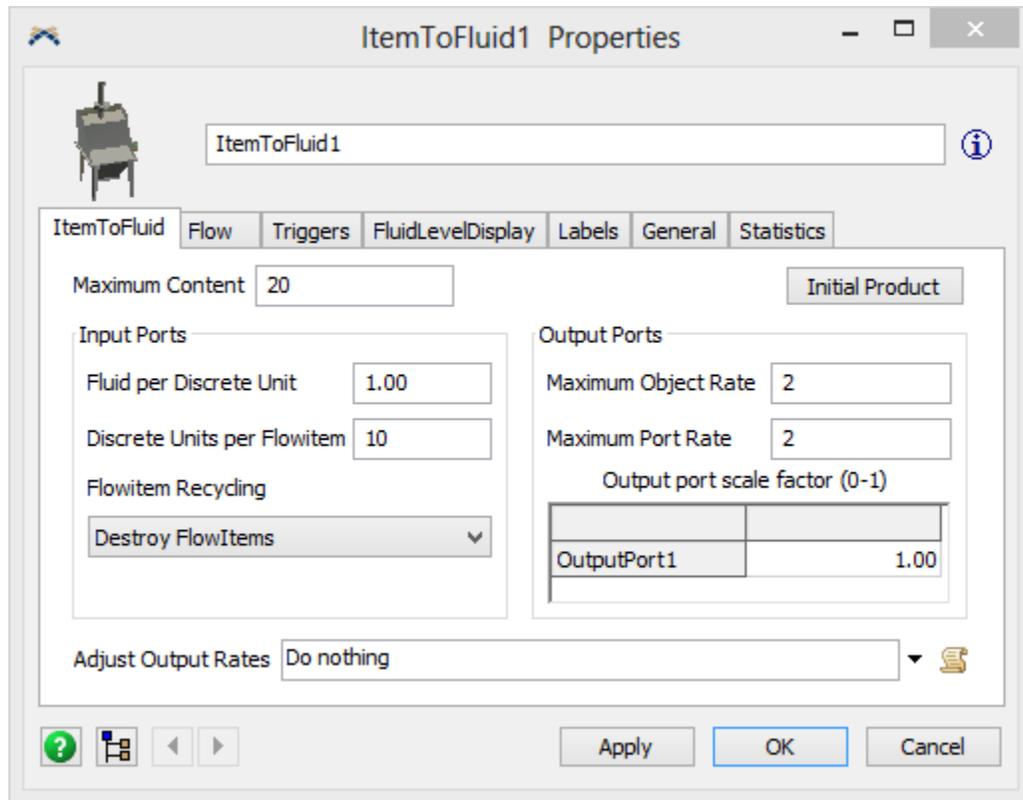


Etapa 4: Configure o ItemToFluid

Agora o objeto *ItemToFluid* tem que ser configurado para criar a correta quantidade de material para cada flowitem que chega.

- Dê um duplo clique em *ItemToFluid1* para abrir a janela **Properties**. Na guia **ItemToFluid**, mude o campo **Discrete Units per Flowitem** para 10. Isto significa que o *ItemToFluid* irá criar 10 galões de fluido para cada flowitem que entra no modelo.

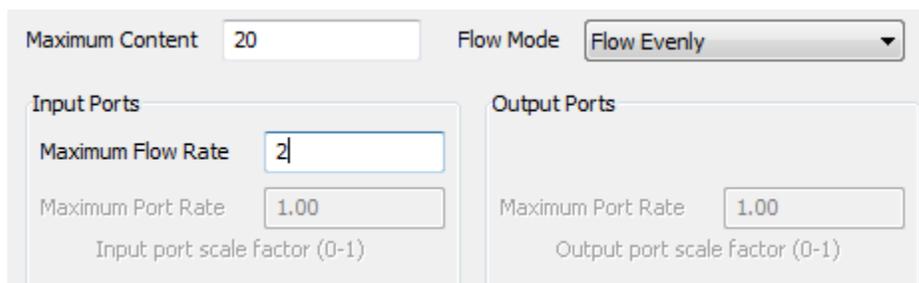
- Mude a taxa máxima do objeto (**Maximum Object Rate**) e taxa máxima da porta (**Maximum Port Rate**) para 2.
- Mude a capacidade máxima (**Maximum Content**) para 20.
- Clique **OK** para aplicar as mudanças e feche a janela **Properties**.
- Repita o passo para o *ItemtoFluid2*.



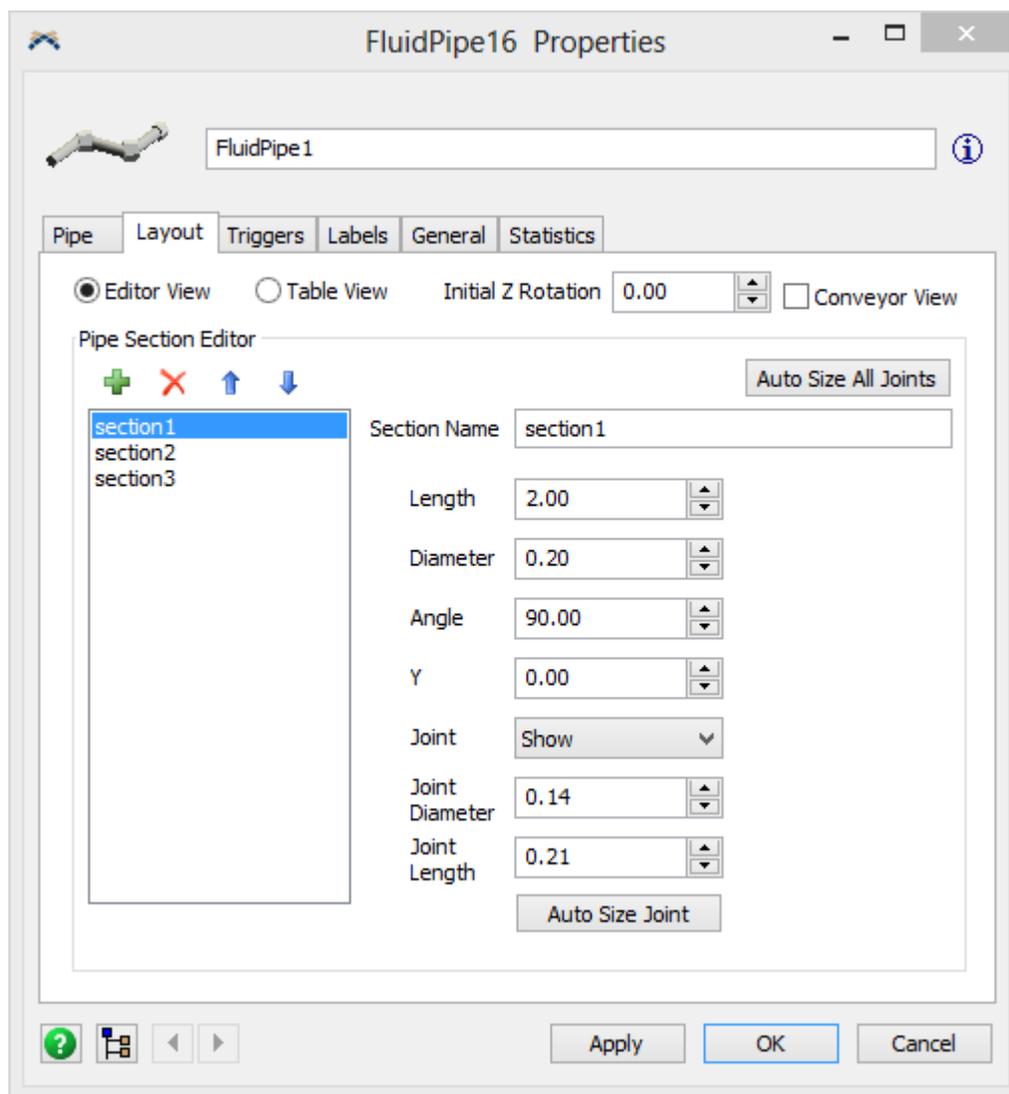
Etapa 5: Configure as tubulações (FluidPipe)

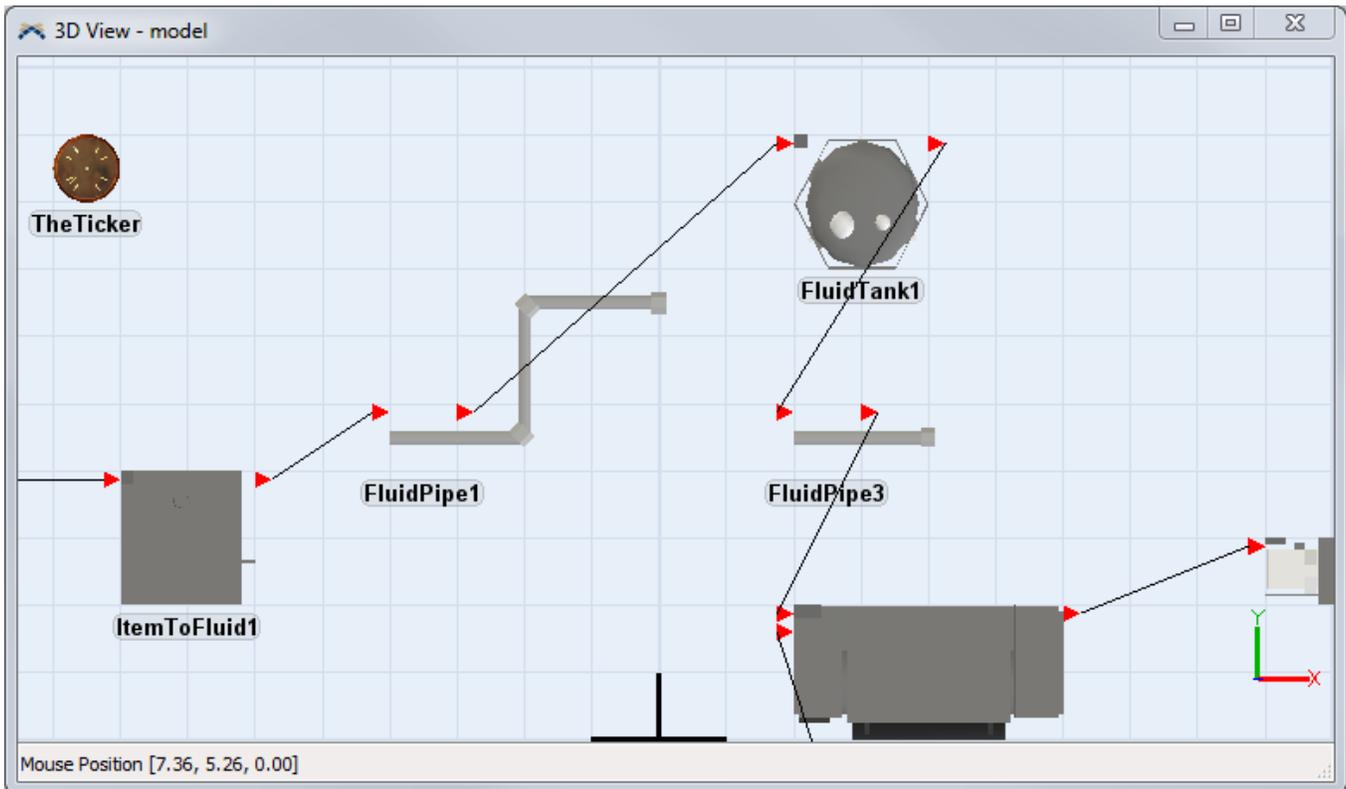
As tubulações (pipes) que transportam o material do ItemToFluids para os demais objetos, são os próximos objetos que precisam ser configurados. Nos Pipes não é específico modelar a especificação de taxas de entradas e saídas. A taxa de saída é baseado na taxa real de material que foi recebido.

- Dê um duplo clique em *FluidPipe1* para abrir a janela **Properties**.
- Na guia **Pipe**, configure a taxa de fluxo máxima (**Maximum Flow Rate**) para 2, e configure a capacidade máxima (**Maximum Content**) para 20. Isto irá fazer com que o material passe um tempo dentro de Pipe, mas não muito tempo.



- Clique na guia **Layout**. Ajuste o FluidPipe, alterando e/ou adicionando seções em pipe, para que possa demonstrar que está iniciando no *ItemToFluid* e terminando próximo a *FluidTank* (o valor real de seu FluidPipe pode ser diferente dos mostrados guiaixo). Alterando o layout não afetará o comportamento dos pipes. Mas informação sobre esta parte pode ser visualizado abaixo.



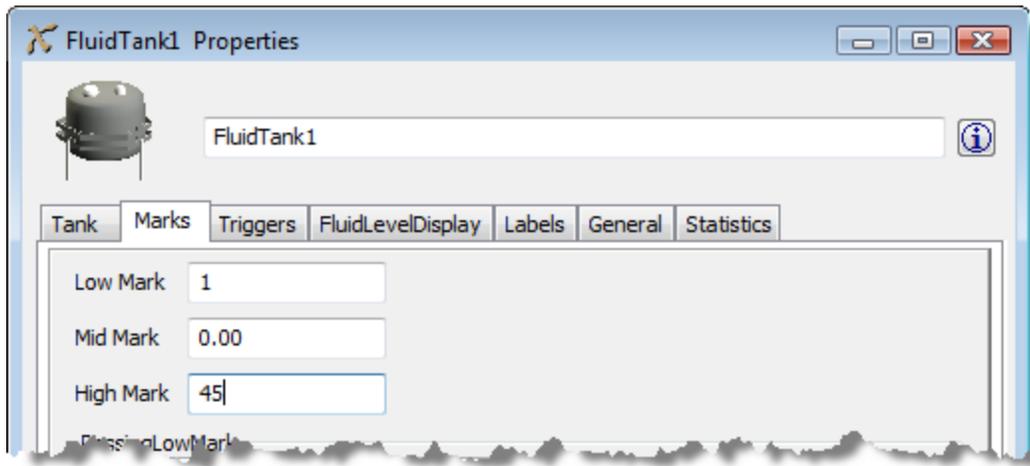


- Clique no botão **OK** para aplicar as mudanças e feche a janela **Properties**. Repita este passo para o *FluidPipe2*.

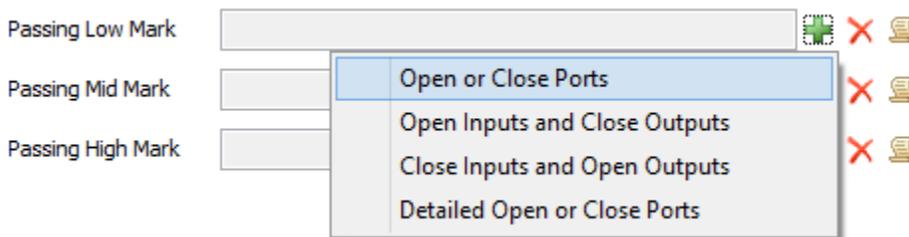
Etapa 6: Configure os tanques de fluídos (FluidTanks)

Os *ItemToFluids* têm agora um taxa máxima de saída de 2, mas os *FluidTanks* têm uma taxa de entrada máxima de 1. Se esses valores são deixados como estão, a taxa do *FluidTank* será usado durante a execução do modelo (porque é o valor mais baixo dos dois apresentados) e o *FluidPipe* não será capaz de enviar o material a jusante tão rápido como você informou ao sistema que ele deveria fazer. Assim, a taxa sobre os *FluidTanks* precisa ser mudado. *FluidTanks* permite o modelador definir três níveis que farão com que as trigger possam ser ativadas e disparadas quando o conteúdo do *FluidTank* alcança-los. Estes valores são chamados de marcas (marks). Eles são editados na guia Marks da janela Propriedades. Para este modelo, os tanques devem manter as portas de saída fechada até que tenham recebido uma certa quantidade de material. Eles, então, abrem as portas de saída e as deixa abertas até que o tanque fique vazio. Eles sempre mantem suas portas de entrada aberta.

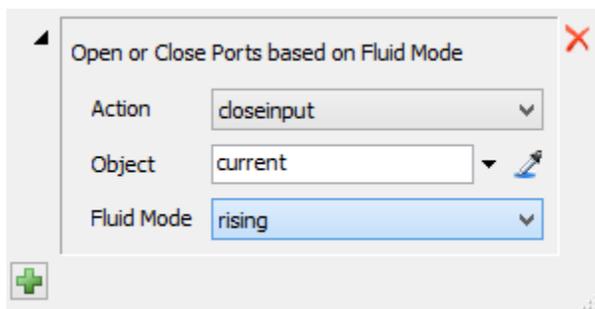
- Dê um duplo clique em *FluidTank1*. Na guia **Tank**, coloque em **Maximum Object Rate** e **Maximum Port Rate** o número 2.
- Clique na guia **Marks**. Mude o **Low Mark** para 1, e o **High Mark** para 45. Deixe o **Mid Mark** com 0. Se mark está registrado como 0, o trigger para este mark não será disparado.



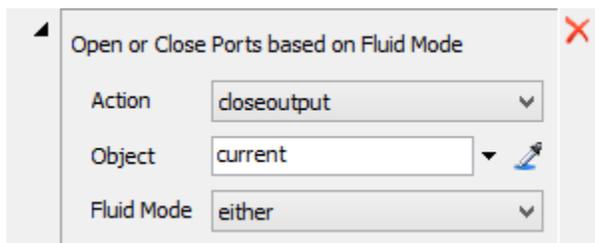
- Na guia **Marks**, clique em  para adicionar uma função na trigger **Passing Low Mark** . Selecione a opção **Open or Close Ports**.



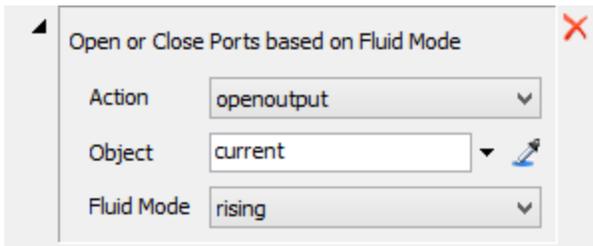
- Clique onde está dizendo rising. As opções disponíveis irão aparecer.



- Dê as funções os seguintes parâmetros:



- Adicione a função **Open or Close Ports** para a trigger **Passing High Mark** da mesma forma. Forneça os seguintes parâmetros:



- Clique em **OK** para aplicar as mudanças e feche a janela **Properties**.
- Repita o mesmo procedimento para **FluidTank2**.

Etapa 7: Reoriente e dimensione as próximas tubulações (Pipes)

Dependendo de onde você colocou os objetos em seu modelo, as tubulações (pipes) que conduzem dos Tanks para o Mixer pode precisar ser ajustada de modo que eles apontem para o Mixer. Novamente, isso é completamente visual, o tamanho e o layout dos Pipes não vão afetar o seu comportamento. Use a guia Layout para configurar as tubulações (Pipes) no seu modelo de modo que possa ficar da forma como você achar melhor. O conteúdo máximo padrão destas Pipes atualmente é muito grande para este modelo.

- Dê um duplo clique em *FluidPipe3* para abrir a janela **Properties**. Na guia **Pipe**, altere o **Maximum Content** para 10. Isto irá assegurar que o material que está deixando os Tanks leve pouco tempo para chegar ao Mixer.
- Repita estes passos para o *FluidPipe4*.

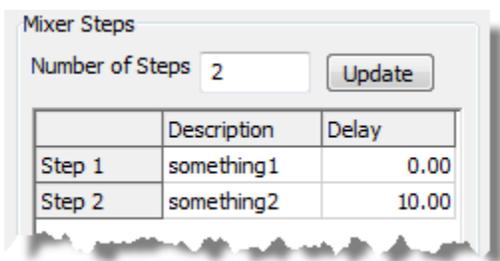
Etapa 8: Configure as etapas de produção do Mixer

O FluidMixer agora precisa ser configurado para receber os dois materiais diferentes e combiná-los em um novo material. Isso é feito alterando as duas tabelas encontradas na guia Steps na janela de Propriedades do FluidMixer. A tabela Mixer Steps é utilizado para definir uma série de passos que o Mixer deve percorrer para cada lote (batelada) que processa. Neste modelo, a tabela Mixer Steps precisa de dois passos. O Mixer deve puxar 10 litros do primeiro material a partir de uma porta de entrada 1, durante o passo 1. Em seguida, ele deve puxar 20 litros do segundo material a partir da porta de entrada 2

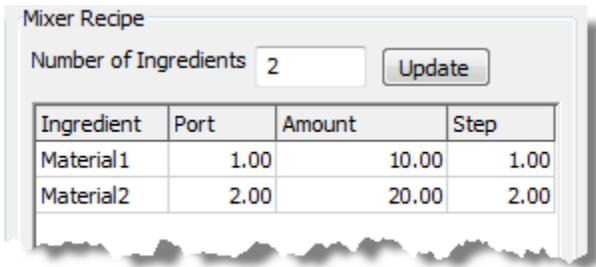
- Dê um duplo clique em *FluidMixer* para abrir a janela **Properties**, e clique na guia **Steps**.
- Na guia abaixo de **Mixer Steps**, mude o **Number of Steps** para 2.
- Clique em **Update**. Os steps (passos) irão aparecer na tabela.

A descrição dos passos (steps) não é importante, você pode chamá-los de qualquer coisa que você quiser. O delay time para cada passo é executado depois que todos os materiais de um determinado passo são recebidos, e antes do Mixer começar a receber o material para a próxima etapa ou passo (step).

- Configure **Delay** para o **Step 1** para 0, e o **Delay** do **Step 2** para 10.

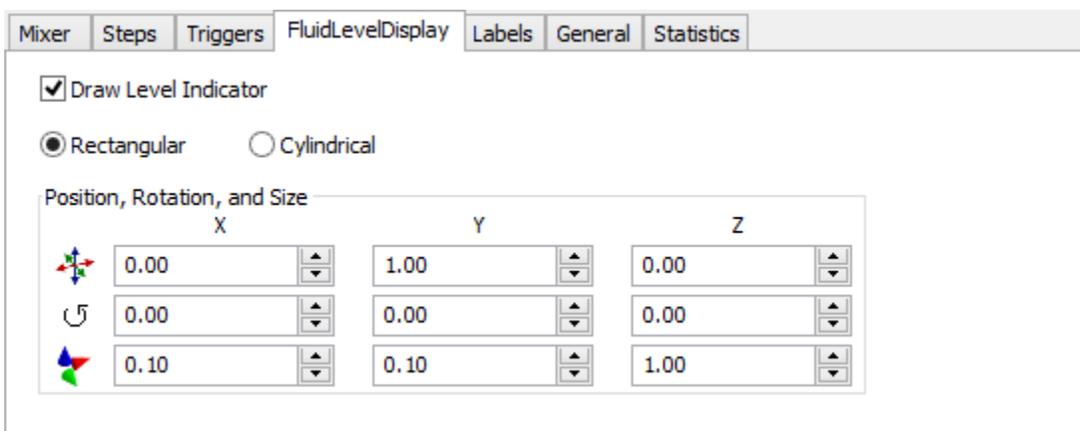


- Na guia **Steps**, guiaixo de **Mixer Recipe**, mude o **Number of Ingredients** para 2.
- Clique em **Update**. Os raw material ou ingredientes irão aparecer na tabela. Uma vez mais, a descrição pode ser qualquer coisa que você queira porque o objeto não se refere a estes nomes. Chame eles *Material1* e *Material2*.
- Para *Material1*, mude o **Port** para 1, e o **Amount** para 10, e o **Step** para 1.
- Para *Material2*, mude o **Port** para 2, e o **Amount** para 20, e o **Step** para 2.



O mostrador de nível do Mixer é uma ferramenta útil para visualizar o nível de fluido enquanto o modelo roda. Ele irá mostrar quanto de cada material no Recipe Table o Mixer tem recebido em qualquer ponto do tempo. Por padrão, fica Escondido atrás do Mixer's 3D shape. Mais informações sobre o nível do fluido pode ser encontrado nas propriedades dos objetos. Você pode também configurar o tamanho e o formato do visualizador do nível de fluídos para que possa parecer-se parte do objeto para ficar com uma visualização mais realística.

- Clique na guia **FluidLevelDisplay**. Mude o **Y** para 1.



Etapa 9: Verifique o FluidProcessor

Os valores padrões para o FluidProcessor irão funcionar bem para este modelo. Ele irá receber material da porta de entrada 1, processar o material por um certo período de tempo e enviá-lo para a porta de saída 1. O tempo que será gasto para processar o material está baseado em seu **Maximum Content** e no **Maximum Output Rate** que o modelador define nas propriedades do FluidProcessor na guia **FluidProcessor**. Você pode alterar estes valores para verificar como isso afetará o modelo. Para fazer o nível do fluido aparecer em uma barra, você precisará alterar o valor de **Y** em FluidLevelDisplay para 1, como realizado anteriormente para o FuidMixer

Etapa 10: Configure o FluidToltem

O FluidToltem próximo do final da linha de produção irá converter o material fluido que está vindo do FluidProcessor e alterá-lo para flowitems. O Fluid per Discrete Unit e Discrete Units per Flowitem serão

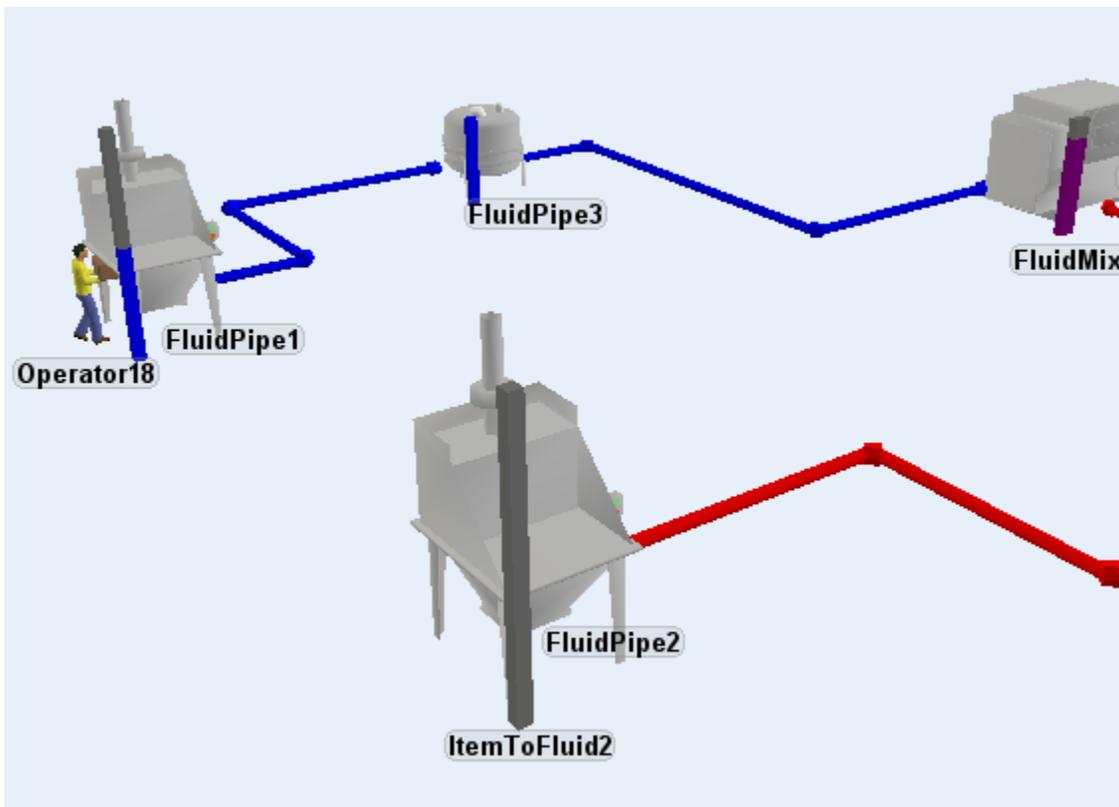
multiplicados juntos para determinar quanto material terá que ser coletado para criar um simples flowitem. Neste caso, 10 galões de fluido serão convertidos em 1 flowitem.

- Dê um duplo clique em *FluidToItem* para abrir a janela **Properties**.
- Na guia **FluidToItem**, abaixo de **Flowitem Output**, mude **Fluid per Discrete Unit** para 10.

Na guia **FluidToItem**, altere **Maximum Content** para 10. Isto irá falar ao objeto, que ele pode coletar apenas material suficiente para 01 flowitem a qualquer instante. Se este valor for maior, o *FluidToItem* poderá parar de se comportar como um Queue e criar muito mais espaço de armazenagem em seu modelo.

Etapa 11: Resete e execute o modelo.

- **Resete e Salve** o modelo.
- **Execute** o modelo. Você deverá ver as barras indicadoras de nível sobre os objetos subindo e descendo. Você também deve ver as tubulações (pipes) piscando e mudando de cor. Se a tubulação estiver em cinza, ele está vazio. Se for uma cor pulsante, o material está fluindo através da tubulação (pipe). Se for uma cor sólida, o material está bloqueado. Haverá também flowitems sendo transformada em fluido e fluido que está sendo transformado em flowitems



Depois de completar esta lição você deve ter uma idéia de como estes Fluid Objects trabalham e conhecerá algumas de suas capacidades. Há muito mais que se pode fazer do que foi aprendido nesta lição. Leia a documentação e tente outras opções e configurações. Logo você estará reunindo conhecimentos para desenvolver grandes modelos baseados em fluido, e mais detalhados.

Gráficos e Relatórios

1. Dashboard
2. Reports and Statistics (Relatórios e Estatísticas)

Dashboard

1. Concepts

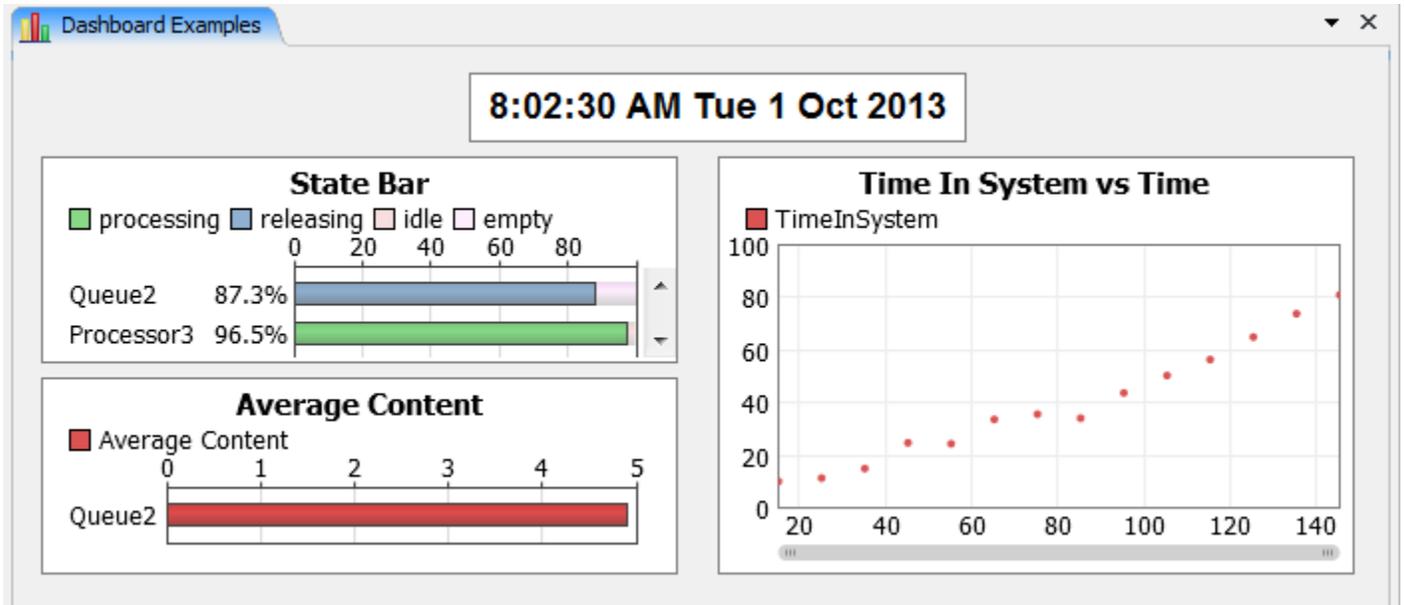
- Custom Chart
- Model Input

2. Example

3. Reference

- Graph Properties
- Model Input Properties

Conceitos do Dashboard

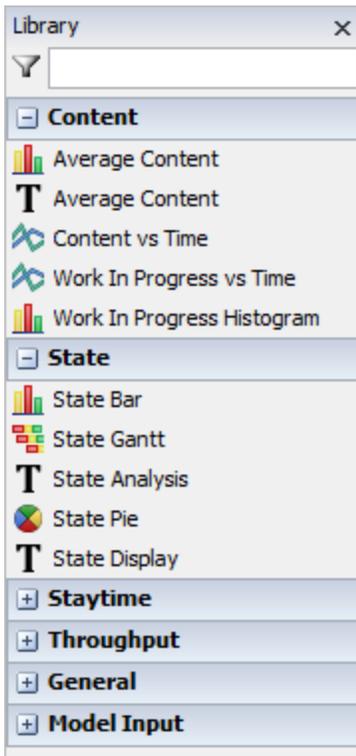


A janela do Dashboard lhe permite visualizar gráficos e estatísticas do modelo que lhe interessam conforme o modelo roda. Esta ferramenta é bastante útil para realizar comparações entre objetos lado-a-lado.

Para mais informações sobre a diferença dos tipos de gráficos, veja a página de referência.

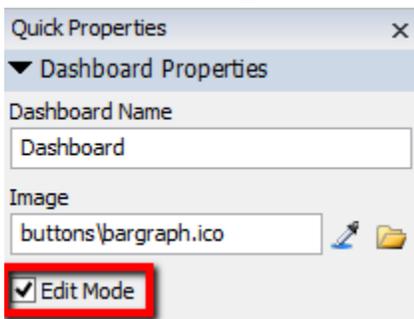
Nota: Nem todas estatísticas fazem sentido para todos objetos. Caso o objeto selecionado não tenha determinada estatística, o gráfico não será gerado para aquele objeto.

Dashboards are accessed from the Toolbox. (**View menu > Toolbox > Add > Statistics > Dashboard**)
A grade de biblioteca de objetos será alterada para mostrar todos os recursos de dashboards que podem ser criados.



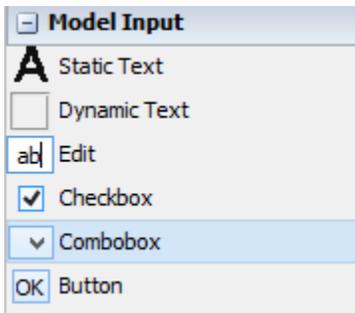
Tipos de Gráficos Dashboard: A lista de gráficos do Dashboard mostrado na grade de biblioteca de objetos (Library Icon Grid) são somente um ponto inicial, uma lista básica e pré-configurada. Os gráficos podem ser customizados para mostrar informações adicionais ao editar as propriedades dos gráficos.

Modo de Edição



Quando você cria pela primeira vez um Dashboard, ele estará no modo de edição. Isto permite a você movimentá-lo e redimensionar as janelas do Dashboard assim como abrir suas propriedades ao dar um duplo clique sobre a janela. Desmarcando o Edit mode, irá travar a janela e permitir você interagir com os objetos de dados de entrada do modelo.

Input do Modelo



Anteriormente, as interfaces controladas por usuários ou custom user interfaces/ view variables and parameters (variáveis de visualização e parâmetros) do modelo estavam somente disponíveis através do Graphical User Interfaces. Os Dashboards agora tem a capacidade de manusear muitas das mesmas interfaces que somente eram feitas com os GUIs.

Para mais informações sobre como trGuialha cada janela de Input do Modelo, veja a página de referência.

Creating and Arranging Widgets

To add a widget from the Library Icon Grid click and drag the widget to the dashboard window or left-click once on the desired widget to enter creation mode, then left-click again on the Dashboard to create a new widget. Right-click or press the Escape key to exit creation mode.

To arrange widgets, left-click on the widget you wish to move. Sizers will appear around the outside of the widget. Click and drag the widget to move it, or click and drag on one of the sizers to resize the widget.

Hold the Control key down and click on dashboard widgets to select multiple widgets.

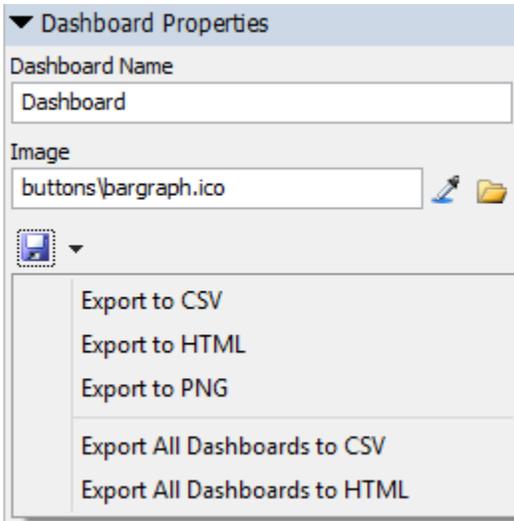
Arrow Keys - Arrow keys may be used to moved Dashboard widgets. Select the desired widget(s) then use the up, down, left or right arrow keys to move the widget(s) 1 pixel. Hold the shift key while pressing the arrow keys to move the widget(s) 5 pixels at a time.

Copy and Paste - Select a single or multiple model input widgets and press Ctrl + C to copy them. Press Ctrl + V to paste into the same Dashboard, another Dashboard in the same model, or Dashboards in other instances of FlexSim. This only works for Model Input widgets, not for Statistics Widgets.

Locking to an Edge

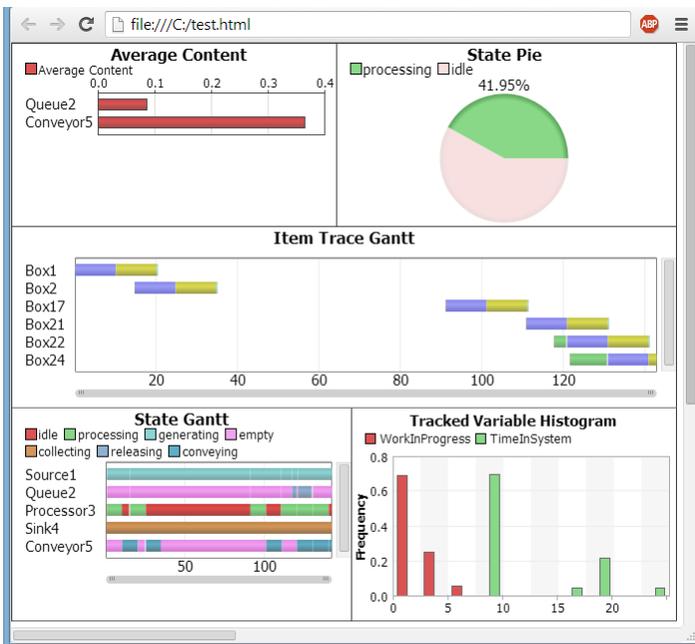
Widget can be "locked" to an edge of the dashboard window by click and dragging a sizer to the desired side of the window. Sides that are locked will show yellow sizers. If a widget is locked to the bottom or right side of the window, that widget will resize as the window is resized. Widgets locked to the top and left side of the window will cause the widget to resize so that the top and left sides of the widget are against the edge of the window, rather than a few pixels off. Once locked, moving a widget will cause it to resize in order to maintain its lock with the edge.

Exporting Dashboards



 - Exports dashboard to CSV, HTML or PNG formats. Saving CSV files will only save the Dashboard Data. Saving to HTML will save the Dashboard with its layout to an HTML file that allows you to move, resize and explore the widgets (see image below). Choosing Export All Dashboards to HTML will create one HTML file with all dashboard widgets contained in it. These HTML files are completely self contained and may be opened on any computer with a web browser (though some web browsers may not be supported). After resizing and arranging widgets in the HTML view, the file may be saved again to preserve the layout. Individual widgets may be exported to these formats as well by right-clicking on the widget.

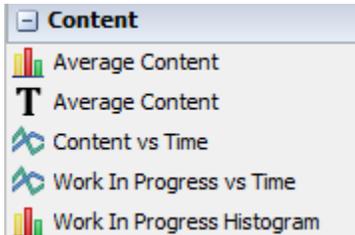
Exporting Model Inputs: Model Input objects, like buttons and fields, will not be exported using the Dashboard's Export to HTML. Only when exporting as PNG.



Tipos de Gráficos

Dashboard Graph Types: A lista de gráficos do Dashboard mostrado na grade de biblioteca de objetos (Library Icon Grid) são somente um ponto inicial, uma lista básica e pré-configurada. Os gráficos podem ser customizados para mostrar informações adicionais ao editar as propriedades dos gráficos.

Content



Average Content (Gráfico) - Mostra o conteúdo médio dos objetos selecionados em um gráfico de barra.

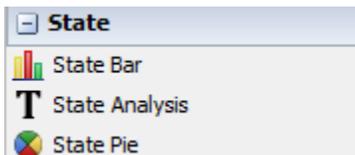
Average Content (Texto) - Mostra o conteúdo médio dos objetos selecionados em uma tabela de valores.

Content vs Time - Plota um gráfico com o conteúdo do objeto selecionado vs o tempo de simulação.

Work In Progress vs Time - Mostra um gráfico de linha da variação do número total de itens no modelo vs o tempo de simulação.

Work In Progress Histogram - Mostra um histograma da variável Work In Progress.

State

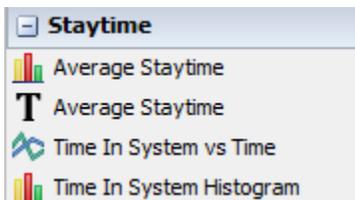


State Bar - Mostra o estado dos objetos selecionados através de valores percentuais.

State Analysis - Mostra a porcentagem do tempo total que que o objeto permaneceu em um estado específico. Cada estado é mostrado em uma coluna diferente.

State Pie - Mostra um gráfico de pizza relacionando o estados dos objetos selecionados. Quando mais de um objeto é selecionado, o gráfico de pizza mostra a média percentual de cada estado.

Staytime



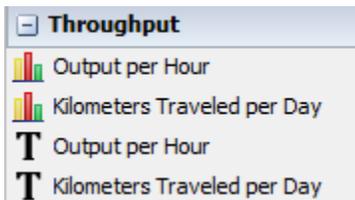
Average Staytime (Graph) - Mostra o tempo médio de permanência dos itens no objeto selecionado em um gráfico de barras.

Average Staytime (Text) - Mostra o tempo médio de permanência dos itens no objeto selecionado em uma tabela de valores.

Time In System vs Time - Plota um gráfico do tempo de permanência do último item destruído no modelo vs o tempo de simulação.

Time In System Histogram - Mostra um histograma da variável Time In System.

Throughput



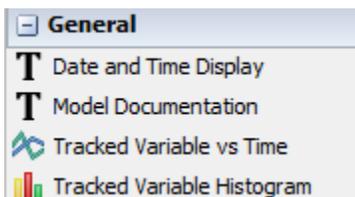
Output per Hour (Graph) - Mostra a taxa de saída por hora dos itens de cada objeto em um gráfico.

Kilometers Traveled per Day (Graph) - Mostra a distância média diária que um objeto se desloca em um gráfico de barras (distância e unidade de tempo podem ser alteradas).

Output per Hour (Text) - Mostra a taxa de saída por hora dos itens de cada objeto em uma tabela de valores.

Kilometers Traveled per Day (Text) - Mostra a distância média diária que um objeto se desloca em uma tabela de valores (distância e unidade de tempo podem ser alteradas).

General



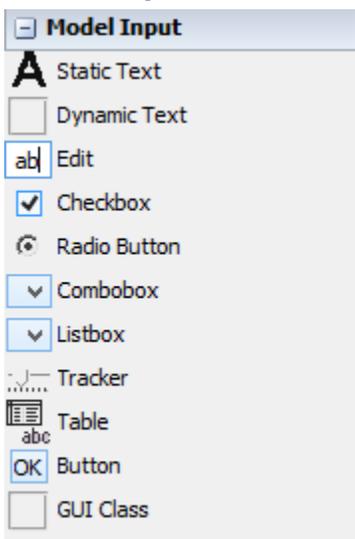
Date and Time Display - Mostra o tempo atual do modelo como um tempo do dia, a data do modelo e o tempo (conforme definido em configurações do modelo) ou formato customizado.

Model Documentation - Mostra um documento em HTML contendo a documentação do modelo. Este painel permite a você imputar códigos flexscript para mostrar de forma customizada.

Tracked Variable vs Time - Mostra uma variável Tracked definida pelo usuário que é plotado em função do tempo.

Tracked Variable Histogram - Mostra um histograma de Tracked Variable definida pelo usuário.

Model Input



Model Input controls allow you to have interaction between the user and the model without the need for making separate Graphical User Interfaces. Each control has customizable properties that can be set through the Quick Properties window. Each control can have its own ID string that can be used to reference the control using `getdashboardcontrol("idName")`.

Static Text – Mostra um texto estatístico ou imagem.

Dynamic Text - Este campo pode ser linkado com uma variável ou parâmetro no modelo e será atualizado conforme os valores linkados se alteram, exemplo, linkando com uma label sobre um objeto, ou linkando com dados de entrada ou de saída de um objeto. O valor pode ser uma string ou dados numéricos.

Edit - Similar ao Dynamic Text, um campo Edit permite ao usuário linkar para uma variável ou parâmetro no modelo e alterar o valor, exemplo linkando a capacidade máxima de um queue permitirá ao usuário alterar a capacidade máxima sem editar os parâmetros do Queue diretamente.

Checkbox - Um checkbox pode ser linkado com variáveis ou parâmetros no modelo para mostrar o seu valor atual e permitir ao usuário trocar aquele valor. Checkboxes são muito úteis para valores booleanos, onde os valores esperados são 1 ou 0. O código pode também ser executado quando um checkbox é pressionado ao adicionar código para sua trigger OnPress.

Radio Button

The Radio Button control can be linked to a node in the model and it can be linked to other radio buttons. To attach to a group of radio buttons, click on the Link  and then click on another radio button in the same Dashboard. This will "Attach" the radio button to the group. Any number of radio buttons may be added to a group. Each radio button in the group has a designated value. If the parent radio button is linked to a node in the model, that node will be set to the value of the selected radio button. You can also add code to the Radio Button's OnPress trigger.

Combobox - Combobox pode também ser linkado a uma variável ou parâmetro no modelo. Qualquer número de opções pode ser adicionado para a listagem drop down. A trigger OnSelect é então ativada quando o usuário seleciona uma das opções fornecidas. O código podem então ser adicionado para esta trigger para alterar os aspectos do modelo.

Listbox

The Listbox control can be linked to a label or node with number or string data in the model. If the node is string data then the name of the listbox item will be set to the node. You can also add code to the Listbox's OnSelect trigger.

Tracker

The Tracker control can be linked to a label or node with number data in the model. Tracker's have a set of additional parameters that can be set including the Minimum, Maximum and Exponential values as well as the Style and if it's a vertical or horizontal tracker. You can also add code to the Tracker's OnDrag trigger.

Table

The Table control can be linked to any table in the model. The table will display the linked table's current values. Editing the table will update the linked values directly.

Button - Buttons podem também ser linkados para variáveis ou parâmetros no modelo. Linkando os button irá alterar os nomes dos botões que mostrarão os valores daquela variável ou parâmetro. Quando o botão é pressionado, a trigger OnPress é disparada, permitindo ao usuário adicionar código customizado em seu botão (button).

GUI Class

For more advanced functionality, the GUI Class object allows you to build a GUI using the GUI Builder and then point the GUI class to the GUI. Changes made in the GUI Builder will be updated in the Dashboard when the widget is refreshed. You can refresh the widget either by closing and reopening the Dashboard, or by right clicking on the widget and selecting *Refresh*.

Using paths with the GUI Class - If your GUI created in the GUI Builder and then referenced with a GUI Class widget has nodes being referenced using the @ symbol (ie @>objectfocus+>variables/maxcontent), the returned paths will not be correct in the Dashboard. The @ symbol moves up the tree until it finds the top most object so it will move up through the Dashboard and the other panels in the view.

Exemplos de Gráficos

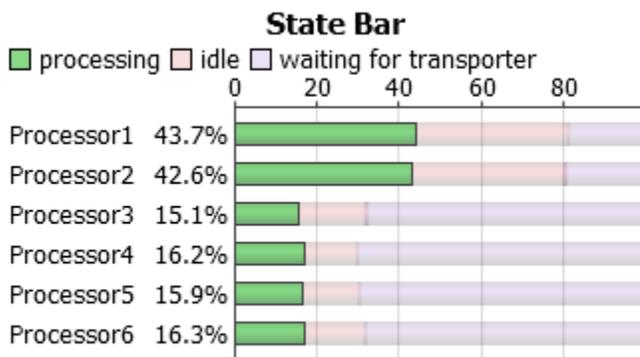
Dashboard Graph Types: The list of Dashboard Graphs displayed in the Library Icon Grid are just a starting point, a list of presets. Graphs may be customized to display additional information by editing the Graph's properties.

Text Panel

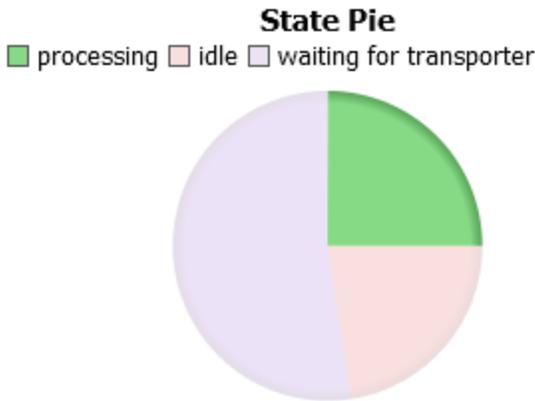
	State Analysis			
	Total	processing	idle	waiting for transporter
Processor1	43.9%	43.9%	36.5%	19.6%
Processor2	41.3%	41.3%	39%	19.7%
Processor3	15.5%	15.5%	14.1%	70.4%
Processor4	15.7%	15.7%	15.5%	68.8%
Processor5	16.2%	16.2%	15.5%	68.3%
Processor6	16.4%	16.4%	18.6%	65%

Displays text data in a table format. This panel can be used to display state values, statistics and custom data.

Bar Chart

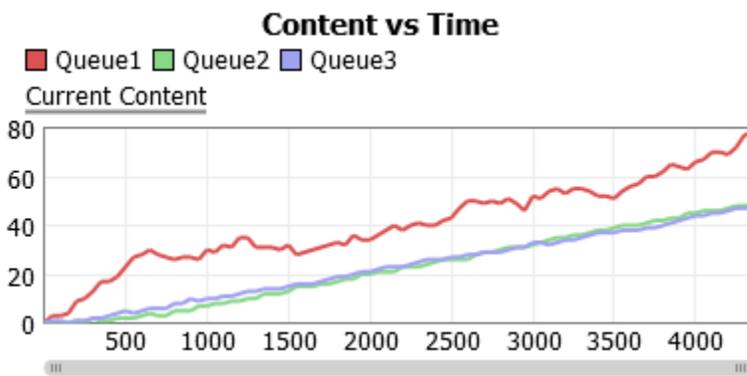


Pie Chart



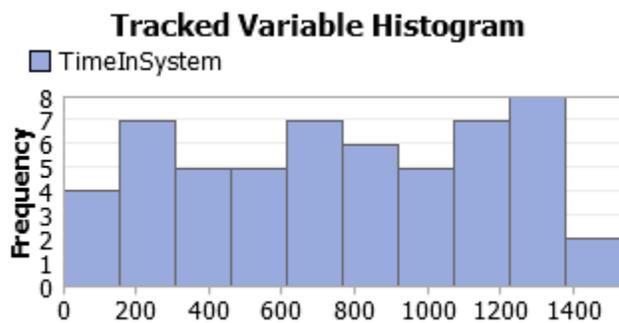
Pie Charts are only available when recording object State data.

Line Graph



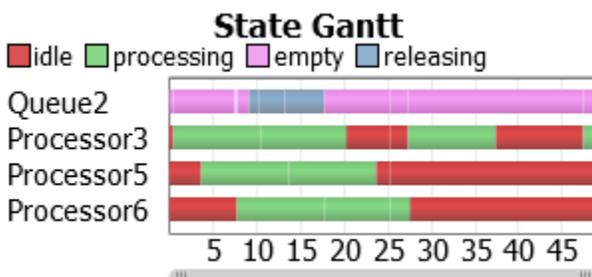
The Line Graph displays data over time. Specify a start time and a time interval between updates.

Histogram



The Histogram is only available for Tracked Variables.

Gantt Chart



There are two types of Gantt Charts:

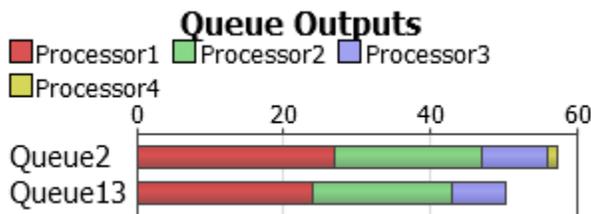
- **State Gantt Chart (displayed above)** - The State Gantt chart shows a time graph of what times and object was in a specific state. State changes in 0 time are not recorded.
- **Item Trace Gantt Chart** - shows what time an item is created and which objects the item travelled through and at what times. Item Trace charts record data even when an item is in an object for 0 time, however, that data is not graphically displayed on the chart. Export the chart's data to CSV to see the full history.

Financial Analysis

Financial Analysis	
▷ Totals	\$20,696.00
Fixed	(\$6,100.00)
Time	\$0.00
State Fixed	\$0.00
State Time	(\$616.00)
Flowitems Fixed	\$27,720.00
Flowitems Time	(\$308.00)

The Financial Analysis graph allows you to specify financial values for objects and flowitems in your model. Individual objects and groups may be added to this graph any number of times. Positive and negative values may be defined for each value. Negative values will be displayed in red surrounded by parentheses.

Custom Chart



The Custom Chart allows you to graph any kind of numeric data in either a table of values, bar chart, or line graph. Rather than adding only objects to this chart, you can add objects, nodes, tables, global variables, bundles etc. For more information, see the Custom Chart page.

Date and Time Display

8:50:30 AM Monday 01/27/14

The Date and Time Display graph can display the model's current run date and time in multiple formats:

- **No Format** - Displays the current model time beginning from Day 1 at the model start time (as defined in the Model Settings window).
- **Use Default Format** - Displays the current model time using the format defined in the Model Settings window.
- **Custom Format** - This option allows you to define a custom format for the date (displayed above).

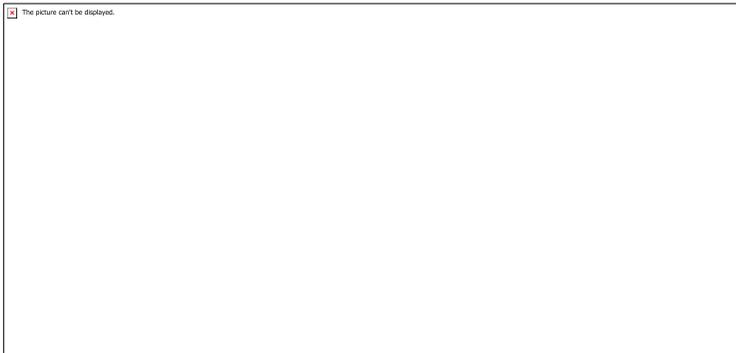
Model Documentation

Model Documentation

This model shows how adding another welding machine will reduce bottlenecks.

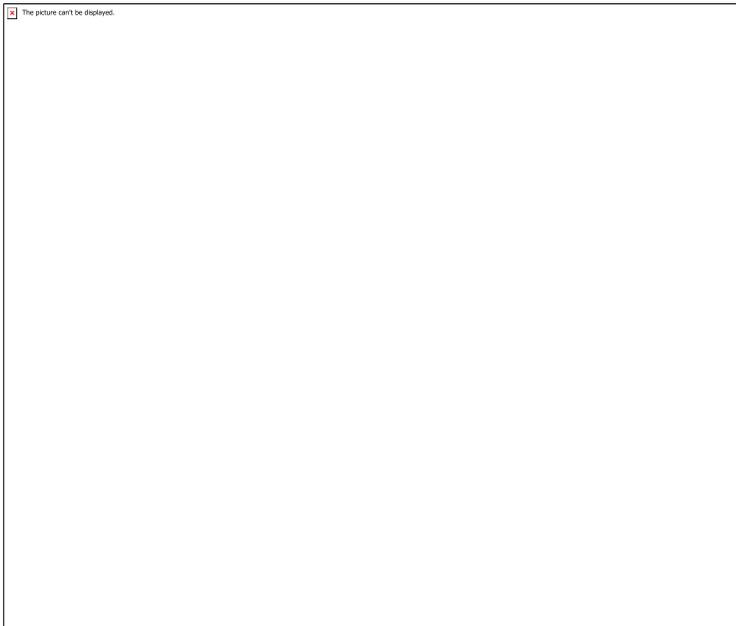
The Model Documentation graph is a custom HTML widget that gives you a blank canvas to add in any custom HTML or flexscript code. Adding flexscript code, you can display any information you want using the `pd()`, `pf()`, `pt()` and `pr()` commands. These commands will create HTML code to display on the graph.

Associations



Custom charts are tied to associations, rather than restricted to only objects as with other Dashboard Widgets. An association can be a node in the tree, a global variable, an object, global table, label table, etc. Associations can be added to the custom chart any number of times, however, the first association is what will be used to specify the number of series to display (see # Series below).

Displaying Data



There are 5 picklists that allow you to specify how many series and how many categories are displayed in the chart. Think of the series as columns in a table, and the categories as rows in a table. All picklists are fired each time the graph is drawn, or for a time graph, each update time. This allows the number of series

and categories to be dynamically changed, ie a global table that has columns or rows added to it. A list of presets is available for displaying current values, tables, etc.

Global Variables: Global Variables that are added to the custom chart to be accessed as an association must be either an integer or double type. When using the # Series, # Categories or Data Point Value picklists with global variables, use `getnodenum(current)` to get the current value of the global variable. This allows you to tie to multiple global variables and other nodes and display all of their data in the same chart. Any other reference to current will give you the global variable's node, ie `/Tools/GlobalVariables/myVariable`.

Other global variables may be used directly, however, the special functionality of using `getnodenum(current)` will only return valid data for integer and double types.

Series

The number of series is only calculated once for all associations. This picklist is the first to be fired and passes in the first association object/node. The value returned should be an integer.

Access Variables

- `current`: the first association object/node in the list.

Series Title

This picklist is fired for each series. These titles will display in the charts legend. The value returned should be a string.

Access Variables

- `current`: the first association object/node in the list.
- `seriesnum`: the index of the series.

Categories

The number of categories is calculated for each association in the list. Each category will be a row in the chart, beginning with the first association, in order, down to the last. The value returned should be an integer.

Access Variables

- `current`: the association object/node.

Category Title

This picklist is fired for each category, for each association. If two or more categories who are sequential (next to each other) have the same title, those categories will become a group. By default, a group's total values are calculated as an average of all of its members. To calculate the total as a sum, the title must have a `#sum` at the end. ie, `"Total#sum"`. When using groups, the default titles displayed for the group members will be their path in the model. To change this to a custom display, set a Display Name for the association in the Associations tab. The value returned should be a string.

Access Variables

- `current`: the association object/node.
- `categorynum`: the index of the category for the association.

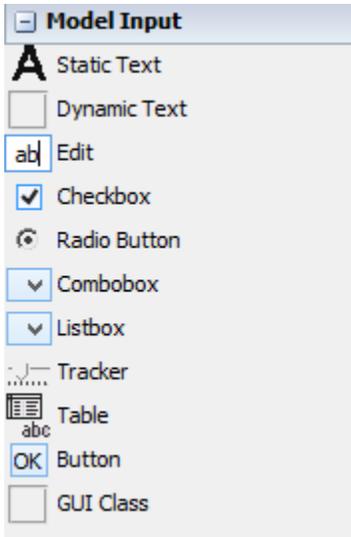
Data Point Value

This picklist is fired for each association, for each category, for each series. The value returned should be a number.

Access Variables

- current: the association object/node.
- seriesnum: the index of the series.
- categorynum: the index of the category for the association.

Model Input



Model Input controls allow you to have interaction between the user and the model without the need for making separate Graphical User Interfaces. Each control has customizable properties that can be set through the Quick Properties window. Each control can have its own ID string that can be used to reference the control using `getdashboardcontrol("idName")`.

Static Text

The Static Text control can be used to either display text, or to display an image.

Dynamic Text

The Dynamic Text control can be linked to a variable, label or node in the model and will display its current value. This can include number data or string data. Dynamic Text is not editable and is for display only.

Edit

The Edit control is similar to the Dynamic Text control but it can also be edited. Changing the value of the Edit will also change the value of the node in the model that it is linked to. ie, linking to the Max Content of a Queue would allow the user to change the Max Content without editing the Queue's parameters directly.

Checkbox

The Checkbox control can be linked to a variable or node in the model. The node will be 'evaluated' as a boolean (true or false, 1 or 0), so linking to a non-number node will have no affect. You can also add code to the Checkbox's OnPress trigger.

Radio Button

The Radio Button control can be linked to a node in the model and it can be linked to other radio buttons. To attach to a group of radio buttons, click on the Link  and then click on another radio button in the same Dashboard. This will "Attach" the radio button to the group. Any number of radio buttons may be added to a group. Each radio button in the group has a designated value. If the parent radio button is linked to a node in the model, that node will be set to the value of the selected radio button. You can also add code to the Radio Button's OnPress trigger.

Combobox

The Combobox control can be linked to a variable, label or node with number data. Any number of options can be added to the drop down list. You can also add code to the Combobox's OnSelect trigger.

Listbox

The Listbox control can be linked to a label or node with number or string data in the model. If the node is string data then the name of the listbox item will be set to the node. You can also add code to the Listbox's OnSelect trigger.

Tracker

The Tracker control can be linked to a label or node with number data in the model. Tracker's have a set of additional parameters that can be set including the Minimum, Maximum and Exponential values as well as the Style and if it's a vertical or horizontal tracker. You can also add code to the Tracker's OnDrag trigger.

Table

The Table control can be linked to any table in the model. The table will display the linked table's current values. Editing the table will update the linked values directly.

Button

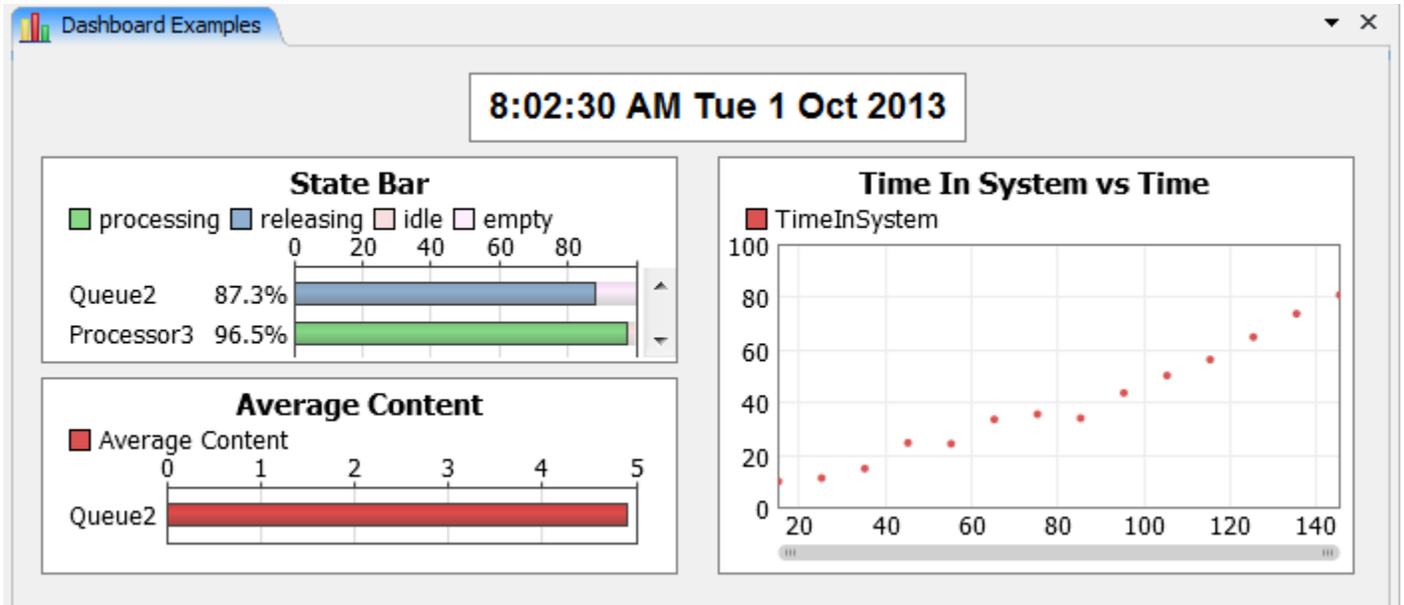
The Button control can be linked to a node in the model. The button can display a text title, or it can display an image. If linked to a node, the button will display the current value of the node. You can also add code to the Button's OnPress trigger.

GUI Class

For more advanced functionality, the GUI Class object allows you to build a GUI using the GUI Builder and then point the GUI class to the GUI. Changes made in the GUI Builder will be updated in the Dashboard when the widget is refreshed. You can refresh the widget either by closing and reopening the Dashboard, or by right clicking on the widget and selecting *Refresh*.

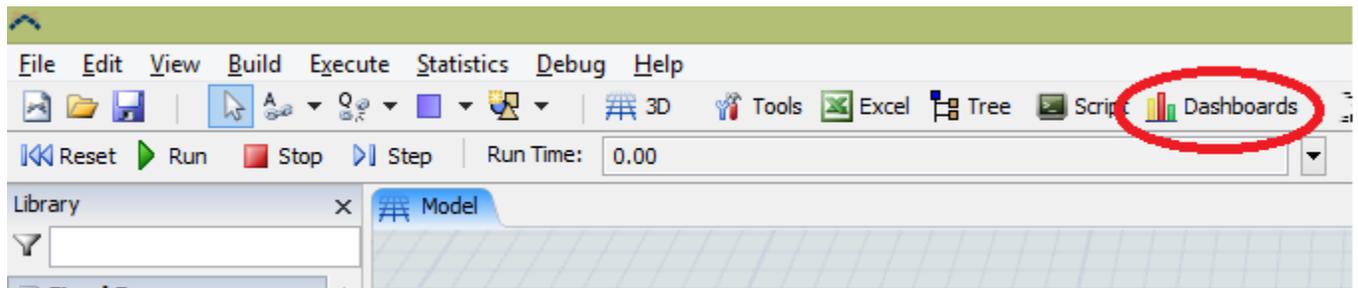
Using paths with the GUI Class - If your GUI created in the GUI Builder and then referenced with a GUI Class widget has nodes being referenced using the @ symbol (ie @>objectfocus+>variables/maxcontent), the returned paths will not be correct in the Dashboard. The @ symbol moves up the tree until it finds the top most object so it will move up through the Dashboard and the other panels in the view.

Exemplo de Dashboard

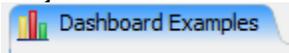


Adicionando Gráficos

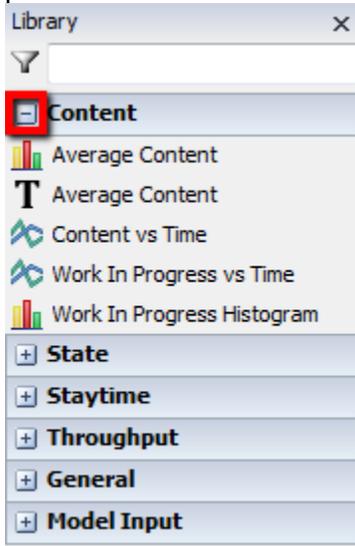
1. Adicione um novo Dashboard ao modelo **no ícone Tools** conforme imagem abaixo.



2. Se já existir um Dashboard sendo mostrado, assegure-se que a guia Dashboard está selecionado.

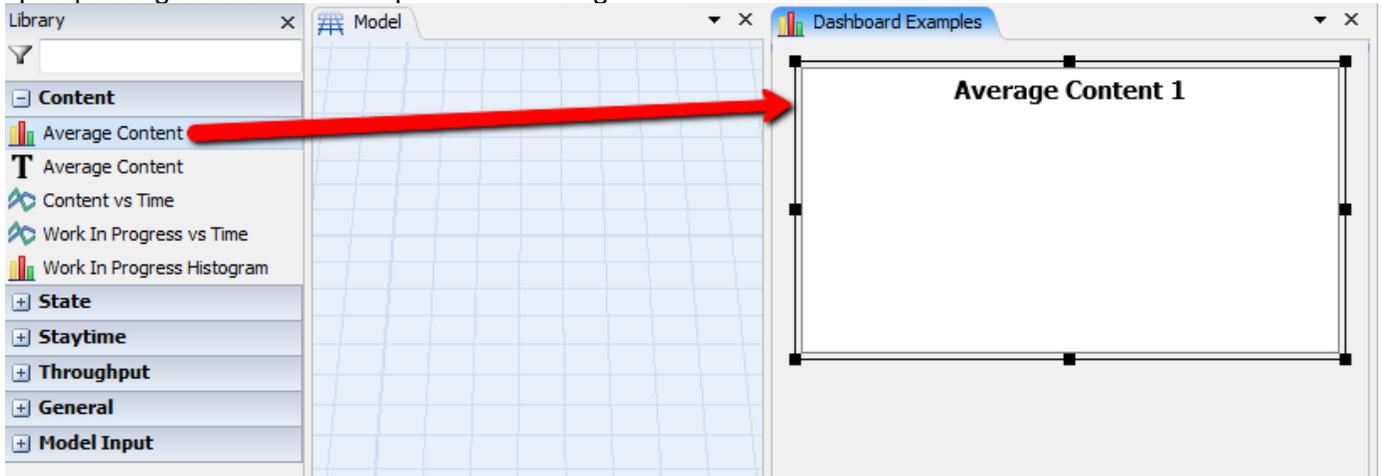


3. Se não estiver expandido por padrão, expanda o tipo de gráfico necessário da biblioteca (Library). Você pode também recolher qualquer tipo de gráfico que não seja mais necessário.



4. **Escolha uma estatística**

Clique e arraste uma das opções do menu para a área do Dashboard e selecione uma das opções em qualquer lugar no Dashboard para mostrar o gráfico.



5.

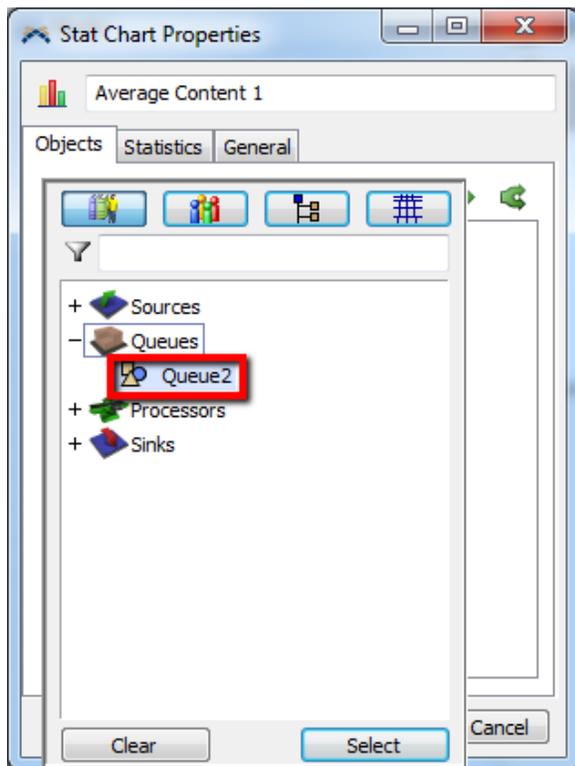
Model Edit Modes: Se você clicou uma vez sobre o gráfico na grade de biblioteca de objetos (Library Icon Grid), você entrará no modo de edição (Model Edit Mode). Isto permitirá a você criar vários gráficos ao clicar no Dashboard View. Para sair do modo de edição, clique com o botão direito do mouse na superfície de simulação ou pressione a tecla Escape.

6. **Adicione Objetos**

O menu de propriedades dos gráficos deve automaticamente aparecer. Clique sobre o botão **+** e selecione qual objeto você incluir neste gráfico. Você pode selecionar vários objetos através desta janela.

Clique sobre o botão Select e então pressione OK.

Você pode sempre dar um duplo clique sobre os gráficos para reabrir suas propriedades.



7. Resete e Execute o Modelo

O novo gráfico estará em branco até que o modelo inicie a rodar.

Dashboard Reference

Tópicos

- **Trabalhando com gráficos**
- **Tipos de Gráficos**
- **Exemplos**

Trabalhando com Gráficos

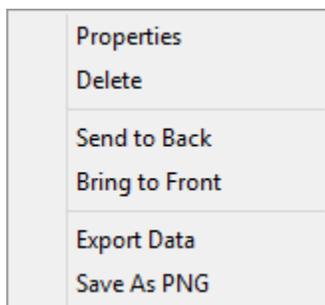
Editando – Com um duplo clique no gráfico, sua janela de propriedades se abrirá. O tipo da janela que se abrirá depende diretamente do tipo de gráfico selecionado. Para maiores informações veja Gráficos do Dashboard.

Movendo – Clique no gráfico, e arraste-o através das linhas pretas que ficam ao entorno da janela.

Redimensionando – Clique no gráfico, e faça os ajustes do tamanho através dos quadrados pretos que estão em volta do gráfico.

Auto-redimensionamento – O redimensionamento automático do gráfico no Dashboard consiste simplesmente em arrastar as bordas do gráfico até as duas extremidades da janela. Desta maneira, quando o Dashboard é redimensionado, o gráfico também sofre com as alterações de tamanho.

Contexto do Menu – Clique com o botão direito para abrir o menu do gráfico.

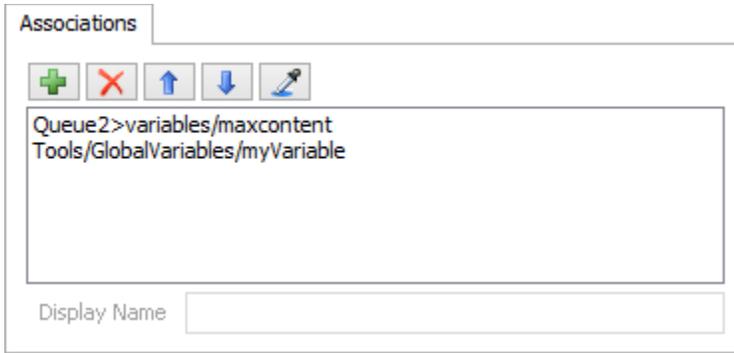


- **Properties** – Esta opção abre a janela de propriedades do gráfico.
- **Delete** – Esta opção deleta o gráfico do Dashboard.
- **Send to Back** – Esta opção envia o gráfico selecionado para trás dos outros objetos do Dashboard.
- **Bring to Front** – Esta opção leva o gráfico selecionado à frente dos outros objetos do Dashboard.
- **Export Data** - Esta opção permite a você salvar os dados atuais do gráfico como um arquivo .csv.
- **Save As PNG** - Esta opção permite você salvar um arquivo imagem .png do gráfico atual sendo mostrado.

Gráficos do Dashboard

1. Associations
2. Colors Page
3. Data
4. Data e Mostra do Tempo
5. HTML Statistic (Documentação do Modelo)
6. Item Trace Page
7. Objects Page
8. Statistic Page
9. Tracked Variable
10. Página de Análise de Utilização

Associations Page



Associations are objects, nodes, or variables in your model.

-  - Adds associations to the current list of associations. See the Adding Associations section for more info.
-  - Removes the selected associations from the list.
-  - Moves the selected associations up in the list
-  - Moves the selected associations down in the list
-  - Sample a node or object in the model.

Display Name - When using groups, if no display name is specified, the titles displayed for the group members will be their path in the model.

Adding Associations

There are multiple ways to add associations to the custom chart.

Tree Browse Dialog

For more information on the tree browse dialog, see the tree browse dialog page.

Global Variables / Tables

The list of Global Variables and Global Tables will be populated based upon what is available in your model. The list of global variables will only include integer and double types.

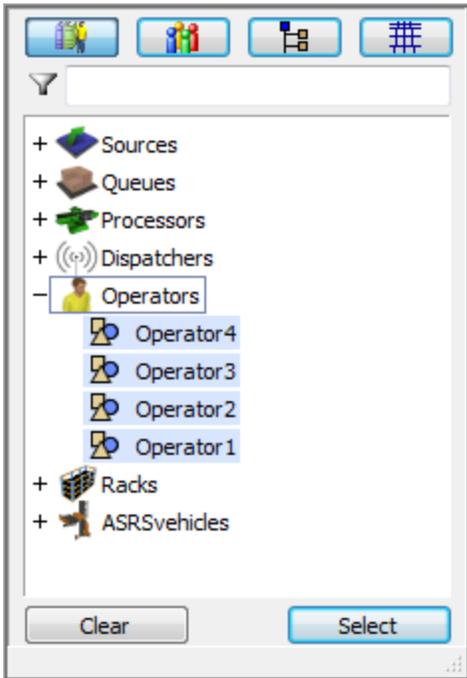
Sampler

For more information on using the Sampler to select objects and nodes, see the Sampler page.

Object Selection Window

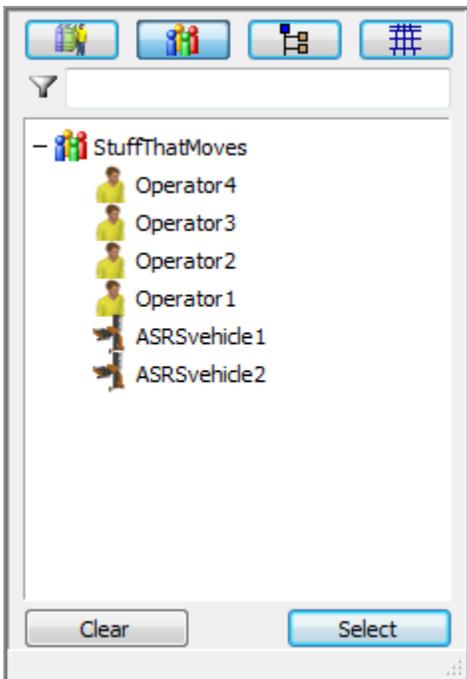
The current mode is highlighted at the top of the browsing window.

Browse by Class



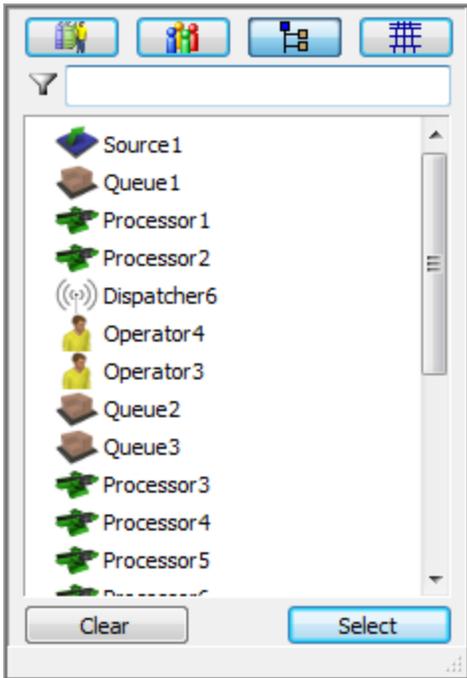
This method of adding objects sorts the objects by class. To select an entire class, click on the type icon. Click on an object to select or deselect it.

Browse by Group



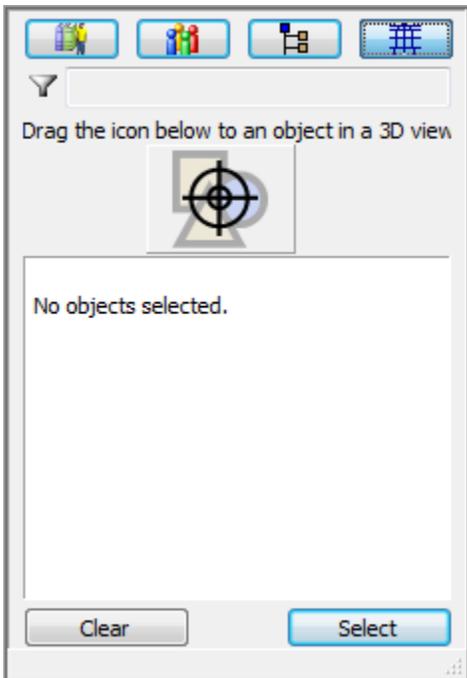
This method of adding objects sorts the objects by group. To select an entire group, click on the type icon. Click on an object to select or deselect it.

Browse by Object



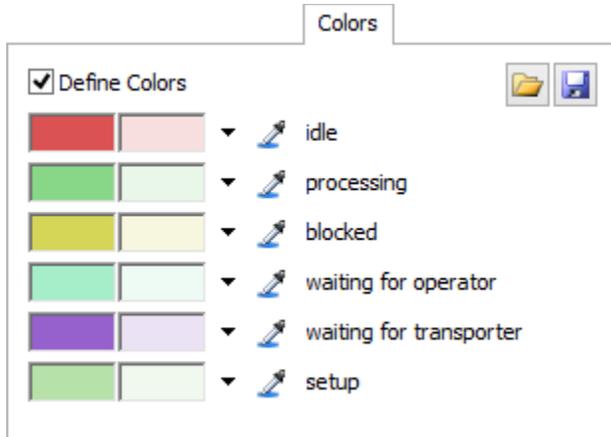
This method of adding objects lists all objects in the model. Click on an object to select or deselect it.

Select by Dragging



This method of adding objects uses the 3D model view. Simply drag the target icon from the current window to an object in the model you would like to add.

Colors Page



The colors page allows you to define colors for your graph. The options available are based upon the graph. For State charts, each State has a color. Notice in the above image there are two colors displayed. The color on the left is the normal color for that state. The color on the right is the translucent color used when *Show Yellow Checked States as Translucent* is checked from the Utilization Analysis page. For statistic graphs (Avg. Content, etc), the colors are for each statistic. Line charts define colors for each object or group. Gantt Charts either display colors for States or for objects.

Colors are not available for Text graphs.

Define Colors - Check to define a custom set of colors.

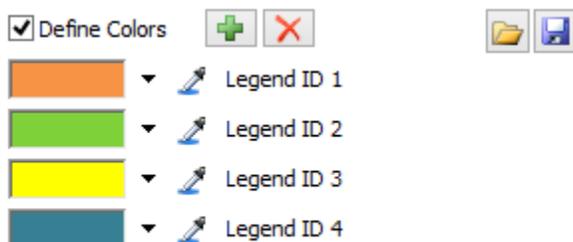
 - Loads saved color schemes and allows you to edit color schemes in the tree.

 - Saves the current color scheme.

▼ - Displays a color palette.

 - Allows you to sample a color in FlexSim or outside of FlexSim.

Item Trace Gantt and Custom Chart



The Item Trace Gantt Chart and the Custom Chart dynamically add Legends or Series to their charts so a  and  are available to add and remove colors.

Data Page

Data

Apply Preset ▾

Series  

Series Title 

Categories  

Category Title 

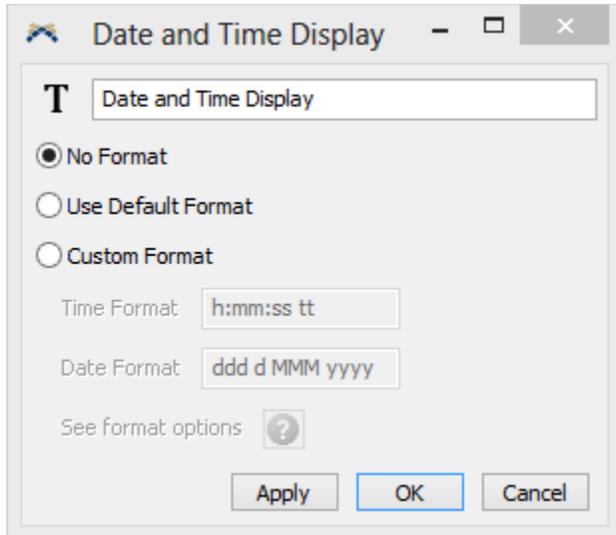
Data Point Value  

The data table is used by the Custom Chart. There are five picklists that allow you to customize the data to be displayed in a table of values, bar chart or line graph.

Apply Preset - This button contains a list of presets to get you started on using the custom chart. The presets include code to access current node values, table values, bundle values, and getting object input/output values.

See the Custom Chart page for more information on the picklists.

Display de Data e Tempo



Name Field - Altera o nome do painel para um nome específico.

No Format - Exibe a hora atual modelo decorrido desde o dia 1 a partir das 8:00 no formato: **Day 1, 08:00:00**

Use Default Format Mostra a data e horário do modelo atual conforme definido na janela Model Settings.

Custom Format - Mostra a data do modelo atual e a hora usando o formato customizado.

Financial Objects Page

Objects

Processor 1

Display Name

Object Values

Fixed 0.0000 Time 0.0000

FlowItems

Item Type(s)	Fixed	Time
	0.0000	0.0000

States Default State Profile

State	Fixed	Time
idle	0.0000	0.0000

- Adds objects/groups to the current list of objects. Objects and groups may be added multiple times.

- Removes objects/groups from the current list of objects

- Moves the selected object up in the list

- Moves the selected object down in the list

- Sample an object in the model.

Display Name - Enter text to display on the graph. Leave this field blank to display the object or group name.

Object Values

All object values can be negative or positive numbers. Negative numbers are shown in red and surrounded by parentheses.

Fixed - This value is added once when the model is reset. This value might represent the initial purchase price of a resource.

Time - This value is continually added as the model runs. A value of 1.0 would mean at time 50.0 the total time value for the object would be 50.0.

Use the to convert units to model time units.

FlowItems - Use the and to add or remove items from the table. Each item has an Item Type(s), Fixed and Time. To apply the values to all ItemTypes, leave the field blank. To define multiple ItemTypes, separate numbers using commas and dashes. For example: 1,2,5,10-15,20

- Fixed value is applied when a FlowItem of the specified ItemType enters the object.
- Time value is applied continually while the flowitem is in the object.

Use the  to convert units to model time units.

States - Use the  and  to add or remove items from the table. Each item has a State, Fixed and Time.

- Fixed value is applied when the object enters the specified state. If an object is in the state for 0 time, the fixed value will NOT be applied.
- Time value is applied continually while the object is in the specified state.

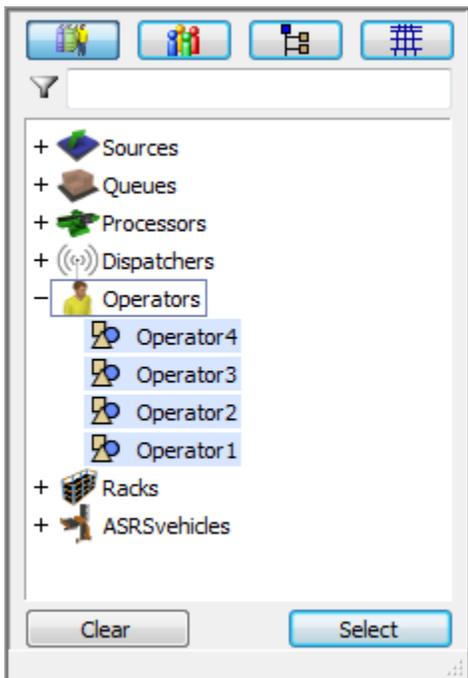
Use the  to convert units to model time units.

Adding Objects

There are five ways to add objects to a graph. For more information on using the Sampler to select objects, see the Sampler page.

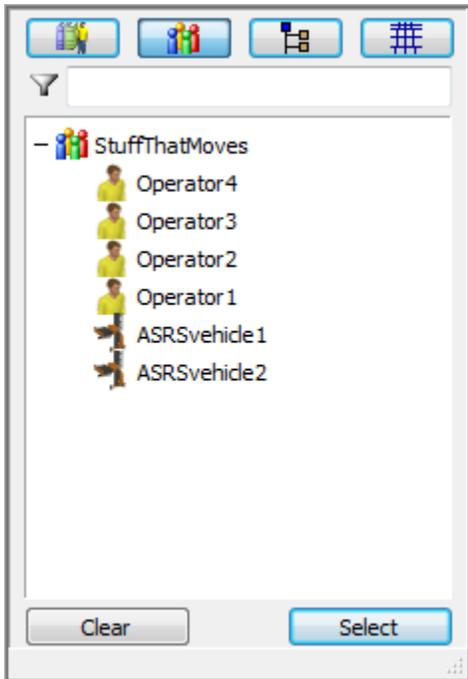
The current mode is highlighted at the top of the browsing window.

Browse by Class



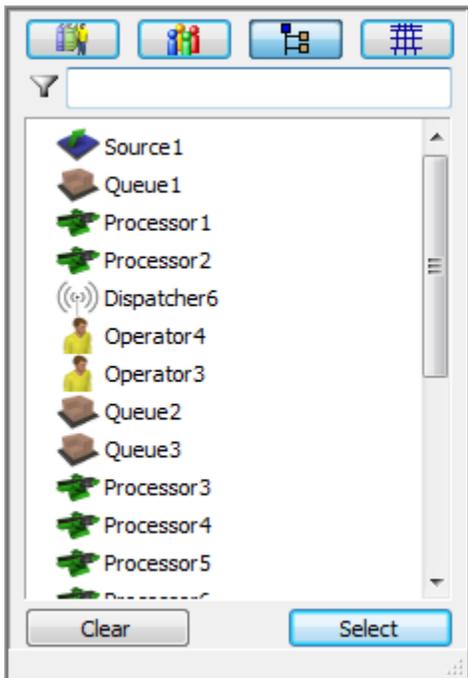
This method of adding objects sorts the objects by class. To select an entire class, click on the type icon. Click on an object to select or deselect it.

Browse by Group



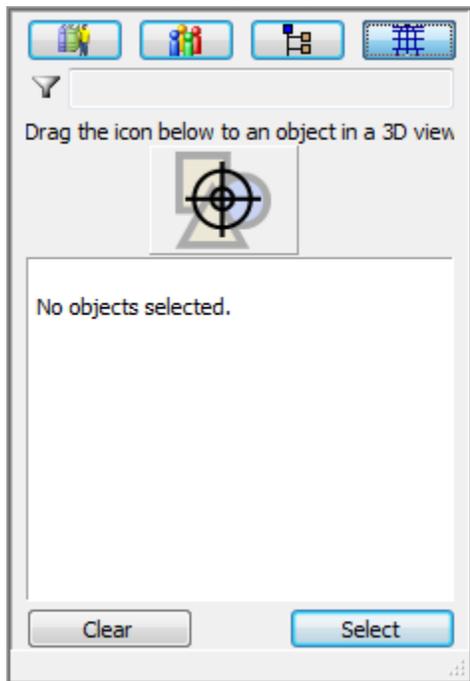
This method of adding objects sorts the objects by group. To select an entire group, click on the type icon. Click on an object to select or deselect it.

Browse by Object



This method of adding objects lists all objects in the model. Click on an object to select or deselect it.

Select by Dragging



This method of adding objects uses the 3D model view. Simply drag the target icon from the current window to an object in the model you would like to add.

Página General

The General page of Dashboard graphs vary depending on the statistic object and upon the display type.

Graph Types

- **Line Chart**
- **State Statistic Graphs**
- **Financial Analysis**
- **Gantt Chart**
- **Tracked Variable Graphs**

Line Chart

General

Display Type **Line Chart** ▼

Stacked Bar Chart

Show Legend

Font Size Bar Size

Only Collect Data for a Defined Time Interval

From Time To Time

Start Time Interval

Time Scale **Seconds** ▼

Y Axis Title X Axis Title

Display Type – Esta opção altera o estilo do gráfico.

Stacked Bar Chart – Esta opção agrupa cada segmento dos dados para um objeto dentro de uma única barra.

Show Legend – Esta opção adiciona uma legenda ao gráfico.

Show Legend - This option adds a legend to the graph.

Font Size - This defines the font size of graph text.

Bar Size - Bar Chart Only. This defines the bar height for Bar Charts

Only Collect Data for a Defined Time Interval – Esta opção armazena os dados apenas um período de tempo pré-determinado.

From Time – Este campo define quando as estatísticas serão gravadas.

To Time – Este campo define quando as estatísticas não serão mais gravadas.

Start Time – Este campo define o primeiro momento mostrado no gráfico.

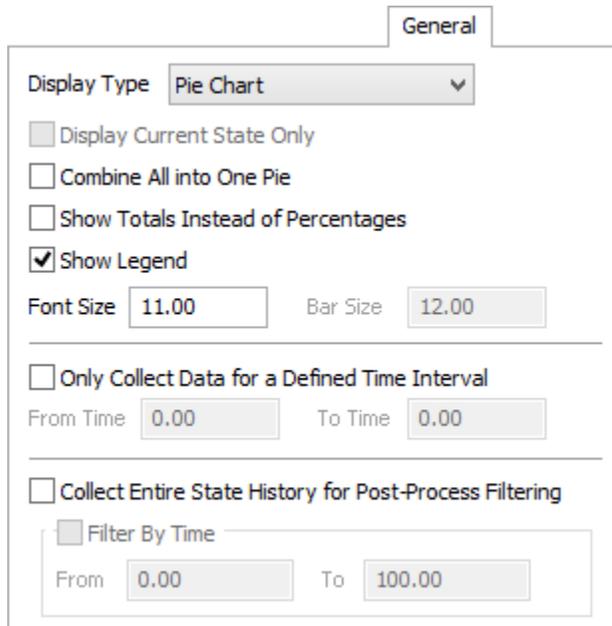
Interval – Este valor define o intervalo entre uma atualização e outra do gráfico.

Time Scale - This defines the time scale that numbers will be displayed in along the x-axis.

Y Axis Title - Text that will display along the Y Axis.

X Axis Title - Text that will display along the X Axis

State Statistic Graphs



General

Display Type: Pie Chart

Display Current State Only

Combine All into One Pie

Show Totals Instead of Percentages

Show Legend

Font Size: 11.00 Bar Size: 12.00

Only Collect Data for a Defined Time Interval

From Time: 0.00 To Time: 0.00

Collect Entire State History for Post-Process Filtering

Filter By Time

From: 0.00 To: 100.00

Display Type - This option changes the display style of graph.

Display Current State Only - This option is only available for Table of Values charts. Displays the current state name of objects.

Combine All into One Pie - This option combines all pie charts into a single pie chart. It averages the values from all objects.

Show Totals Instead of Percentages - This option changes the displayed values from percentages to overall totals.

Show Legend - This option adds a legend to the graph.

Font Size - This defines the font size of graph text.

Bar Size - Bar Chart Only. This defines the bar height for Bar Charts.

Only Collect Data for a Defined Time Interval - This option creates a time period during which statistics for this graph will be recorded.

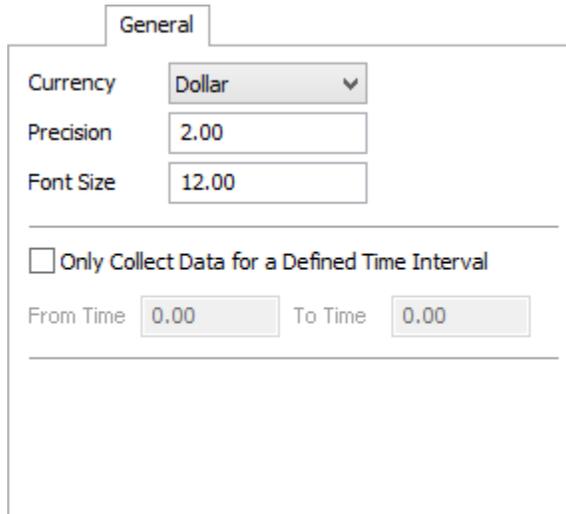
From Time - This defines when statistics will start being recorded.

To Time - This defines when statistics will no longer be recorded.

Filtragem Após o Processo - Post-Process Filtering

Caso a opção **Collect Entire State History for Post-Process Filtering**, o FlexSim irá gravar os dados de cada objeto selecionado a cada vez que ele mudar de estado. Depois de rodar a simulação, você pode checar a opção **Filter By Time** e entrar com um intervalo de tempo nos campos **From** e **To**. Se você clicar depois em **Apply** ou em **Update**, o gráfico mostrará o estado apenas naquele intervalo de tempo. A opção **Filter By Time** será desabilitada quando o modelo for **Resetado**.

Financial Analysis



The screenshot shows the 'General' tab of a dialog box for financial analysis. It contains the following fields and options:

- Currency:** A dropdown menu set to 'Dollar'.
- Precision:** A text input field containing '2.00'.
- Font Size:** A text input field containing '12.00'.
- Only Collect Data for a Defined Time Interval:** An unchecked checkbox.
- From Time:** A text input field containing '0.00'.
- To Time:** A text input field containing '0.00'.

Currency - This options changes the prefix of each value to be displayed. The *Other* option allows you to define a custom string value, or leave the field blank to display no prefix.

Precision - This defines how many numbers will be displayed right of the decimal point.

Font Size - This defines the font size of graph text.

Only Collect Data for a Defined Time Interval - This option creates a time period during which statistics for this graph will be recorded.

From Time - This defines when statistics will start being recorded.

To Time - This defines when statistics will no longer be recorded.

Gantt Chart

The screenshot shows the 'General' tab of a configuration dialog for a Gantt Chart. It includes the following controls:

- Display Type:** A dropdown menu set to 'States'.
- Show Legend:** A checked checkbox.
- Font Size:** A text input field containing '11.00'.
- Bar Size:** A text input field containing '12.00'.
- Only Collect Data for a Defined Time Interval:** An unchecked checkbox.
- Interval Type:** A dropdown menu set to 'Absolute Time'.
- From Time:** A text input field containing '0.00'.
- To Time:** A text input field containing '0.00'.
- Time Units:** A dropdown menu set to 'Seconds'.
- X Axis Title:** An empty text input field.

Display Type - This option changes the display type of graph. This may either be States or Item Trace.

Show Legend - This option adds a legend to the graph.

Font Size - This defines the font size of graph text.

Bar Size - This defines the bar height.

Only Collect Data for a Defined Time Interval - This option creates a time period during which statistics for this graph will be recorded.

Interval Type - Some charts may have an additional option of specifying the Interval Type, like the Gantt Chart. The Interval Type can be set to:

- **Absolute Time:** Starts collecting data at the From Time and ends collecting data at the To Time.
- **Time Window:** Collects data for a specified length of time. The Length field defines the length of time from the first recorded statistic to the last recorded statistic and dynamically updates as the model runs. As the data is gathered in a model run, data from the start will be removed in order to keep all of the data in the chart within the specified length of time.

From Time - This defines when statistics will start being recorded.

To Time - This defines when statistics will no longer be recorded.

Tracked Variable Graphs

The screenshot shows the 'General' tab of a configuration dialog for Tracked Variable Graphs. It includes the following controls:

- Display Type:** A dropdown menu set to 'Histogram'.
- Show Legend:** A checked checkbox.
- Normalize Values:** An unchecked checkbox.
- Number of Bars:** A text input field containing '10.00' with up and down arrow buttons.

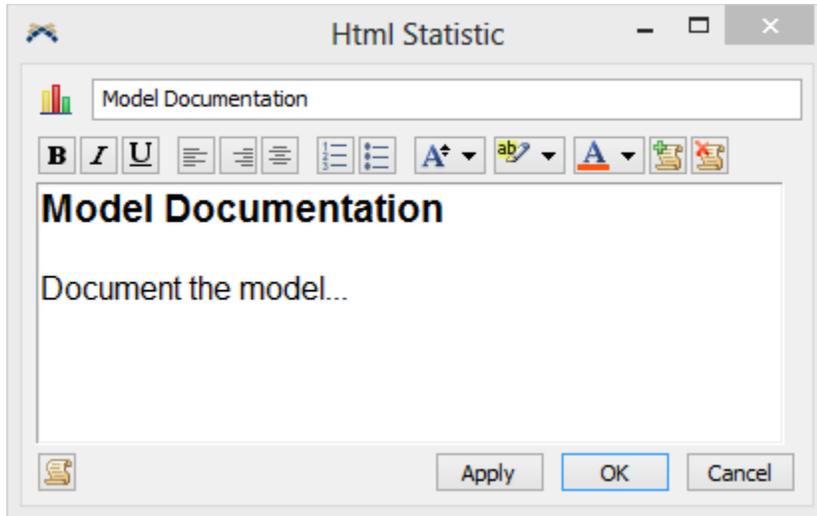
Display Type - This option changes the display style of graph.

Show Legend - This option adds a legend to the graph.

Normalize Values - This option shows the values of the histogram as a percent, rather than an actual value.

Number of Bars - This option adjust the number of bars that the histogram has.

HTML Statistic (Model Documentation)



Name Field – Este campo define um nome para a documentação gráfica mostrada.

 - Este botão faz com que o texto fique em negrito. Caso toda a seleção já estiver em negrito, então ele remove este efeito.

 - Este botão faz com que o texto fique em Itálico. Caso o texto selecionado já esteja em Itálico, então este botão desfaz o efeito.

 - Este botão sublinha o texto selecionado. Caso o texto selecionado já esteja sublinhado, então este botão desfaz o efeito.

 - Este botão alinha o texto selecionado à borda da esquerda. Caso várias linhas estejam selecionadas, todas ficarão justificadas a esquerda.

 - Este botão alinha o texto selecionado à borda da direita. Caso várias linhas estejam selecionadas, todas ficarão justificadas a direita.

 - Este botão alinha o texto selecionado ao centro. Caso várias linhas estejam selecionadas, todas ficarão justificadas ao centro.

 - Este botão coloca a linha selecionada em uma lista ordenada. Caso múltiplas linhas sejam selecionadas, todas serão colocadas na lista ordenada.

 - Este botão coloca a linha selecionada em uma lista desordenada. Caso múltiplas linhas sejam selecionadas, todas serão colocadas na lista.

 - Este botão muda a fonte do da seleção atual para o tamanho especificado.

 - Este botão destaca o texto selecionado na cor especificada.

 - Este botão altera a cor da fonte selecionada para a cor desejada.

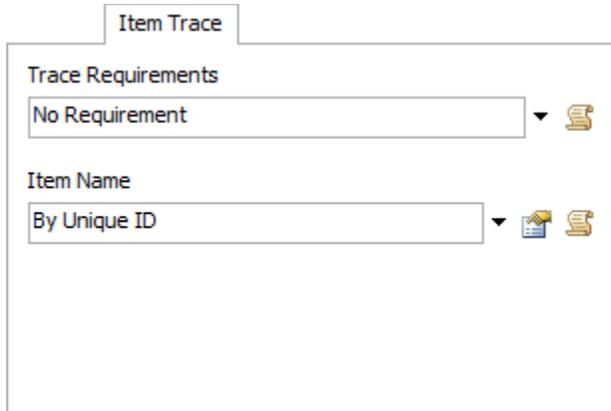
 - Este botão insere uma seção de códigos flexscript, do qual pode ser usado para alterar dinamicamente a documentação do modelo conforme a simulação roda.

 - Este botão remove a seção de códigos flexscript.



- Este botão alterna entre o modo de visão do editor visual e do editor em html.

Item Trace Page



The screenshot shows a window titled "Item Trace" with two picklist controls. The first control is labeled "Trace Requirements" and has a dropdown menu with "No Requirement" selected. The second control is labeled "Item Name" and has a dropdown menu with "By Unique ID" selected. Both picklists have a small icon to their right, likely representing a help or search function.

Trace Requirements - This picklist allows you to define which flowitems will be traced.

- current: The object as defined in the Objects page.
- item: The involved flowitem.

Item Name - This picklist allows you to define a custom display name for traced items.

- current: The object as defined in the Objects page.
- item: The involved flowitem.
- itemnode: The node associated with the item and its graph data. Setting the name of this node will set the display name in the graph.

Página de Objetos

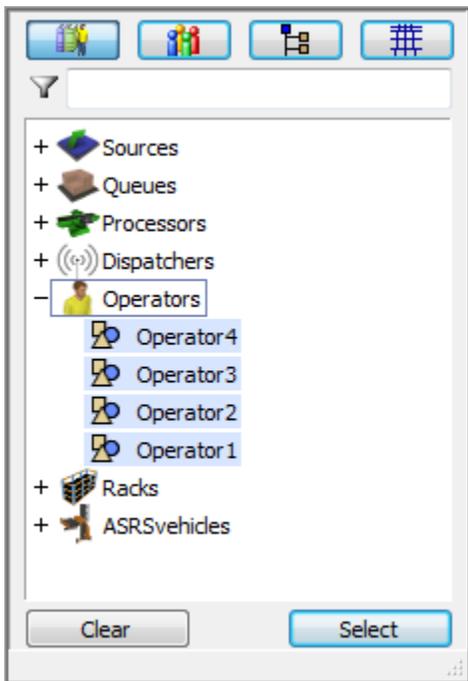


-  - Adiciona os objetos a lista atual de objetos
-  - Remove os objetos da lista atual de objetos
-  - Sample (copia as propriedades) de um objeto no modelo.
-  - Move o objeto selecionado para uma posição acima da atual na lista
-  - Move o objeto selecionado para uma posição Guiaixo da atual na lista
-  - Combina diversos objetos em um único
-  - Divide objetos combinados anteriormente.

Adição de Objetos

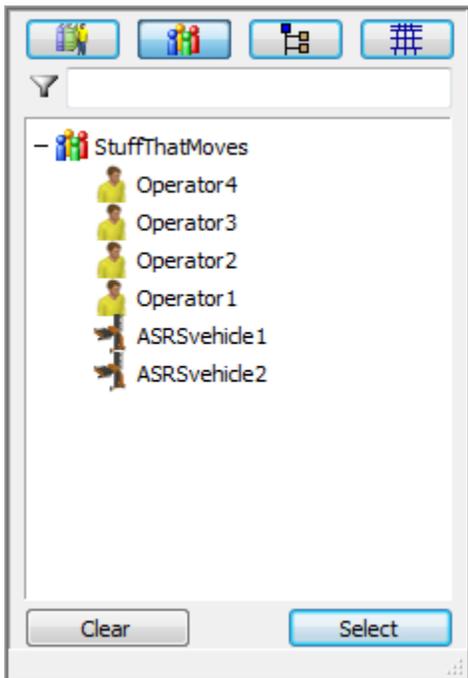
Existem 4 maneiras de adicionar objetos ao gráfico. O modo selecionado fica destacado no topo da janela.

Escolher através da Classe



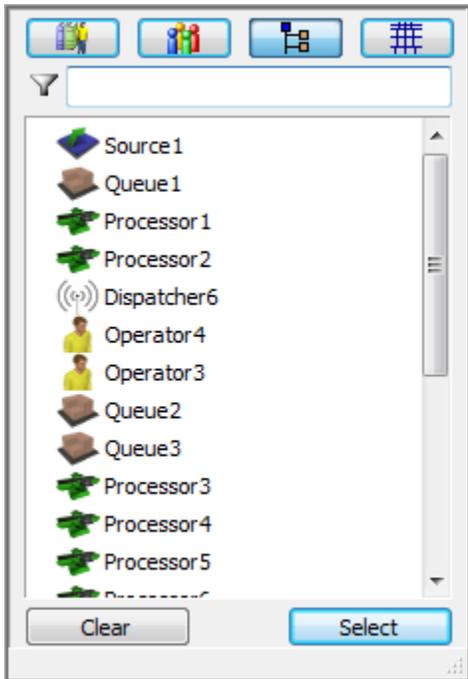
Este método de adição dos objetos faz a classificação através das classes de cada objeto. Para seleccionar toda uma classe, clique no ícone desta classe. Clique em um objeto para seleccionar/tirar a seleção.

Escolher através dos Grupos



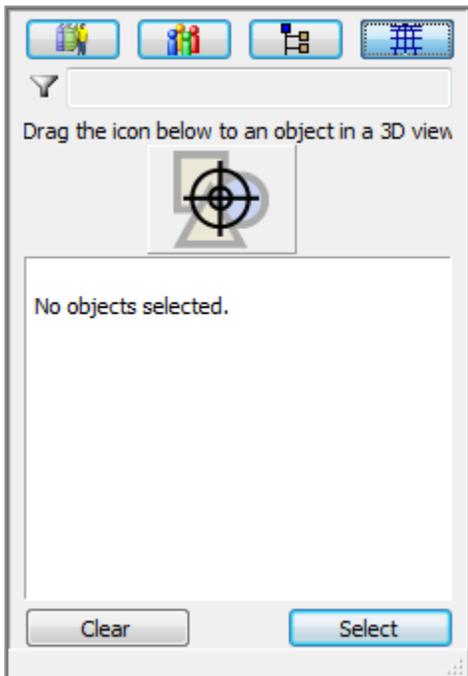
Este método de adição dos objetos faz a classificação através dos grupos criados pelo usuário. Para seleccionar um grupo todo, clique no ícone deste grupo. Clique em um objeto para seleccionar/tirar a seleção.

Escolher diretamente no Objeto



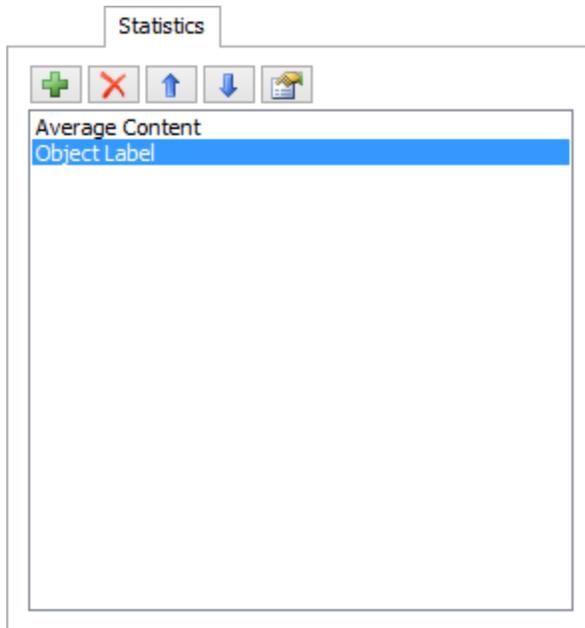
Este método de adição lista todos os objetos existentes no modelo. Clique em um objeto para selecionar/tirar a seleção.

Selecionar Arrastando



Este método de adição utiliza o modelo 3D para adicionar os objetos. Simplesmente arraste o alvo na tela para o objeto no modelo que você gostaria de adicionar.

Página Estatística



- + - Adiciona um dado estatístico à lista atual de estatísticas
- X - Remove um dado estatístico da lista atual de estatísticas
- ↑ - Move o dado selecionado para uma posição acima da atual na lista
- ↓ - Move o dado selecionado para uma posição Guiaixo da atual na lista.

Estatísticas Disponíveis

Average Content	
Minimum Content	
Maximum Content	
Current Content	
<hr/>	
Average Staytime	
Minimum Staytime	
Maximum Staytime	
<hr/>	
Output per...	▶
Input per...	▶
Total Output	
Total Input	
<hr/>	
Distance Traveled per Time	▶
Total Distance Traveled	▶
<hr/>	
Object Label Value	
Custom	

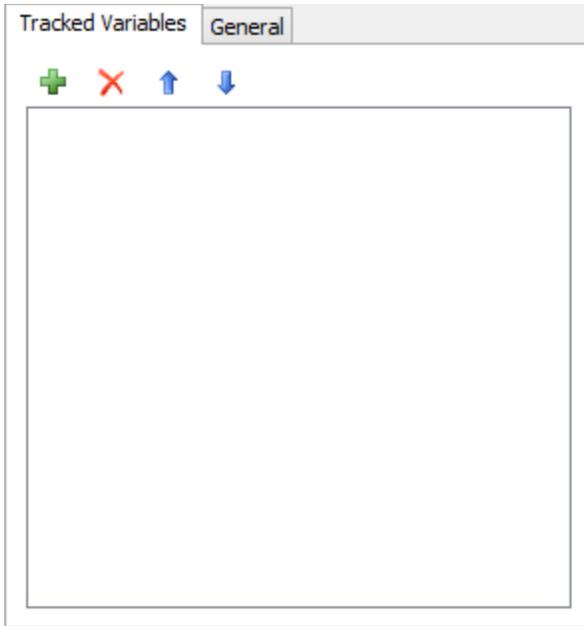
Output per... – Esta estatística medirá o número de itens que saem do objeto por cada unidade de tempo, onde a unidade de tempo poderá ser selecionada em uma lista lateral.

Input per... - Esta estatística medirá o número de itens que entram no objeto por cada unidade de tempo, onde a unidade de tempo poderá ser selecionada em uma lista lateral.

Distance Traveled per Time – Esta estatística medirá a velocidade de um objeto. Tanto a unidade de distância e a unidade de tempo podem ser selecionadas na lista lateral.

Total Distance Traveled – Esta estatística medirá a distância total que um objeto se deslocou ao longo da simulação. A unidade de distância pode ser selecionada na lista lateral.

Tracked Variables



- + - Adiciona uma Tracked Variable a lista atual de tracked variables
- X - Remove uma Tracked Variable da lista atual de tracked variables
- ↑ - Move a Tracked Variable selecionada para uma posição acima da atual na lista
- ↓ - Move a Tracked Variable selecionada para uma posição Guiaixo da atual na lista

Página de Análise de Utilização

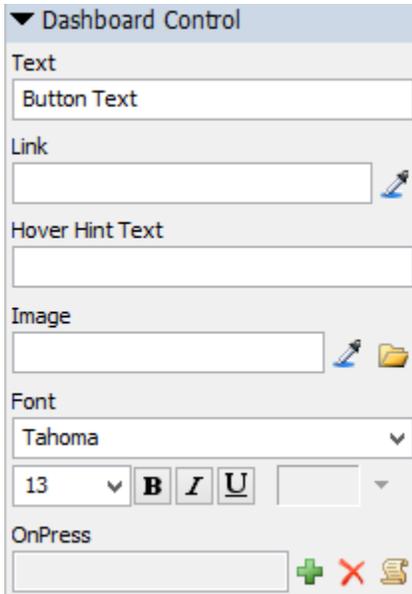
The screenshot shows a dialog box titled "Utilization Analysis" with three tabs: "Objects", "Utilization Analysis", and "General". The "General" tab is active. It contains three radio button options: "Show All Checked States (Do Not Calculate Percentage)" (selected), "Show Only Green Checked States", and "Show Yellow Checked States as Translucent". Below these is a list of states with checkboxes: "idle" (checked with a red X), "processing" (checked with a green checkmark), "blocked" (checked with a yellow checkmark), "waiting for operator" (checked with a yellow checkmark), "waiting for transporter" (checked with a yellow checkmark), and "setup" (checked with a green checkmark). At the bottom, there is a legend: a green checkmark for "Included in the percentage calculation (i.e. utilization)", a yellow checkmark for "Not included in the percentage calculation", and a red X for "Excluded from the total time calculation".

Show All Checked States – Esta opção mostra todos os estados com cores sólidas. Deste modo, a utilização do objeto não é calculada.

Show Only Green Checked States – Esta opção mostra a percentagem de tempo que o objeto permaneceu nos estados marcados em verde. Esta opção não mostra nenhuma outra informação dos estados.

Show Yellow Checked States as Translucent – Esta opção mostra os estados checados em verde em cores sólidas e os estados checados em amarelo são mostrados também, mas em cores translucidas.

Model Input Properties



Text - Especifica o título ou o texto do controle.

- Disponível para: Texto Estatístico, Checkbox, Button

Link - Especifica o caminho para um node no modelo.

- Disponível para: Dynamic Text, Edit, Checkbox, Combobox, Button

Hover Hint Text - Este texto será mostrado como uma dica quando o usuário passar seu mouse sobre o controle.

- Disponível para: Dynamic Text, Edit, Checkbox, Combobox, Button

Image - Mostra uma imagem ao invés de texto para o controle.

- Disponível para: Static Text, Button

Font - Especifica o nome da fonte, propriedades e cor dos textos de controle (cor somente disponível para Static Text, Dynamic Text e controles de Edição).

- Disponível para: Todos

OnPress - Trigger que disparam quando o controle é pressionado.

- Disponível para: Checkbox, Button

Combobox Only

OnSelect

+

×

📄

▼ Combobox Options

+

×

a	1.00
b	2.00
c	3.00

OnSelect - Trigger que dispara quando um item no combobox é selecionado.

Combobox Options - Uma lista de itens que mostram no drop down do combobox. Valores devem ser numéricos.

Options - A list of items to display in the combobox/listbox drop down. Values must be numeric.

Linking to string data: If a combobox or listbox is linked to a node with string data like a label, selecting an option will set the value of the linked label or node to be the name of the option, rather than its numeric value.

Tracker

▼ Tracker

Minimum Value

0.00

Maximum Value

1.00

Exponential Value

1.00

Style

Bottom ▼

Vertical

Minimum Value - The minimum value of the tracker.

Maximum Value - The maximum value of the tracker.

Exponential Value - Changes the distribution of values between the minimum and maximum to be exponential. Setting this value to 1 will cause the tracker to be linear.

Style - Changes the style of the thumb button.

Vertical - If checked, the tracker will display vertically.

Reports and Statistics

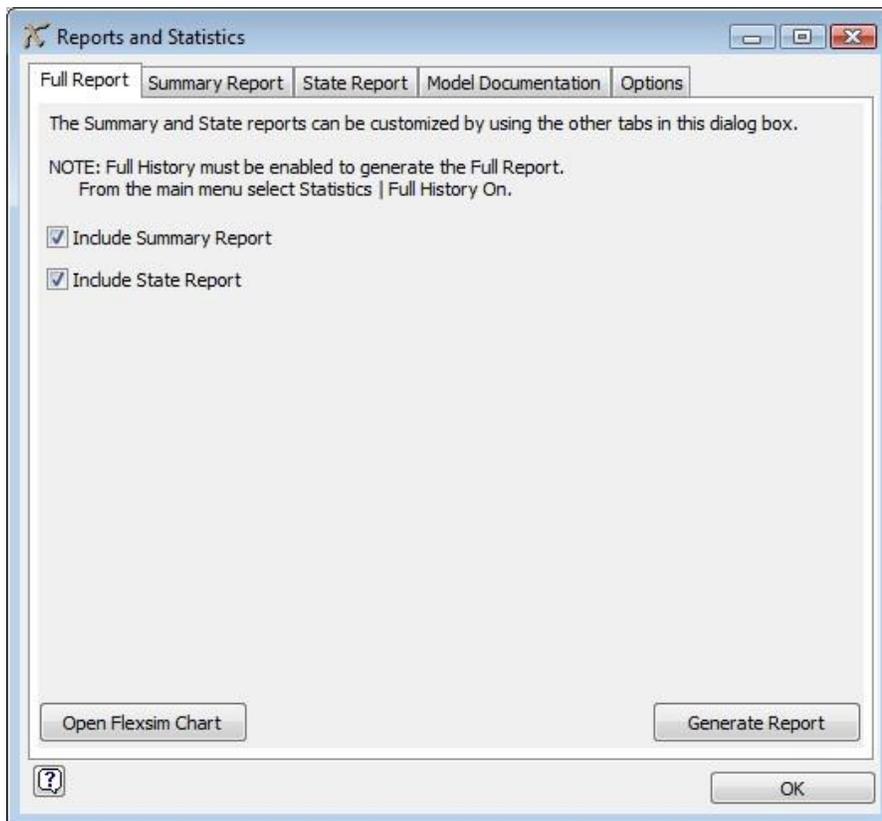
A janela do Reports and Statistics permite ao modelador criar vários relatórios baseados nas estatísticas coletadas durante a simulação do modelo. Estes dados podem conter informações sobre o fluxo de itens, tempos de permanência, histórico de estados e outros dados que o usuário pode selecionar e customizar. Esta janela também permite ao modelador criar um documento que contenha os mais importantes detalhes sobre os objetos no modelo.

Referencia

- **Full Report Tab**
- **Summary Report Tab**
- **State Report Tab**
- **Model Documentation Tab**
- **Options Tab**

Guia Full Report

A guia Full Report lhe permite gerar um relatório completo do modelo simulado, e será salvo em uma base de dados e aberto com o aplicativo FlexSim Chart. Esta base de dados poderá apenas ser gerada se a opção Full History na aba statistics estiver habilitada durante a simulação. Caso esta opção não esteja habilitada, então uma mensagem aparecerá dizendo que os dados não foram coletados. Para coletar estes dados e abri-los com o aplicativo gerador de gráficos, pressione o botão "Generate Report". As classes que aparecem o relatório completo pode ser customizada através das opções existentes na aba Options.



Include Summary Report – Caso esta opção esteja checada, o Summary Report (explicado abaixo) será incluído na base de dados que será salva. O aplicativo gerador de gráficos pode mostrar apenas as primeiras 32 colunas do relatório. Caso você precise ver mais do que isso, gere um relatório através da aba Summary Report. As classes que serão analisadas podem ser customizadas através da aba Options. Os atributos e os outros valores mostrados podem ser escolhidos através da aba Summary Report.

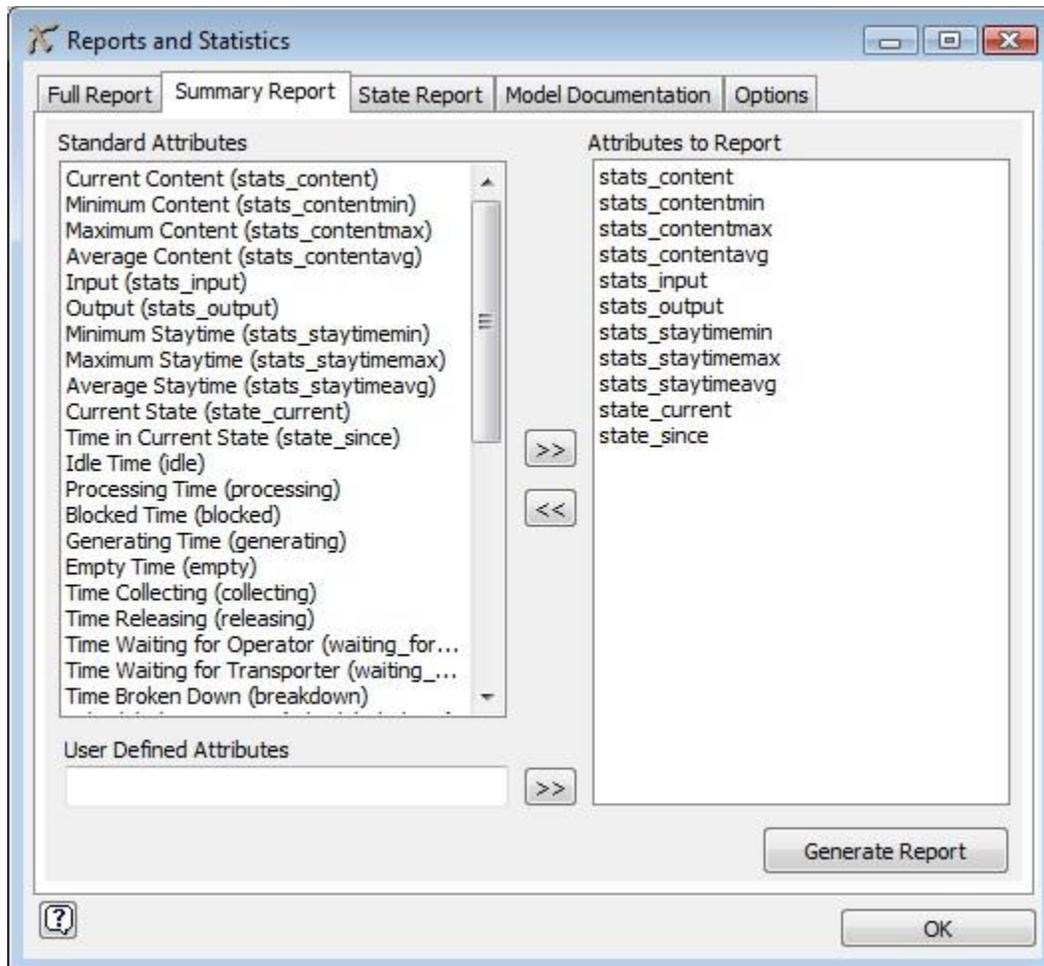
Include State Report – Caso esta opção esteja checada, o State Report (explicado abaixo) será incluído na base de dados que será salva. O aplicativo gerador de gráficos pode mostrar apenas as primeiras 32 colunas do relatório. Caso você precise ver mais do que isso, gere um relatório através da aba State Report. Geralmente não será necessário analisar menos estados no relatório final, e neste caso, você pode remover alguns estados e adicionar outros de seu interesse através da aba State Report.

Open FlexSim Chart – Pressione este botão para abrir um arquivo com uma base de dados do FlexSim Chart já existente. Caso ainda não exista este arquivo, então clique no botão “Generate Report”.

Generate Report – Pressione este botão para gerar um relatório no FlexSim Chart e criar uma base de dados.

Summary Report Tab

A guia de relatório padrão lhe permite criar um relatório do seu modelo. O FlexSim reporta uma lista de atributos que você define para o modelo. Uma vez criada esta lista com os atributos desejáveis, clique no botão “Generate Report”, que o relatório será então criado e exportado para o Microsoft Excel.



Standard Attributes – Esta opção lhe permite selecionar as variáveis padrões presentes no modelo, como conteúdo (contente), tempo de permanência (staytime), variáveis de estados, etc. Pressione o botão  no topo da lista para selecionar alguma tributo para o relatório desejado.

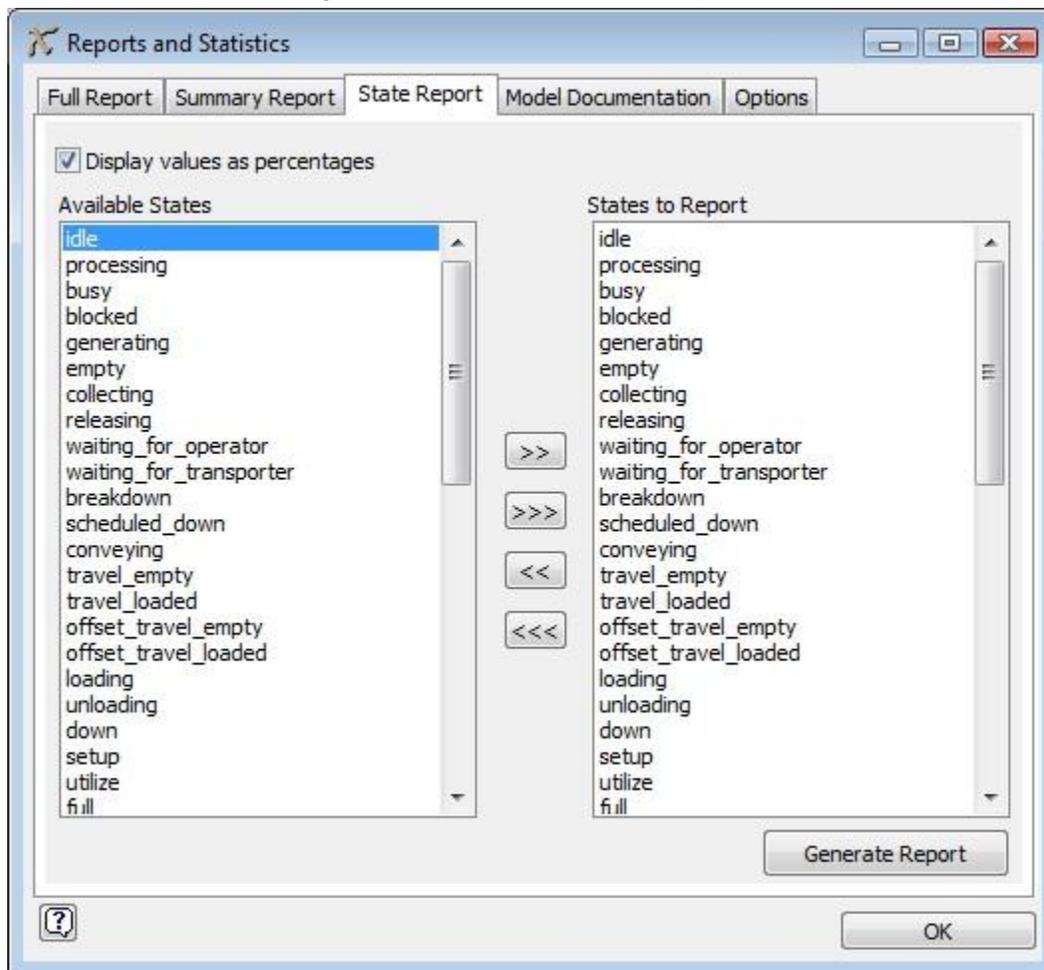
User Defined Attributes – Nesta opção você pode digitar o nome de alguma label ou alguma variável que você quer inserir no relatório e então clicar no botão  para inserir seu próprio atributo a lista de variáveis. Por exemplo, se um ou mais Queues possuírem uma label chamada “ultimoitemvermelho”, e você quer um relatório de todas estas labels, então digite “ultimoitemvermelho” neste campo e pressione o botão .

Attributes to Report – Esta é a lista de atributos que será reportada. Para remover algum item desta lista, basta selecionar este item e pressionar o botão .

Generate Report – Pressione este botão para gerar o relatório.

State Report Tab

A guia State Report lhe permite criar um relatório relacionado ao tempo de permanência de um objeto em um estado específico. Estes dados podem ser relatados tanto de forma numérica exata, como também em porcentagem. Uma vez criada a lista dos estados desejados para inclusão no relatório, clique no botão “Generate Report” para gerar o relatório e exporta-lo para o Microsoft Excel.



Display values as percentages – Se esta opção estiver checada o relatório será apresentado em forma de porcentagem, dividindo o tempo total da simulação entre todos os estados assumidos pelo objeto ao longo deste tempo. Se este não estiver checado, então no relatório aparecerá o tempo exato que um objeto permaneceu em cada estado.

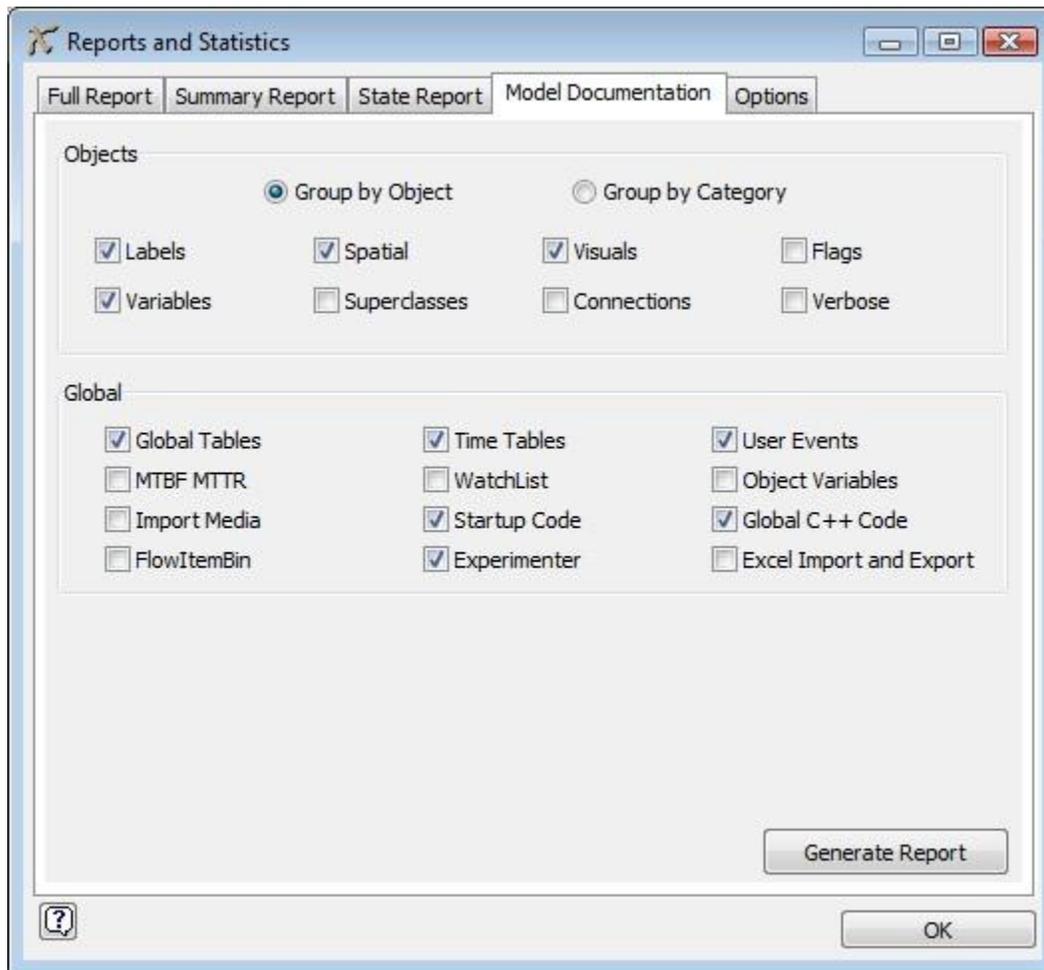
Available States – Estes são os estados disponíveis no FlexSim que podem ser incluídos no relatório. Para inserir um estado na coluna States to Report, selecione o estado desejado e clique no botão. Para colocar todos os estados na coluna States to Report, basta pressionar o botão >>>.

States to Report – O tempo que os objetos permaneceram nestes estados serão mostrados no relatório. O tempo para todos estes estados serão reportados para todos os objetos, mesmo que o objeto jamais tenha assumido aquele estado específico. Para remover um estado desta lista, selecione o estado desejado e clique no botão <<. Para remover os estados desta lista, pressione <<<.

Generate Report – Pressione este botão para gerar o relatório.

Model Documentation Tab

A guia Model Documentation lhe permite criar um documento no formato de texto (.txt) que contém diversas informações sobre o modelo. Cheque as caixas apropriadas que você deseja documentar-las e então clique no botão “Generate Report” para criar o arquivo.



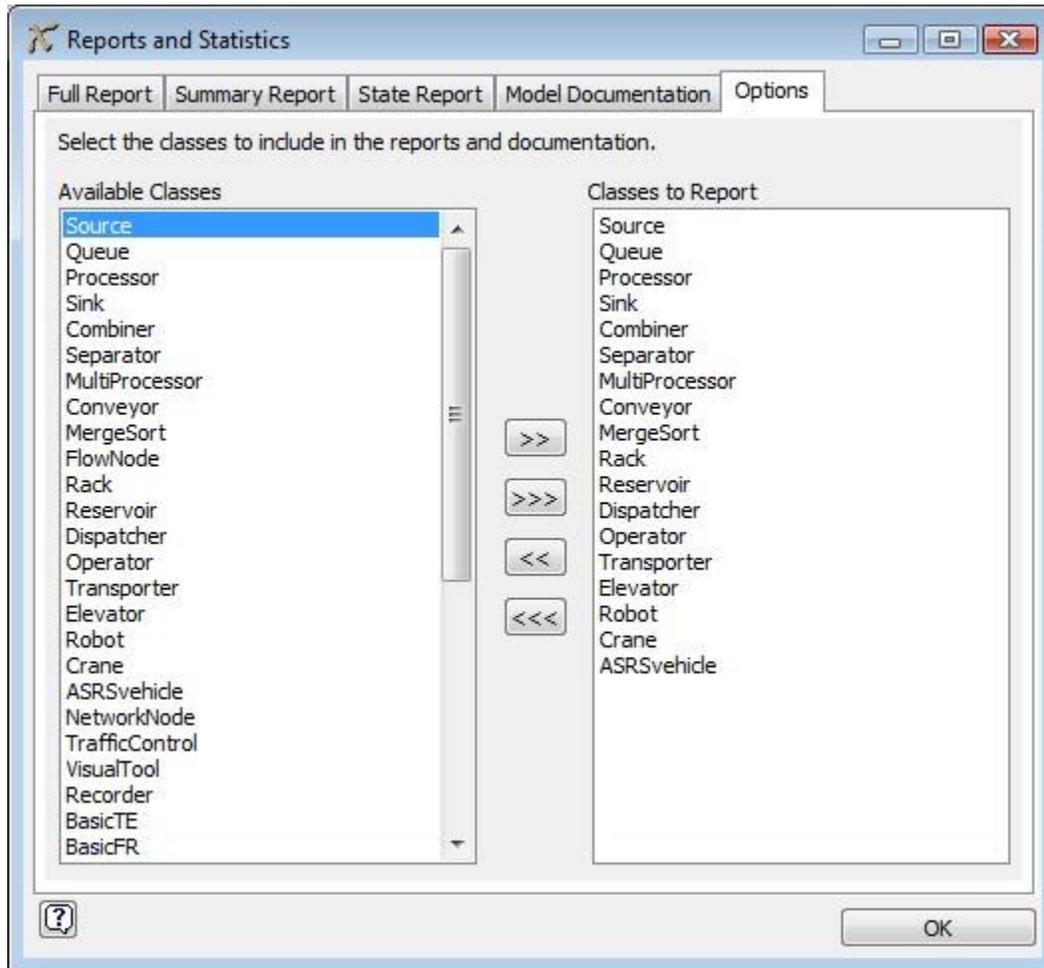
Objects – Estas opções permitem ao modelador selecionar quais atributos dos objetos do modelo deverão ser incluídos no relatório. Se a opção “Group by Object” estiver selecionadas, todos os atributos selecionados para cada objeto estarão juntos no relatório final. Se a opção “Group by Category” estiver selecionada, então todos os valores de cada atributo estarão juntos no relatório final. Caso a opção “Verbose” estiver selecionada, então cada campo de código (triggers, process time, etc) será documentada com todo o texto presente naquele campo. Caso “Verbose” não esteja selecionada, então apenas o texto que aparece na janela da template será mostrado na documentação.

Global – Esta opção permite ao modelador selecionar quais recursos e objetos globais serão incluídos no relatório.

Generate Report – Pressione este botão para gerar o relatório.

Options Tab

A guia options lhe permite escolher quais classes dos objetos serão mostradas nos relatórios.



Available Classes – Estas são todas as classes disponíveis no FlexSim das quais suas instâncias poderão ser incluídas nos relatórios. Estas classes são as que aparecem no Library Icon Grid. Você pode adicionar uma classe ao campo “Classes to Report” selecionando a classe desejada e clicando no botão . Para adicionar todas as classes às classes a serem reportadas, basta clicar no botão .

Classes to Report – Todos os objetos do modelo que são instâncias destas classes serão incluídas em qualquer relatório gerado. Quaisquer instâncias das classes que não foram selecionadas não serão incluídas nos relatórios. Para remover alguma classe desta lista, selecione a classe e clique no botão . Para remover todas as classes, clique no botão .

Experimenter / Optimizer

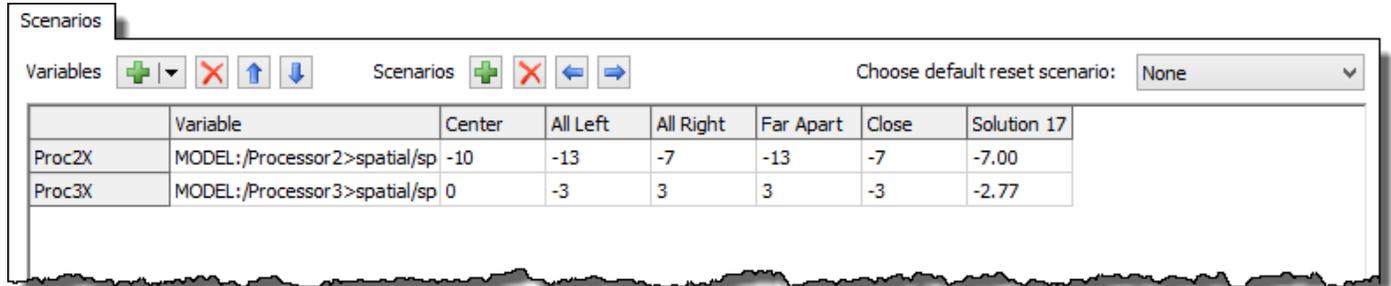
1. Experimenter
2. Otimização no FlexSim
3. Exemplo

Experimenter

O Experimenter é usado para definir, rodar e analisar experimentos sobre os cenários definidos do modelo. Veja o exemplo do Experimenter como um demonstrativo de como usar o Experimenter.

Scenarios Tab

A guia Scenarios é onde você define as variáveis e cenários associados com o experimento.



	Variable	Center	All Left	All Right	Far Apart	Close	Solution 17
Proc2X	MODEL:/Processor2>spatial/sp	-10	-13	-7	-13	-7	-7.00
Proc3X	MODEL:/Processor3>spatial/sp	0	-3	3	3	-3	-2.77

Variables (Variáveis)

As variables são coisas no modelo que você quer alterar como parte de um dado do experimento. Elas podem ser valores simples em labels ou global tables ou elas podem ser o número de operadores para uma determinada equipe. Você cria e edita variáveis na guia Scenarios. Cada variável é uma linha na tabela Scenarios.

Adding / Removing / Defining Variables – Use o painel de botões  para adicionar, remover e posicionar as variáveis na tabela. Clique sobre a seta preta drop-down para escolher o tipo de variável que você gostaria de adicionar. Uma vez escolhida a variável, você pode alterá-la ao clicar na célula “Variable”.

O botão  irá aparecer. Você pode usar estas para alterar as variáveis uma vez que foi adicionado. Defina o nome das variáveis na parte esquerda da tabela. Você pode alterar o nome simplesmente clicando na célula.

Scenarios (Cenários)

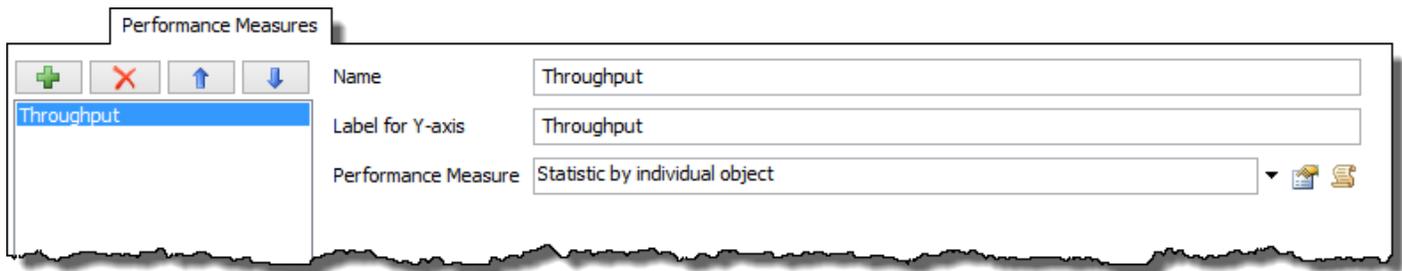
Um cenário (scenario) é uma configuração específica do conjunto de variáveis que você definiu. Você define um cenário ao configurar os valores na coluna da tabela de cenários associada com aquele cenário.

Adding / Removing / Defining Scenarios – Use o painel de botões  para adicionar, remover e reorganize os cenários. Uma vez adicionado, entre com o valor de cada variável na coluna da tabela daquele cenário.

Choose Default Reset Scenario - Escolha None se você não quer o modelo defina seu cenário quando resetado. Você pode escolher um cenário que você quer que o seu modelo vá quando você reseta seu modelo fora do experimento. Uma vez escolhido, sempre que você resetar, o modelo retornará para aquele cenário. Escolha None se você não quiser que o modelo

Performance Measures Tab

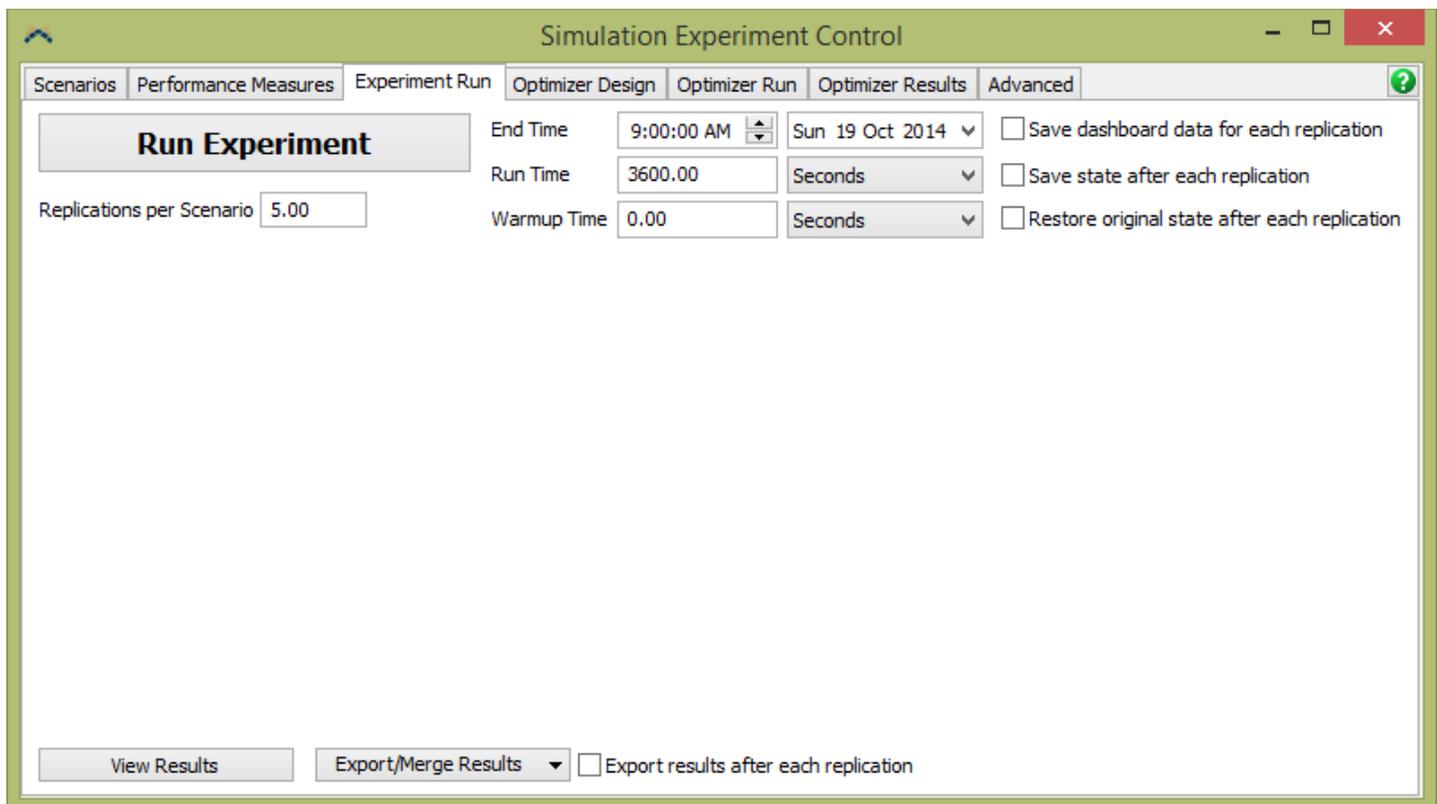
A guia Performance Measures é onde você define as medidas de performance para o experimento, por exemplo, algo como throughput (produtividade), average wait time (tempo médio de espera), etc. para comparação de cenários.



Add / Remove / Define – Use o botão para     adicionar, remover e organizar medidas de performance na listagem à esquerda. Defina cada nome das medidas de performance (**Name**, e **Label for Y-axis**) (mostrado nos resultados). Então use ▼ para escolher o tipo de medida de performance informado. Se você adicionar informações estatística para um dashboard, o botão  vai mostrar uma listagem drop-down onde você pode adicionar medidas de performance associado com dashboards estatísticos específicos.

Experiment Run Tab

A guia Experiment Run é onde você define os parâmetros para o experimenter e roda ele.



Experiment Parameters (Parâmetros do Experimenter)

Run Time - O tempo total de simulação que cada experiment irá rodar.

Warmup Time - O tempo de simulação que cada replicação irá rodar antes de resetar suas informações estatísticas (tempo de estabilização do experimento). As estatísticas irão além do que ser somente coletada para um período de tempo (Run Time – Warmup Time).

Replications Per Scenario - O número de replicações que irão rodar para cada cenário.

Save Dashboard Data for Each Replication - Se esta opção estiver marcada no final de cada replicação, o FlexSim irá salvar os dados em cada dashboards, e então eles poderão ser visualizados mais tarde como parte dos resultados.

Save State After Each Replication - Se marcado, cada estado de todas as replicações da simulação serão salvas em um arquivo no final da replicação. Isto permitirá você abrir a replicação no estado onde foi finalizado.

Restore Original State After Each Replication - Se marcado, o Flexsim irá recarregar completamente o modelo entre a execução de cada modelo. Você pode querer marcar esta caixa se o seu modelo não resetar adequadamente para exatamente o mesmo estado toda vez que você resetar, e você não vai querer que este estado “fora do contexto” afetando os subseqüentes resultados da replicação.

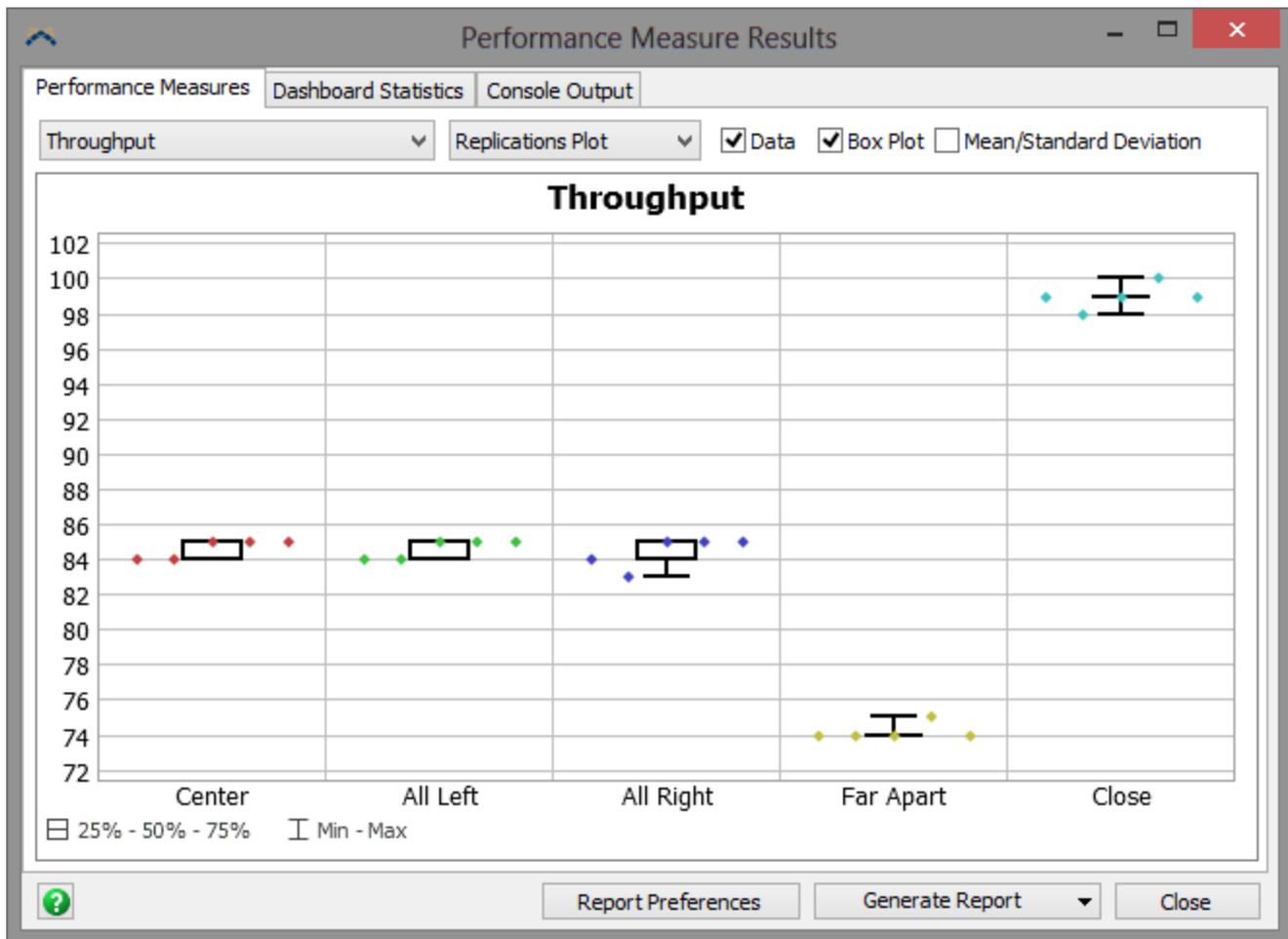
Analyzing Results (Analisando os Resultados)

Uma vez que o experiment está finalizado (todas as barras do Experiment Status estão em verde), você pode analisar os resultados de seu experimento ao clicar no botão **View Results**.

Export / Merge Results – Este botão permite a você salvar seu resultado para um arquivo, assim como carregar/juntar os resultados de um arquivo já salvo, dentro de seus resultados atuais. **Export** irá salvar os resultados em um arquivo de extensão.t. **Load** irá carregar os resultados de um arquivo salvo dentro de seu modelo, substituindo qualquer resultado atual. **Merge** irá carregar os resultados de um arquivo, e então juntar estes resultados com os resultados de seu experimento atual. Você pode também ter resultados salvos em um arquivo após cada replicação ao clicar em **Export results after each replication**.

Performance Measure Results (Resultados de Medida de Performance)

A janela **Performance Measure Results** mostra os dados para medida de performance. Existem algumas diversas opções de como mostrar os dados, incluindo gráficos de replicações (Replications Plot), histograma de frequência (Frequency Histogram), gráficos de correlação (Correlation Plot) (para examinar a correlação entre medidas de múltipla performance), um sumário de dados (Data Summary) e visualização bruta dos dados.



Generating Reports (Gerando Relatórios)

Existem duas opções para geração de relatórios.

Report Format - Isto cria um arquivo html com os resultados completos para todas as medidas de performance. Você pode definir as preferências de relatório (**Report Preferences**), permitindo que você adicione/remova os dados que você quer incluir no relatório.

Web Viewer Format - Quando você abrir o arquivo .html você poderá escolher quais gráficos você gostaria de visualizar.

Dashboard Statistics Tab (Guia Dashboard Statistics)

Se você marcou a caixa **Save Dashboard Data for Each Replication** na guia Experiment Run antes de rodar o experiment, então você pode ir para a guia Dashboard Statistics para visualizar os resultados das replicações individuais para várias estatísticas do dashboard. Escolha a estatística, replicação e cenário, e o gráfico associado a estatística será mostrado.

Console Output Tab (Guia Console Output)

O FlexSim também salva os textos de saída do Output Console e do System Console para cada replicação. Isto pode ser usado para debugging/análises de cada replicação.

Otimização no FlexSim

O que é otimização?

Otimização inicia com um modelo. Para otimização em geral, este modelo pode ser qualquer sistema que aceita um conjunto de dados de entrada e então produz os melhores dados de saída para um dado modelo. Dentro do Flexsim, é possível otimizar um modelo de simulação. Os dados de entrada podem ser uma análise das seguintes opções de configurações: Quantas pessoas devem ser contratadas? Onde devem ser colocados os racks de armazenagem? Qual tipo de máquina irá trabalhar melhor? Os dados de saída podem ter resultados como: Qual foi a utilização dos funcionários? Quanto tempo os itens permanecerão no modelo? Qual foi o custo total por item? O FlexSim fornece uma ferramenta fácil de usar que permite a você especificar os dados de entrada e saída para o seu modelo. Uma vez realizado, o FlexSim podem inteligentemente buscar através de possíveis configurações os melhores resultados de saída possíveis; em outras palavras, pode achar uma configuração ótima de seu modelo.

Nota: Este documento contém muita informação. Se você é novo em otimização com o FlexSim, um boa estratégia de aprendizagem seria ler este documento, e então fazer o exercício de otimização, e então ler este documento novamente.

Configurações, Cenários e Soluções

No FlexSim, a ferramenta Simulation Experiment Control permite ao usuário definir os cenários para o modelo. Um cenário consiste de um valor para cada variável do experimento, o qual resulta em uma configuração do modelo. O optimizer também gera configurações para o modelo baseado no mesmo conjunto de variáveis usado pelo experimenter. Para distinguir este tipo de configuração de um cenário, ele é chamado solução. Os usuários geram cenários, e o optimizer gera soluções. Ambos configuram o modelo baseado nas variáveis definidas no Experiment Simulation Control.

General Algorithm

Quando o optimizer está rodando, ele está buscando pela melhor das melhores configurações para o modelo. Aqui uma versão básica deste algoritmo de busca:

1. Gera uma configuração do modelo (uma solução)
2. Configura o modelo atual para atender aquela configuração.
3. Roda a simulação. Isto também pode ser chamado de avaliando a solução.
4. Gera os dados de saída para aquela simulação.
5. Enumera as soluções.
6. Gera uma nova solução baseada nos resultados de todas as soluções de avaliação.
7. Repete as etapas anteriores do passo 2 em diante.

O algoritmo acima repete-se até que o tempo do optimizer se esgota, avalia o número máximo de soluções, é parado pelo usuário, ou até que avalia todas possíveis soluções do modelo.

Definindo uma Otimização

Variáveis

As variáveis de otimização são as opções de seu modelo. Quando o optimizer gera uma solução para o seu modelo, é simplesmente definido cada uma das variáveis para aquele valor específico. Existem diferentes tipos de variáveis baseado na opção que está sendo representado. Não existe limite para o número de variáveis que uma otimização pode ter. Para adicionar um variável em uma otimização, adicione uma variável para um Scenario Table no Simulation Experiment Control. Se esta variável puder ser manipulada pelo optimizer, ela será automaticamente adicionada para a lista de variáveis para o optimizer. Se

necessário, o nome da variável será modificada para atender os requisitos de nomes das variáveis do optimizer.

Type (Type)	Description (Descrição)	Examples (Exemplos)
Continuous	Estas variáveis representam opções com uma ampla opções de valores. Qualquer valor dentro desta faixa é uma opção válida para o modelo (4.354, 6.0, -10.45, etc.).	Valores que podem ser afinados (como posição, comprimento, e tempos são frequentemente apresentados como variáveis contínuas)
Integer	Estas variáveis representam opções com uma ampla opções de valores. Qualquer valor inteiro como (-3, 5, 7, 25, etc.).	Quantidade de objetos discretos (número de pessoas, itens, etc.) são frequentemente representações de números inteiros
Discrete	Estas variáveis representam opções com uma ampla opções de valores e uma etapa entre possíveis valores. Somente valores que são exatamente n etapas do limite inferior são válidos.	Valores com etapas discretas (tamanho das peças, itens emparelhados, etc..) são representações de variáveis discretas.
Binary	Estes valores representam opções que tem somente dois possíveis valores: 0 e 1.	Valores que representam opções como yes/no, on/off, present/not present, etc., são usualmente representado como variáveis binárias.
Design	Estas variáveis representam opções que têm um intervalo, mas um valor mais elevado não representa "mais" ou "ainda mais". Em vez disso, ele representa apenas uma opção diferente. O otimizador não irão assumir que o aumento deste tipo de variável terá um efeito previsível no sistema.	Valores que representam opções como tipo de máquina, layout chão, ou estratégia de embalagem são geralmente representados como variáveis de projeto.
Permutation	Estas variáveis vêm em grupos, uma única variável permutação é inútil. Permutação variáveis podem ter um valor de 1 a n, onde n é o número de variáveis de permuta de um grupo específico. No entanto, cada valor é a garantia de ser distinto. Se existem três variáveis de permutação num único grupo, podem ter valores como 1, 2, 3 ou 3, 1, 2, que não pode ter valores como 1, 1, 2 ou 3,1, 3.	Essas variáveis são normalmente utilizados para representar uma ordem. Por exemplo, um percurso parcial pode ser representada por uma variável de permutação. Em uma configuração, a Estação 1 poderia ser o primeiro, Estação 2 segundo e Estação 3, o terceiro. Em outra configuração, Estação 3 pode ser o primeiro, a Estação 1 poderia ser segunda, e Estação 2 pode ser o terceiro.

Performance Measures

Performance measures são os resultados de uma simulação do modelo. Eles usualmente representam informações como custo, receita, produtividade, risco, tempo médio no sistema, utilização, etc... Não existe limite para o número de performance measures monitorado por uma otimização.

Todas as medidas de performance adicionada na guia Performance Measures no Simulation Experiment Control são automaticamente adicionadas ao optimizer. Se necessário, o nome da performance measure será modificada para atender os requisitos de nome das variáveis do optimizer.

Constraints (Restrições)

Restrições são uma forma de especificar condições adicionais que um dado modelo deve atender. Por causa da natureza da simulação, muitas restrições são naturalmente executadas pelo modelo. Ocasionalmente, no entanto, faz-se necessário aplicar algumas condições adicionais.

Restrições são expressões matemáticas que resultam em valores booleanos. Eles podem ser composto de variáveis, performance measures, literal values e funções matemáticas básicas. Exemplos de restrições válidas são mostrados abaixo:

```
processorPositionX < 30
numberOfEmployees * 500 > 0.5 * totalRevenue
0 < numberOfEmployees + numberOfTrucks < 30
```

Se uma determinada configuração anula estas restrições, é então considerado como inviável. O optimizer irá somente considerar a configuração como ótima se ela for realmente viável.

Uma restrição pode ter alguns valores separados por vírgula; como por exemplo:

```
processorPositionX < (30, 35, 40)
```

Se uma restrição como esta existe, o optimizer irá basicamente rodar a rotina de otimização para cada valor na lista, então reportar os resultados como um conjunto.

Nota: O gráfico exibido pelo optimizer pode não mostrar corretamente a lista de restrições existentes se as funções multi objetivos são definidas como searched. O optimizer irá corretamente executar a busca, mas um programa diferente (como o Excel) será necessário ser usado para visualizar os resultados.

Objectives (Função Objetivo)

Objetivos são funções usadas para determinar se uma configuração é melhor ou pior que a outra. Elas são muito similares as restrições, exceto que elas retornam um valor atual ao invés de um resultado booleano. A função objetivo também tem um direção; elas podem ser maximizada ou minimizada. Por exemplo, uma função objetivo chamada "cost" pode ser algo como:

```
machineCount * 5000 + throughput * 50
```

Se o custo for minimizado, o optimizer irá procurar por soluções dentro do menor custo. Se o custo for maximizado, o optimizer irá procurar por soluções com o maior custo. É muito importante configurar a direção correta para cada objetivo.

É possível também ter múltiplas funções objetivo. Elas podem ser balanceada e combinada dentro de um simples objetivo; o resultado desta busca será uma solução ótima. Elas podem ser buscadas simultaneamente. Isto gera um conjunto de soluções ótimas, onde cada solução representa o melhor trade-off entre os objetivos; melhorando qualquer uma das funções objetivo irá fazer com que as outras piorem. Este conjunto de soluções é também chamado uma configuração de Pareto, ou Pareto frontier.

Other Optimization Options (Outras opções de Otimização)

Em razão do optimizer precisar rodar uma simulação para avaliar cada solução, existem muitas opções sobre estas simulações que pode ser configuradas, conforme segue:

- O tempo de rodar a simulação por avaliação da solução
- O warmup time do modelo por solução de avaliação.
- O número de replicações por solução.
- Os tipos de dados que devem ser salvos por replicação.

Existem muitas opções sobre o como o optimizer propriamente pode ser configurado, conforme segue:

- O wall time – a quantidade do tempo real que o optimizer irá gastar procurando por uma solução ótima.
- O número máximo de soluções para avaliar
- O uso de cenários de experiment como soluções

Rodando uma Otimização

Uma vez que a otimização foi apropriadamente definida, a parte mais difícil foi realizada. Você pode clicar no botão **Optimize** na guia Optimizer Run do Experiment Simulation Control. Neste momento, o Flexsim irá usar o mesmo recurso encontrado no experimenter para avaliar várias soluções simultaneamente. Isto permitirá ao FlexSim rapidamente fazer buscas através das soluções geradas, assim como fornecer um feedback real sobre o progresso da otimização.

Interpretando os Resultados

Quando o optimizer finalizar, um (ou várias) soluções serão marcadas como soluções ótimas. A menos que o optimizer avaliou todas as possíveis soluções (o qual é geralmente impossível), e então as soluções ótimas são simplesmente as melhores soluções que o optimizer encontrou. Eles **não são garantias** de ser absolutamente a melhor solução. Elas são apenas garantias de serem a melhor solução do que todas as demais soluções que foram encontradas.

O verdadeiro valor do optimizer, no entanto, será perdido se você somente olhar para aquelas poucas soluções ótimas. Como o optimizer busca por soluções ótimas, ele cria um banco de dados de cada solução e seus efeitos nos resultados do modelo. Este banco de dados pode ser usado (com a aplicação apropriada de técnicas estatísticas) para encontrar relações previamente desconhecidas entre as variáveis do modelo, ou para determinar quais restrições foram mais restritiva. Usando estes dados apropriadamente, você pode ter um novo insight do comportamento do modelo.

Under the Hood

Flexsim usa a ferramenta OptQuest da OptTek Systems, Inc. OptQuest é uma ferramenta confiável, padrão na indústria de otimização e que usa os últimos algoritmos revolucionários para efetivamente buscar as melhores soluções de um modelo.

Exemplo do Experimenter/Optimizer

Introdução

Este tutorial demonstra o uso do experimenter e do optimizer, presente no Flexsim. Estas ferramentas podem ajudar a responder questões “e se” para um modelo. E se a estação de processamento de for substituída diferentemente? E se mais profissionais forem contratados para um determinado turno? E se as ordens destes processos forem alteradas? O Experimenter permite você testar diferentes versões de seu modelo e avaliar a performance de cada um. Alternativamente, o optimizer pode automaticamente buscar através de muitas diferentes soluções para o seu modelo, procurando pela melhor. São ambas ferramentas poderosas que possibilitam você a aprender a buscar melhorias para o seu sistema.

Nota: O uso do optimizer requer uma licença separada chamada OptQuest. Contate a FlexSim para mais informações sobre como obter esta licença.

O que você irá aprender

Este tutorial irá guiar você através das seguintes etapas:

- Criação de um experimento
- Rodar um experimento
- Visualizar o resultado de um experimento
- Configuração e otimização
- Rodar uma otimização
- Visualização dos resultados de otimização

Ao longo do caminho, nós iremos mostrar as definições de termos correlatados, como:

- Experiment variables (Variáveis de experimento)
- Performance measures (Medidas de performance)
- Scenarios (Cenários)
- Optimizer variable types (Tipos de variáveis do optimizer)
- Constraints (Restrições)
- Objectives (Objetivos)
- Solutions (Soluções)

Construção do Modelo passo a passo

Para este tutorial, vamos examinar uma situação bem simples. Um operador, precisa carregar um item do source para um equipamento (processor). Após o item ser processado, o operador deve levar o item para o segundo equipamento (processor), que leva mais tempo para processar que o primeiro. Após o segundo equipamento processar o item, o operador, precisa levar o item para o sink. Vamos agora supor que nós queremos maximizar a produtividade (que também está ligado a receita) deste sistema ao ajustar a posição dos equipamentos (processors). Se cada equipamento pode ser movimentado 03 metros para cima a direita ou a esquerda, onde deve ser cada um posicionado? Seria muito difícil intuitivamente saber como colocar cada equipamento de forma a maximizar a produtividade. Para resolver este problema de forma mais precisa, nós teremos que usar o experimenter e o optimizer. Agora, obviamente este é um cenário drasticamente simplificado, mas frequentemente na vida real você terá situações onde você quer ver como vários layouts podem afetar a produtividade no geral. Esta é uma implementação bem simples de um experimento.

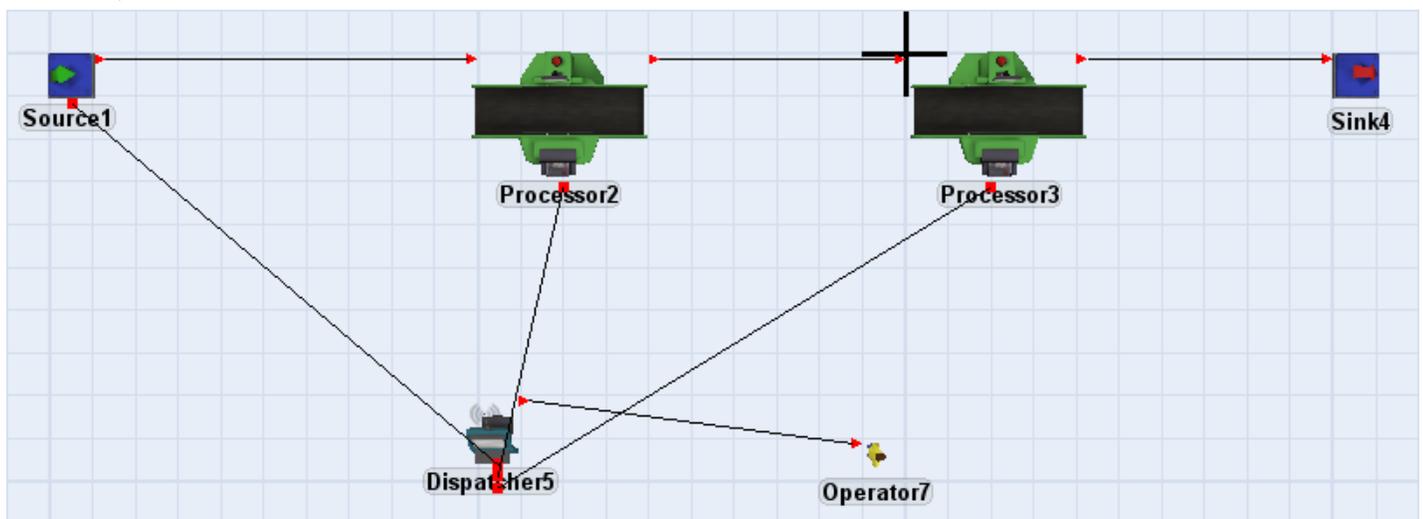
Etapa 1: Construindo o Modelo

Modelo

Crie um novo modelo usando segundos (**Seconds**), metros (**Meters**), e litros (**Liters**) como unidades.

Objetos

Crie um **Source**, dois **Processors**, um **Sink**, um **Dispatcher**, e um **Operator**. Configure-os de acordo com o layout Guiaixo.



Posições

Configure as localizações destes objetos de acordo com a tabela abaixo:

Object	X Position	Y Position
Source1	-20.0	0.0
Processor2	-10.0	0.0

Processor3	0.0	0.0
Sink4	10.0	0.0
Dispatcher5	N/A	N/A
Operator7	N/A	N/A

Dispatcher5 e **Operator7** não precisam estar em um lugar particular, mas eles não devem estar alinhados com o restante dos objetos.

Lógica

Configure a seguinte lógica:

- Configure o **Source1**, **Processor2**, e **Processor3** para usar um transporte (**Use Transport**) (disponível na janela Quick Properties).
- Configure o tempo de processamento do equipamento **Processor2** para `normal(10, 2)` (também disponível na janela Quick Properties).
- Configure o tempo de processamento do equipamento **Processor3** para `normal(12, 3)`.
- Configure a posição de resetar do **Operator7** para sua posição atual.

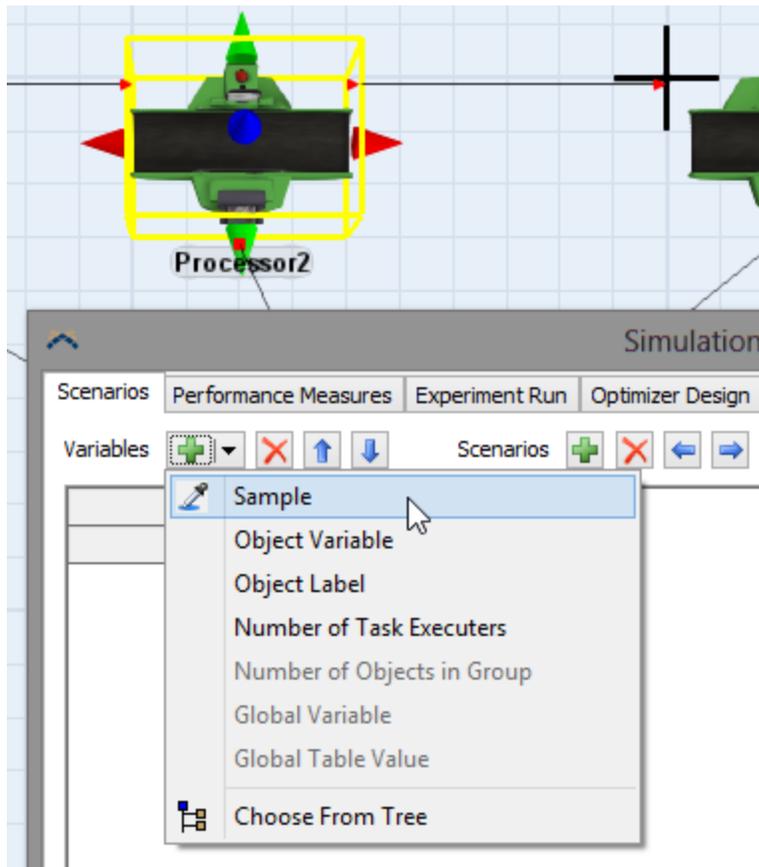
Etapa 2: Defina o Experimento

O Experimenter encontra-se no menu Statistics do FlexSim. Muitas das funcionalidades do Optimizer já estão presentes no experimenter.

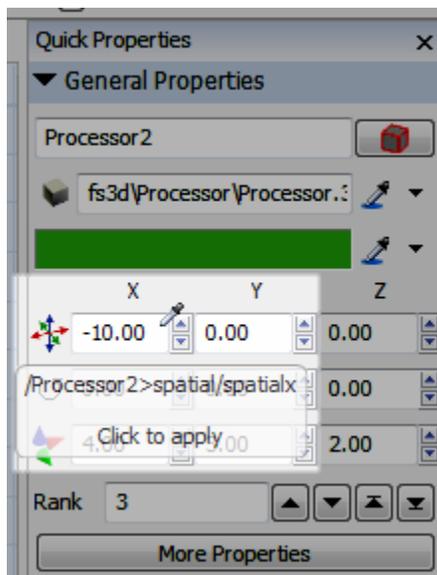
Crie Variáveis

Abra a janela do Experimenter. Posicione a janela e então você poderá ver os equipamentos (processors) no modelo, bem como a janela do Experimenter. Então, para o Processor2 e o Processor3 siga as etapas abaixo:

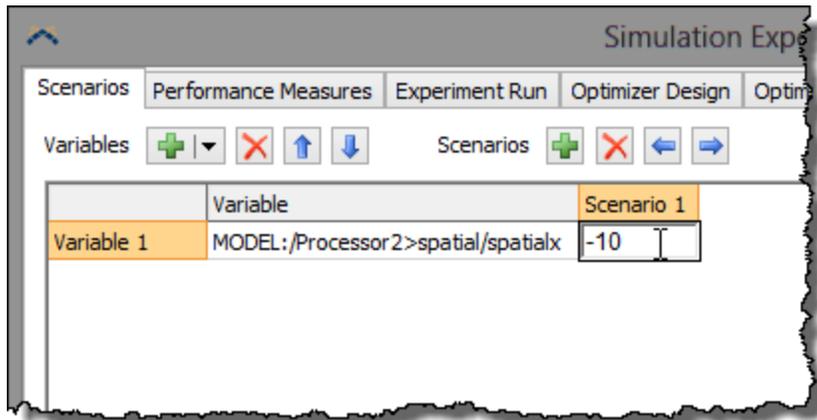
1. Clique sobre o processor na superfície de simulação em 3D.
2. Clique sobre a flecha em preto no botão .
3. Selecione **Sample** do menu popup. Isto coloca o curso no modo sample conforme imagem abaixo.



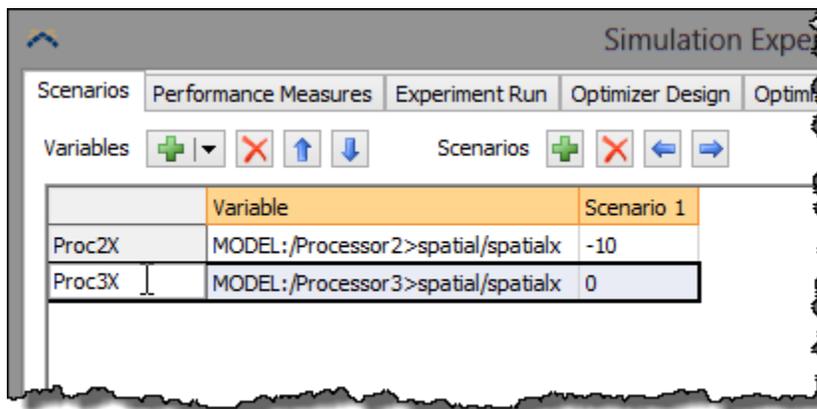
- Coloque o sample sobre o campo da posição X na janela Quick Properties simplesmente clicando sobre o campo. Isto adiciona uma nova variável de experimento.



- Configure o valor do Scenario 1 para a nova variável, dando um duplo clique sobre a célula e entrando com o novo valor.



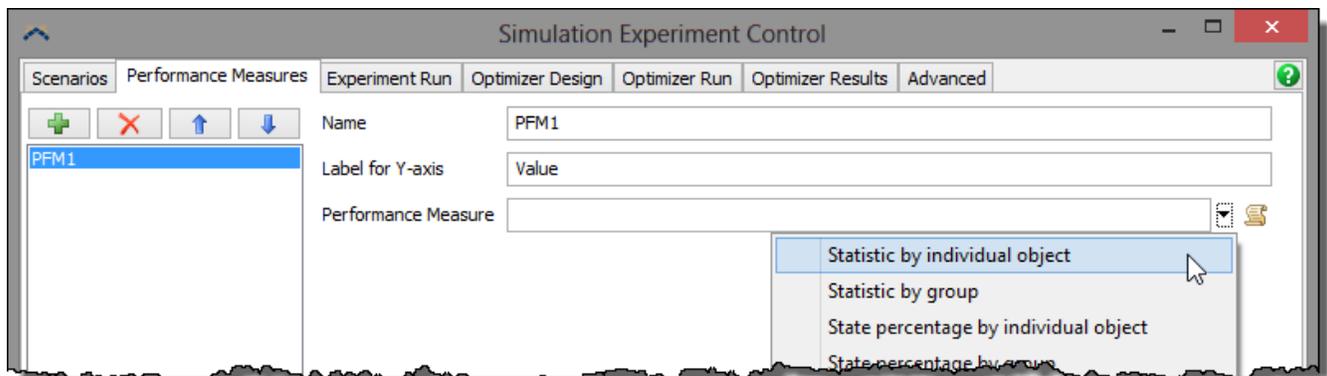
- Configure o nome da variável ao dar um duplo clique sobre o nome atual. Configure o nome do **Proc2X** para **Processor2** e **Proc3X** para **Processor3**.



Crie as Performance Measures (Medidas de Performance)

Vá para a guia Performance Measures na janela do Experimenter. De lá:

- Clique no botão  para adicionar uma nova performance measure.
- Nomeie a nova performance measure com produtividade ou **Throughput**.
- Clique no botão  e selecione a primeira opção. Uma janela popup irá se abrir.



- Selecione **Sink4** para o objeto e **Input** para estatística (statistic). Para selecionar estes objetos, simplesmente digite `/Sink4` (sem aspas) no campo do objeto ou faça o seguinte:

- Clique no botão 
- Selecione **Sink4** da lista de objetos.
- Clique no botão **Select** quando você terminar.

Configure o Experimento

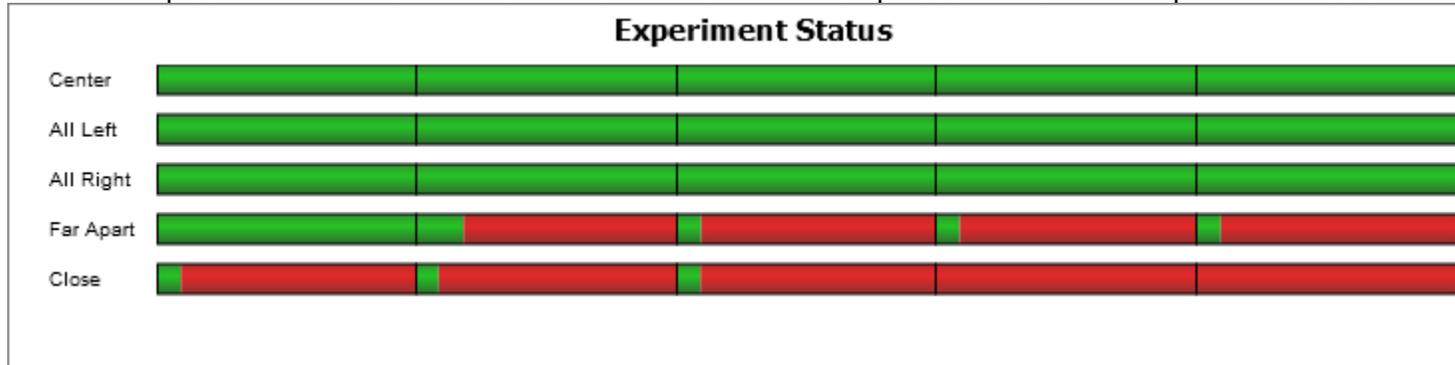
Agora que nós criamos as variáveis e performance measures, iremos configurar alguns cenários para nosso experimento. Retorne para a guia Scenarios:

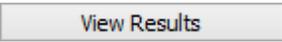
1. Crie 5 cenários ao clicar no botão Scenarios  4 vezes.
2. Entre com os nomes e valores dos cenários conforme abaixo:

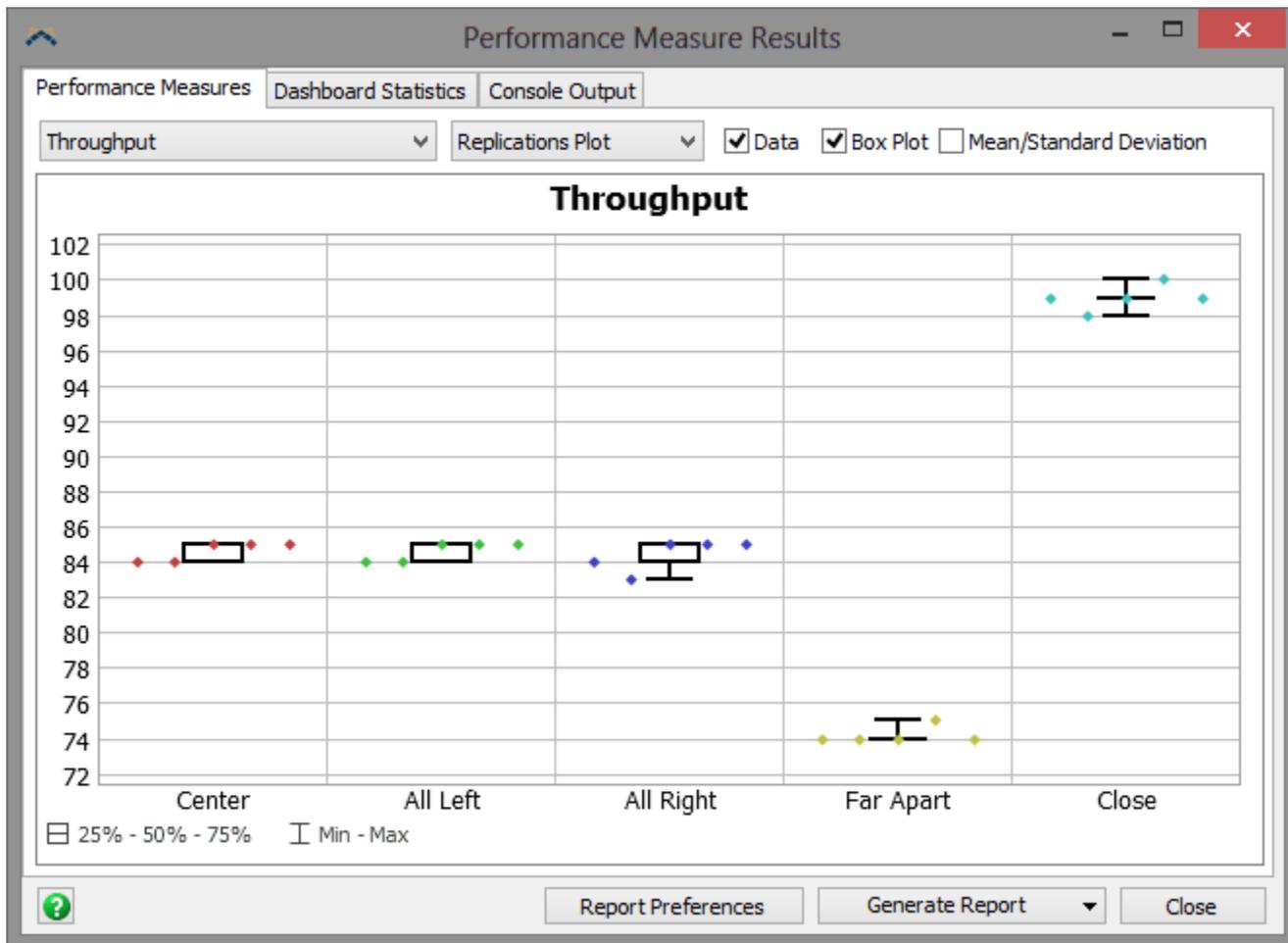
	Variable	Center	All Left	All Right	Far Apart	Close
Proc2X	MODEL:/Processor2>spatial/spatialx	-10	-13	-7	-13	-7
Proc3X	MODEL:/Processor3>spatial/spatialx	0	-3	3	3	-3

Rode o Experimento

Vá para a guia Experiment Run. Clique sobre o botão . Cada cenário irá rodar 5 vezes e o resultado da medida de performance produtividade será coletado no final de cada rodada. A janela de status irá mostrar qual cenário/replicação está atualmente sendo rodado. FlexSim irá rodar múltiplos cenários simultaneamente se o seu computador tiver um cpu multi core.



Uma vez que o experimento terminar, clique no botão . Isto abrirá uma janela onde você pode visualizar dados e medidas de performance para o cenário. Neste exemplo, nos temos apenas uma medida de performance, mas se você tiver vários, você poderá ver os resultados para cada um nesta janela. Existem diversas opções de como mostrar os dados, incluindo o Replications Plot, Frequency Histogram, Correlation Plot (para examinar correlações entre várias performance measures), Data Summary e Raw Data.



Neste experimento, o melhor cenário foi o cenário “Close”, o qual apresentou em média 99 peças de produtividade. O pior cenário foi o “Far Apart”, o qual apresentou em média 75 peças de produtividade.

Otimização.

Além de usar o Experimenter para explicitamente definir cenários, você pode usar o Optimizer. O Optimizer irá automaticamente criar cenários e então testar estes cenários, tentando encontrar um cenário que melhor atende ao objetivo.

Configure o Optimizer

Vá para a guia Optimizer Design na janela Experimenter. Você verá que as duas variáveis criadas inicialmente, estão aparecendo; isto porque o experimenter e o optimizer agora compartilham variáveis.

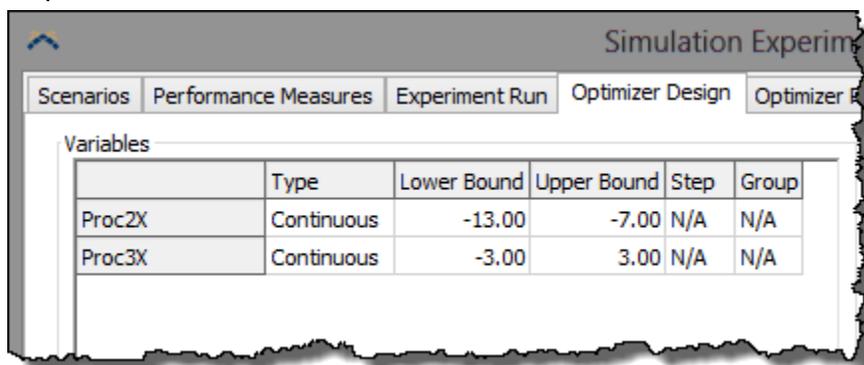
The figure shows the "Simulation Experiment" window with the "Optimizer Design" tab selected. It displays a table of variables for optimization.

Variable	Type	Lower Bound	Upper Bound	Step	Group
Proc2X	Continuous	0.00	10.00	N/A	N/A
Proc3X	Continuous	0.00	10.00	N/A	N/A

No entanto, o optimizer precisa de informações adicionais sobre estas variáveis. Especificamente, você necessariamente precise especificar:

- **Type** - O tipo de variável dita os tipos de valores que são possíveis para uma dada variável. Variáveis contínuas podem ter qualquer valores entre os limites inferiores e superiores.
- **Lower Bound** - O limite inferior especifica o menor valor possível que o optimizer pode ajustar a variável.
- **Upper Bound** - O limite superior especifica o maior valor possível que o optimizer pode ajustar a variável.
- **Step** - Para variáveis discretas e de design, o step na tela Guiaixo especifica a distância entre possíveis valores, iniciando do limite inferior.
- **Group** - Para variáveis de permutação, o grupo especifica qual o conjunto de variáveis de permutação esta variável particular pertence.

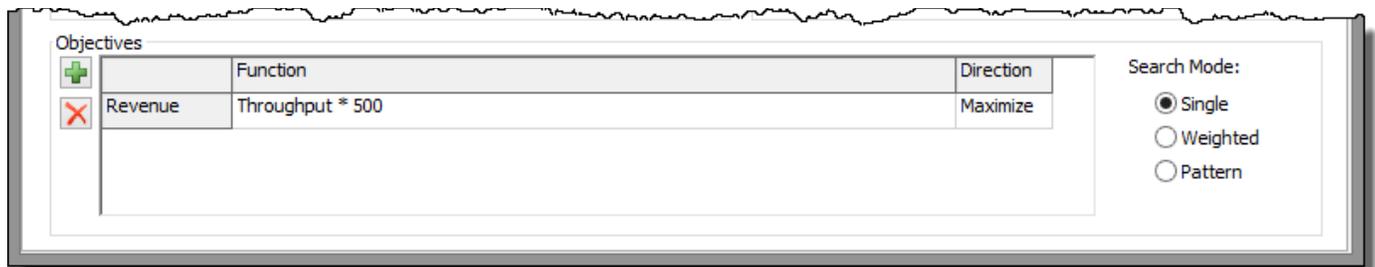
Para esta otimização, nós queremos permitir com que o equipamento (processor) se movimente 3 metros para ambos os lados. Uma vez que nós não estamos limitados em posições específicas dentro dos ranges possíveis, ambas as variáveis de posição são **contínuas (Continuous)**. No entanto, nós precisamos configurar os limites inferiores e superiores de cada variável. Para editar os valores na tabela, dê um duplo clique sobre a célula de interesse e entre com o novo valor. Entre com os valores conforme mostrado abaixo:



The screenshot shows the 'Optimizer Design' tab in the 'Simulation Experiment' software. It features a table with the following data:

	Type	Lower Bound	Upper Bound	Step	Group
Proc2X	Continuous	-13.00	-7.00	N/A	N/A
Proc3X	Continuous	-3.00	3.00	N/A	N/A

A etapa final de configuração é configurar a função objetivo. A função objeto não está presente e encontra-se com o campo em branco. Edite o nome do campo Objectives como **Receita** ou **Revenue**. Então clique sobre o campo Function. Um botão  irá aparecer do lado direito. Clique sobre este botão para fazer aparecer uma listagem de todas as variáveis e performance measures. A função objetivo é um valor derivado de qualquer um destes valores. Selecione **produtividade** ou **Throughput**; isto irá adicionar esta performance measure para a função objetivo, e coloque o cursor exatamente no final. Adicione o texto * 500 para que a **Receita** ou **Revenue** seja igual a $Throughput * 500$. Deixe marcado no campo direction como **Maximize**, porque nós queremos maximizar a **Receita** ou **Revenue**. Visto que nós temos apenas um objetivo e conseqüentemente apenas 01 função objetivo o search mode deve manter-se como **Single**.



The screenshot shows the 'Objectives' configuration window. It contains a table with the following data:

	Function	Direction
Revenue	Throughput * 500	Maximize

Search Mode:

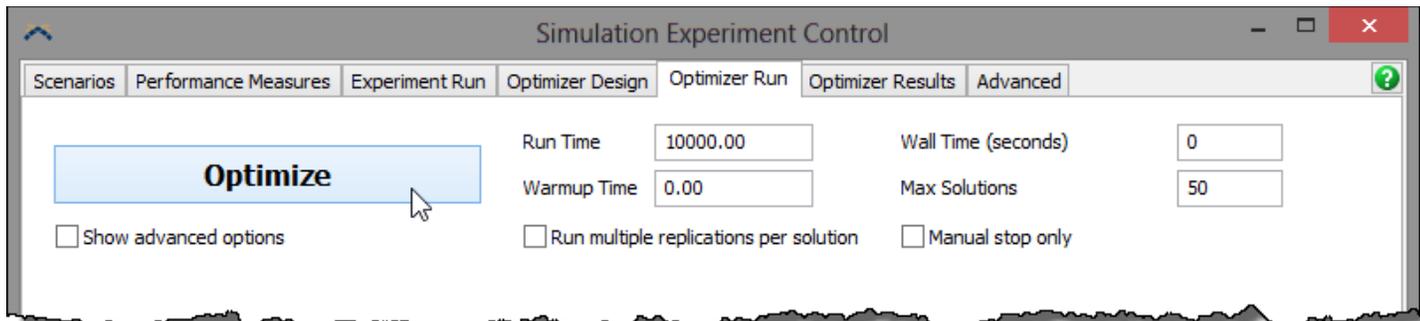
- Single
- Weighted
- Pattern

Etapa 3: Rode a Otimização

Vá para a guia Optimizer Run da janela Experiment Control. Então:

1. Configure o **Run Time** para 10000. Isto representa quanto tempo o optimizer irá rodar cada configuração do modelo (chamada solução) para avaliá-la.

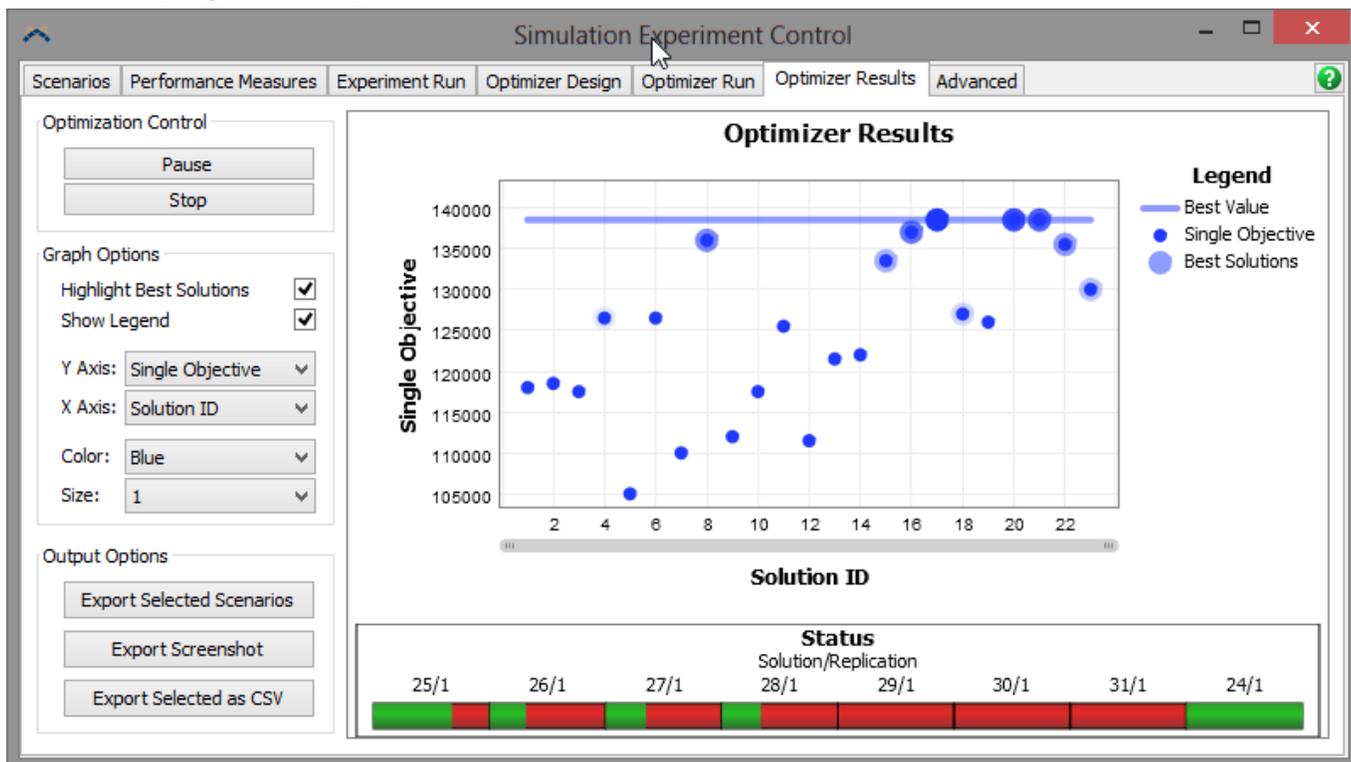
- Configure o **Wall Time** para 0. Isto usualmente significa quanto tempo o optimizer rodará no tempo real. O valor 0 significa que não tem limite de tempo.
- Configure o **Max Solutions** para 50. Isto significa que o optimizer tentará não mais que 50 diferentes soluções para encontrar a solução ótima.
- Clique sobre o botão **Optimize**.



A janela do Experimenter irá automaticamente pular para a guia Optimizer Results. O optimizer começa a rodar através de seguidos loops:

- Determine os valores para **Proc2X** e **Proc3X**.
- Rode o modelo para estes valores com 10000 segundos
- Avalie a performance measure.
- Calcule a função objetivo.
- Rankei as soluções.
- Use a informação desta solução para criar uma nova solução – novos valores para **Proc2X** e **Proc3X**.
- Repita a partir da etapa 2.

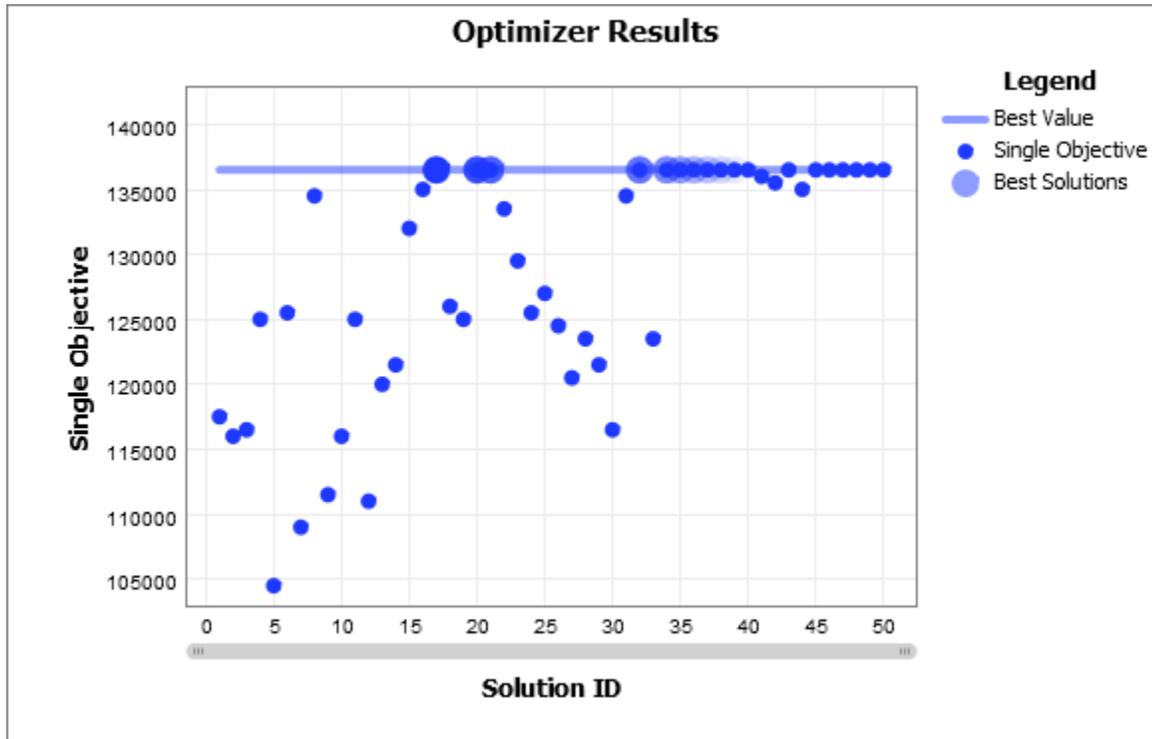
Por causa do optimizer compartilhar da capacidade multi-threaded do experimenter, ele pode avaliar várias soluções ao mesmo tempo. Conforme o progresso da otimização, o gráfico Optimizer Results irá atualizar e mostrar o progresso do optimizer.



Uma vez que o optimizer avalia 50 soluções, uma mensagem irá aparecer informando o porque do optimizer ter parado. Neste caso, irá dizer que o optimizer atingiu o número máximo de soluções. Se algo acontecer de errado, uma mensagem informará o motivo do erro.

Etapa 4: Analise os Resultados

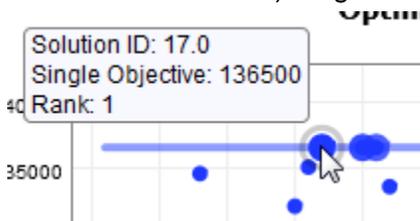
Quando o Optimizer finalizar, o gráfico Optimizer Results deverá parecer com algo semelhante a isso:



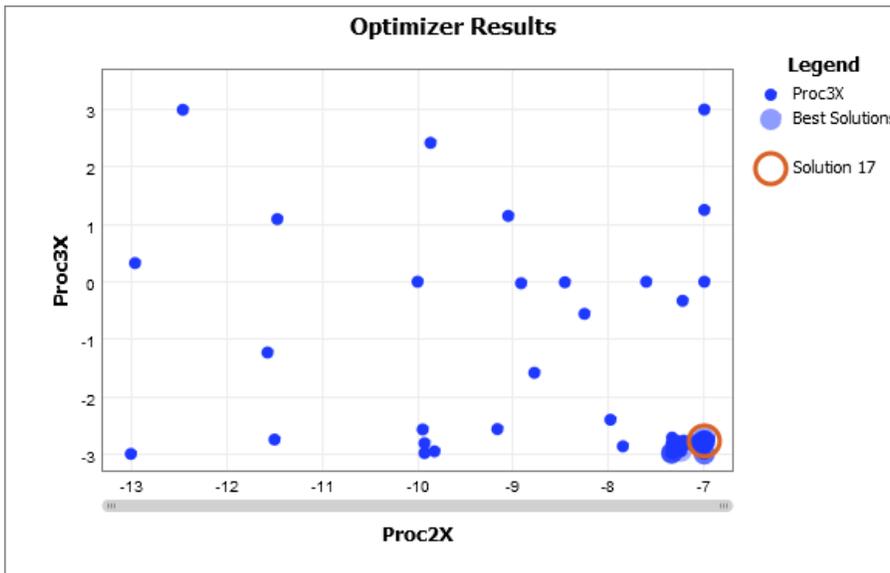
O eixo Y é chamado "Single Objective". Para este exemplo, é o sinônimo de **Receita** ou **Revenue**. A melhor solução está em destaque. Os círculos maiores e mais escuros representam as melhores soluções. Para uma simples função objetivo, as 10 primeiras soluções estão marcadas deste forma. Enquanto a otimização progredia, o optimizer vai buscando as melhores soluções até encontrar as melhores. Isto é chamado de convergência, e é uma forma de dizer se uma otimização é considerada finalizada; se o valor objetivo não melhorou até então, é bem provável que não irá melhorar com buscas adicionais, e a melhor solução atual deve ser usada.

Respondendo a Pergunta Original

O objetivo desta otimização foi descobrir onde colocar os dois equipamentos (processors). Nós podemos agora facilmente encontrar a solução para esta questão. Passe o mouse sobre a melhor solução (o círculo em azul mais escuro) no gráfico; uma pequena janela popup irá aparecer.



Clique sobre esta solução para selecioná-la. Agora, no painel Graph Options, altere o valor do eixo Y para Proc3X e do eixo X para Proc2X.



A melhor solução (e todas as outras melhores soluções) é encontrada onde o Proc2X é maior, e onde o Proc3X é menor. Lembre-se que todas as 10 soluções no topo do gráfico produzem os mesmos resultados; neste caso, tendo dois equipamentos (processors) logo a direita e próximo um dos outros é a melhor configuração para este modelo.

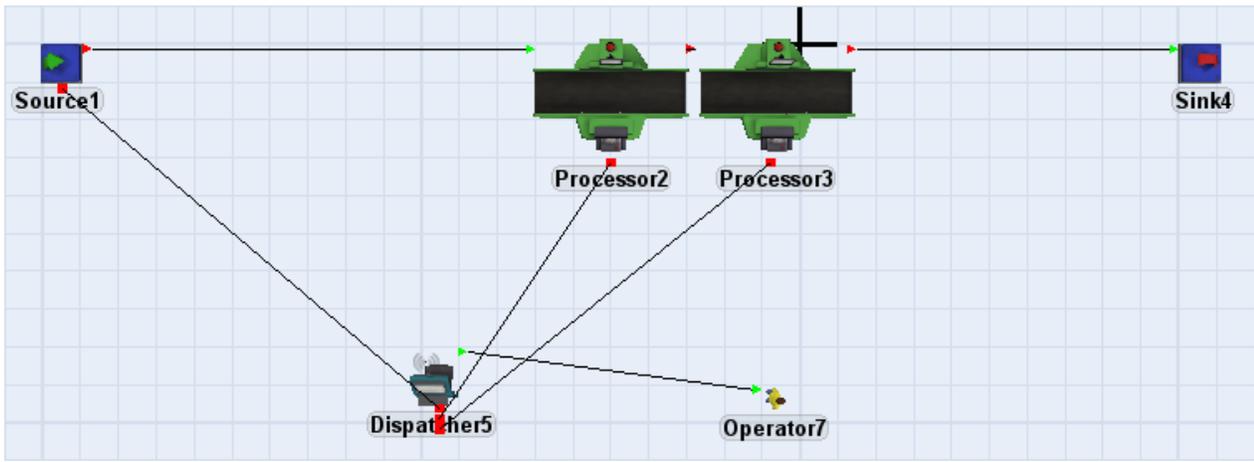
Configure o modelo para a melhor solução

Às vezes pode ser útil configurar o modelo para atender a melhor solução. Para fazer isso, clique no botão **Export Scenarios**. Isto levará todas as soluções selecionadas e criará os cenários de experimento para eles. Retorne para a guia Scenarios na janela Experimenter para visualizar a nova solução.

The screenshot shows the "Simulation Experiment Control" window with the "Scenarios" tab selected. Below the tabs, there are controls for "Variables" and "Scenarios". A table displays the configuration for two variables: Proc2X and Proc3X.

	Variable	Center	All Left	All Right	Far Apart	Close	Solution 17
Proc2X	MODEL:/Processor2>spatial/sp	-10	-13	-7	-13	-7	-7.00
Proc3X	MODEL:/Processor3>spatial/sp	0	-3	3	3	-3	-2.77

Agora, da opção drop-down logo a direita "Choose default reset scenario", selecione o novo cenário. Então resete o modelo para aplicar estes valores para o modelo em 3D.



Visualização da Modelagem

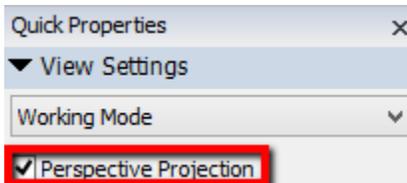
1. Visualização em Perspectiva/Ortográfica
2. O recurso Tree
3. Travel Networks
4. Utilitário de Modelagem
 - Library Icon Grid
 - View Settings (Configurações de Visualização)
 - Light Source Editor
 - Quick Properties
 - Edit Selected Objects
 - Find Objects
 - Views
 - Groups
 - Model Layouts
 - Measure / Convert
 - Command Helper
 - Keyboard Layout

Visualização em Perspectiva/Ortográfica

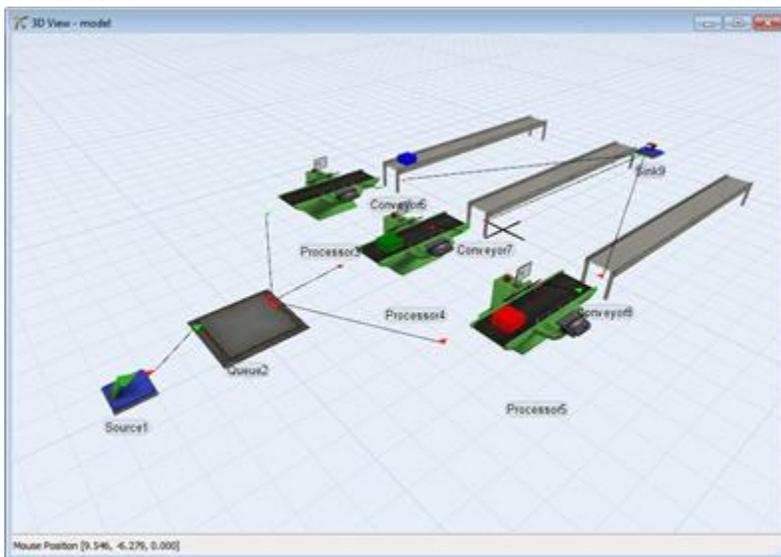
Tópicos

- Moving Around in the View
- Movimentando Objetos
- Dimensionando o tamanho dos Objetos
- Conectando Objetos
- Criando e Editando um seleção de conjuntos
- Editando as propriedades dos Objetos
- Right-Click Menu

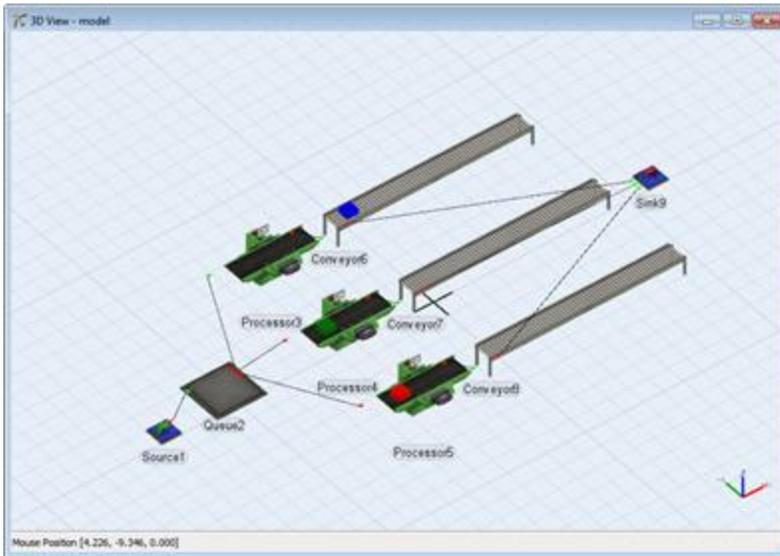
O modo de visualização em 3D pode ser alternado entre os modos em perspectiva e ortográfico, não selecionando nenhum objeto no modelo em 3D e selecionando/desmarcando a opção em **Perspective Projection** da janela Quick Properties.



A projeção em perspectiva confere ao modelo um sentimento em 3D mais realístico.



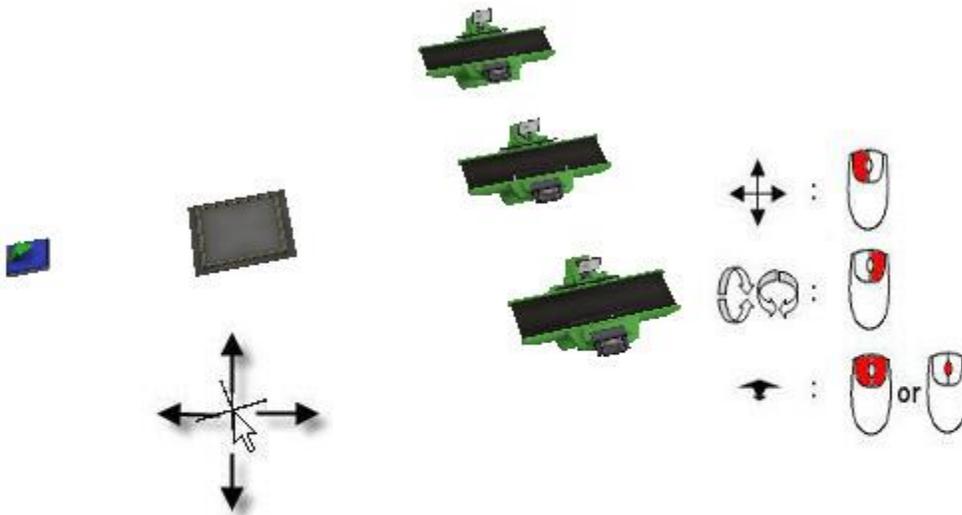
Enquanto a projeção ortográfica não apresenta profundidade de visualização.



Para mais informações sobre a configuração de visualização do modelo em 3D, veja a página de configuração de visualização (View Settings).

Movimentando no campo de Visualização

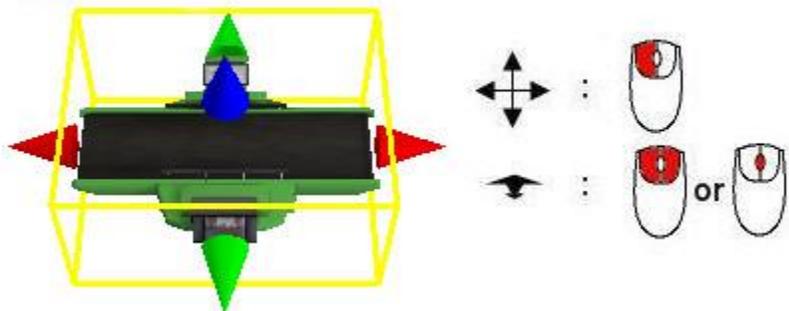
Para se deslocar no campo de visualização do modelo clique e mantenha o botão do mouse pressionado, e arraste o mouse em torno do campo de visualização. Para rotacionar o campo de visualização, clique e segure o botão direito do mouse no campo de visualização e arraste o mouse em diferentes direções. Para zoom in e out, mantenha ambos os botões direito e esquerdo do mouse pressionado e mova o mouse para cima e para baixo. Você também pode dar zoom in e out, deslocando para cima ou para baixo, o botão do mouse



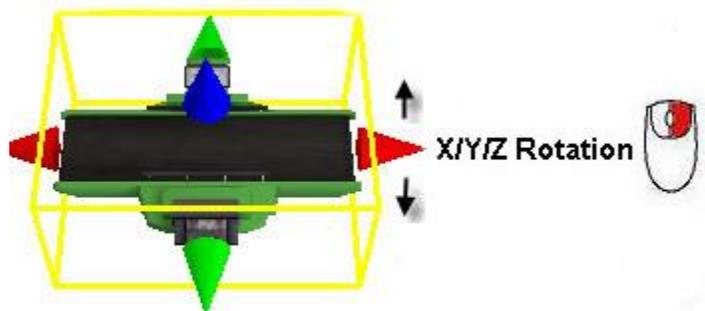
Quando você estiver trGuialhando no modo em perspectiva, você pode mover o cursor do mouse para planar para a frente, até planar para trás, da esquerda para rodar para a esquerda e para a direita para rodar para a direita. Para sair do modo fly-through, pressione F8 novamente. Este estilo de navegação pode levar um pouco de tempo até que você adquira prática para dominar, mas você deve experimentar. Se você planar muito longe e perder de vista o seu modelo, você pode parar o fly-through clicando em F8, e em seguida, clique com o botão direito do mouse no espaço destinado para desenvolvimento do modelo, aponte para **View** e clique em **Reset View**.

Movimentando os objetos

Para movimentar um objeto no plano X/Y, clique sobre um objeto e mantenha o botão do mouse pressionado., então, arraste o objeto para o local desejado. Para movimentar o objeto na direção z, clique sobre o objeto e rode o botão do mouse para cima ou para baixo. Você também pode manter pressionado o botão direito e esquerdo do mouse e arrastar o objeto para cima ou para baixo.



Para rotacionar um objeto, clique sobre um dos três eixos do objeto com o botão direito do mouse pressionado e arraste o mouse para cima e para baixo.



Alterando o tamanho dos objetos

Para alterar o tamanho do objeto clique sobre uma das três setas dos eixos do objeto com o botão esquerdo do mouse e arraste o mouse na direção da seta. Você pode dimensionar o tamanho dos objetos em todos os três eixos de 5% em 5% ao pressionar o Ctrl+L para aumentar e Ctrl+K para diminuir. Você pode alterar o tamanho do objeto em todos os três eixos de uma vez ao manter pressionado o teclado Alt. e clicando sobre um dos três eixos para movimentar o mouse de acordo com o tamanho que deseja para o objeto. Se as três setas não estiverem sendo mostradas, vá no menu Edit e selecione Resize e Rotate Objects.

Conectando os Objetos

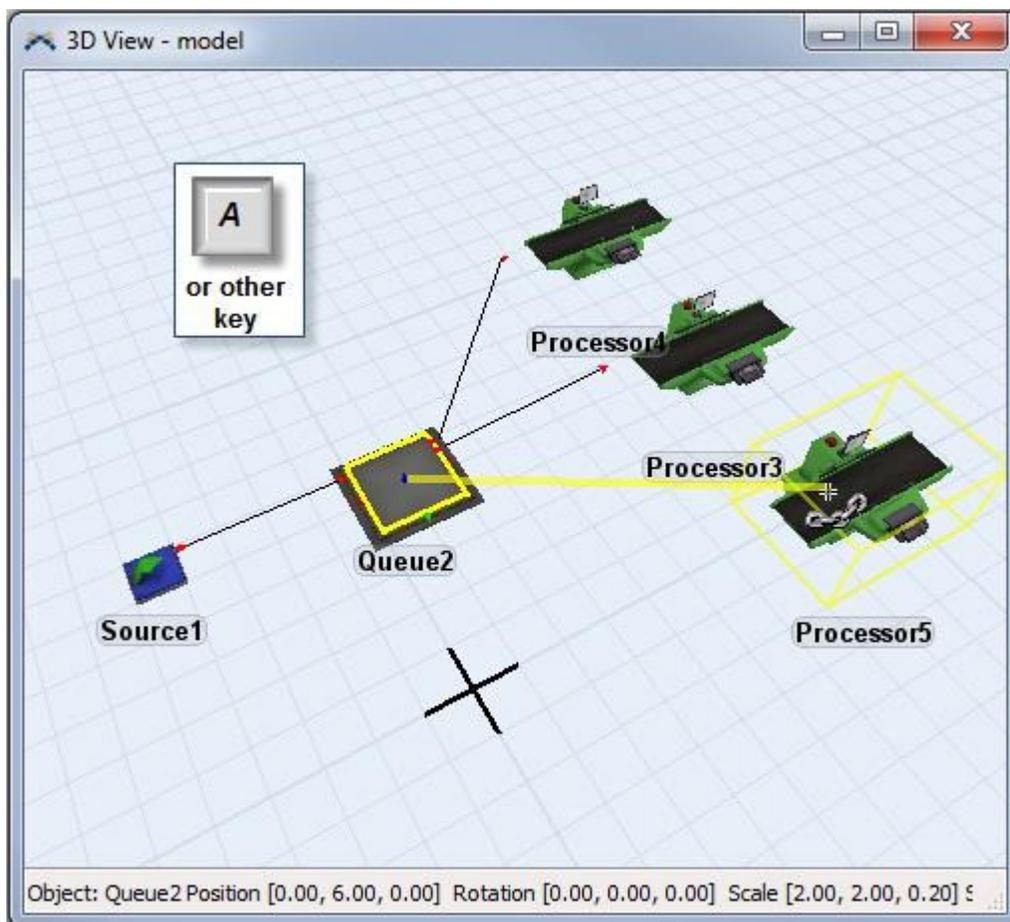
Portas são criadas e conectadas de duas maneiras:

1) Ao clicar em um objeto e arrastando em direção a outro objeto, mantendo pressionada diferentes letras do teclado. Se a letra "A" é pressionada enquanto clicar e arrastar, uma porta de saída será criado no primeiro objeto e uma porta de entrada será criada no segundo objeto. Estas duas novas portas serão, então, conectadas automaticamente. Mantendo pressionada a tecla "S" vai criar uma porta central, em

ambos os objetos e ligar as duas novas portas. As ligações são quebradas e portas eliminadas, mantendo-se o "Q" selecionado para as portas de entrada e saída e a tecla "W" para as portas centrais. A tabela a seguir mostra as letras de teclado usados para fazer e quebrar os dois tipos de ligações de portas.

2) Ao entrar no modo de conexão mantendo botão A do teclado pressionado . Uma vez no modo de conexão, há um par de maneiras de fazer uma conexão entre dois objetos. Você também pode clicar em um objeto, em seguida, clique em outro objeto, ou você pode clicar e arrastar de um objeto para o próximo como no primeiro método. De qualquer forma, tenha em mente que a direção do fluxo de uma conexão é dependente da ordem em que você faz a conexão. Fluxo vai desde o primeiro objeto para o segundo objeto.

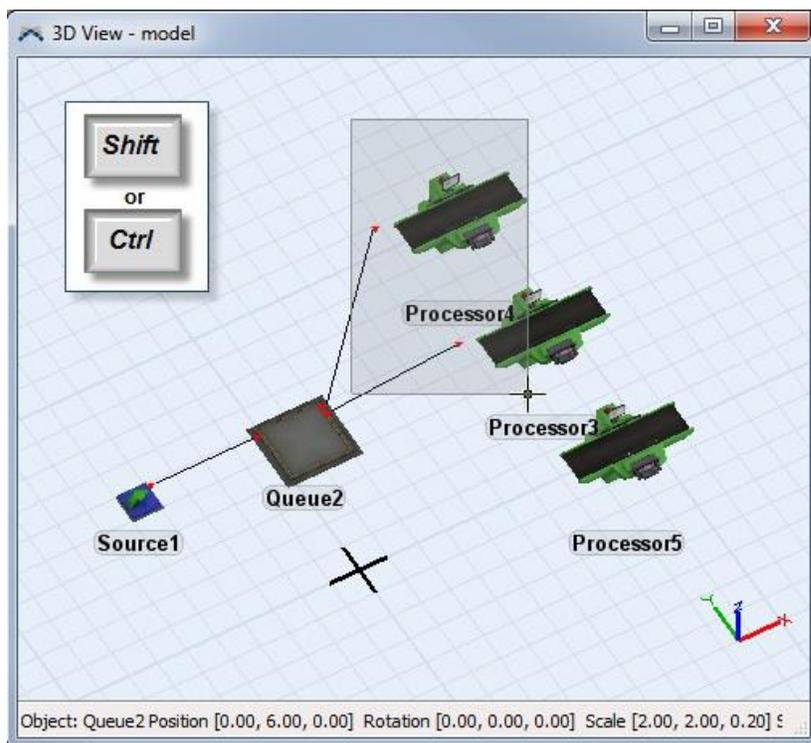
Alíás, as ligações podem ser quebradas, mantendo o botão Q do teclado pressionado  e em seguida, clicar ou arrastar de um objeto para outro, da mesma maneira como quando realizando conexão entre eles. Conexões de portas centrais podem ser realizadas mantendo o botão S do teclado pressionado e em seguida, clicar ou arrastar de um objeto para outro. Para desfazer a conexão da porta central, basta manter o teclado em W pressionado e desfazer a conexão de acordo com o sentido que deve ser desfeito.



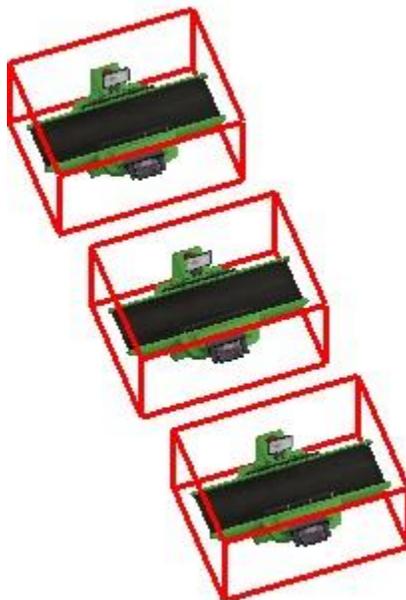
Criando e editando um conjunto de objetos selecionado

Você pode criar conjuntos de seleção para ter operações aplicadas a todo um conjunto de objetos. Para adicionar objetos ao conjunto, segure a tecla Shift ou Ctrl apertadas e arraste uma caixa ao redor dos objetos que você deseja selecionar. Segurando Shift redefine o conjunto de seleção, enquanto segurar Ctrl pressionado, adiciona ou remove os objetos ao conjunto de seleção. Você também pode segurar a tecla Shift ou Ctrl pressionada e clicar nos objetos, em vez de arrastar uma caixa em torno deles.

Outra maneira de criar conjuntos de seleção é usar o modo New Selection na barra de ferramentas principal. Este modo funciona da mesma forma segurando a tecla Shift pressionada. O modo Toggle selection funciona da mesma forma que manter a tecla Ctrl pressionada.



Os objetos selecionados em conjunto, são visualizados com uma linha vermelha ao redor deles



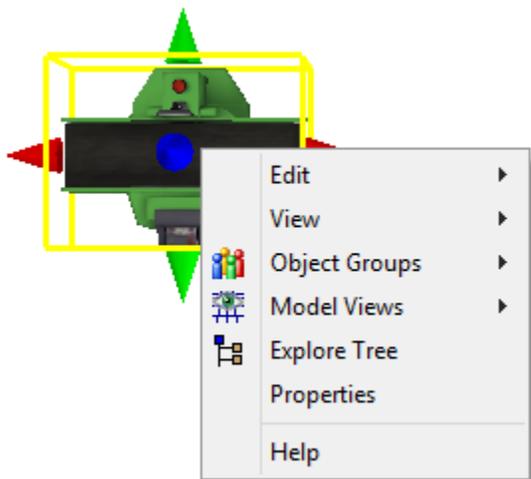
Uma vez que você selecionou o conjunto de objetos, movimentar, rotacionar e redimensionar irá fazer com que outros objetos da seleção sofram as mesmas ações em conjunto. Você pode realizar diversas operações no conjunto de objetos selecionados para editar suas propriedades.

Editando as propriedades dos objetos

Para editar as propriedades dos objetos, dê um duplo clique sobre o objeto ou clique com o botão direito do mouse e selecione "Properties" do menu. Você também pode editar as propriedades dos objetos, ao acessar a janela Quick Properties que fica ao lado direito de sua tela.

Menu acessado clicando com o botão direito do mouse sobre os objetos.

Clique com o botão direito do mouse sobre um objeto, independentemente do modo de visualização, para que mostre o menu de opções conforme Guiaixo.

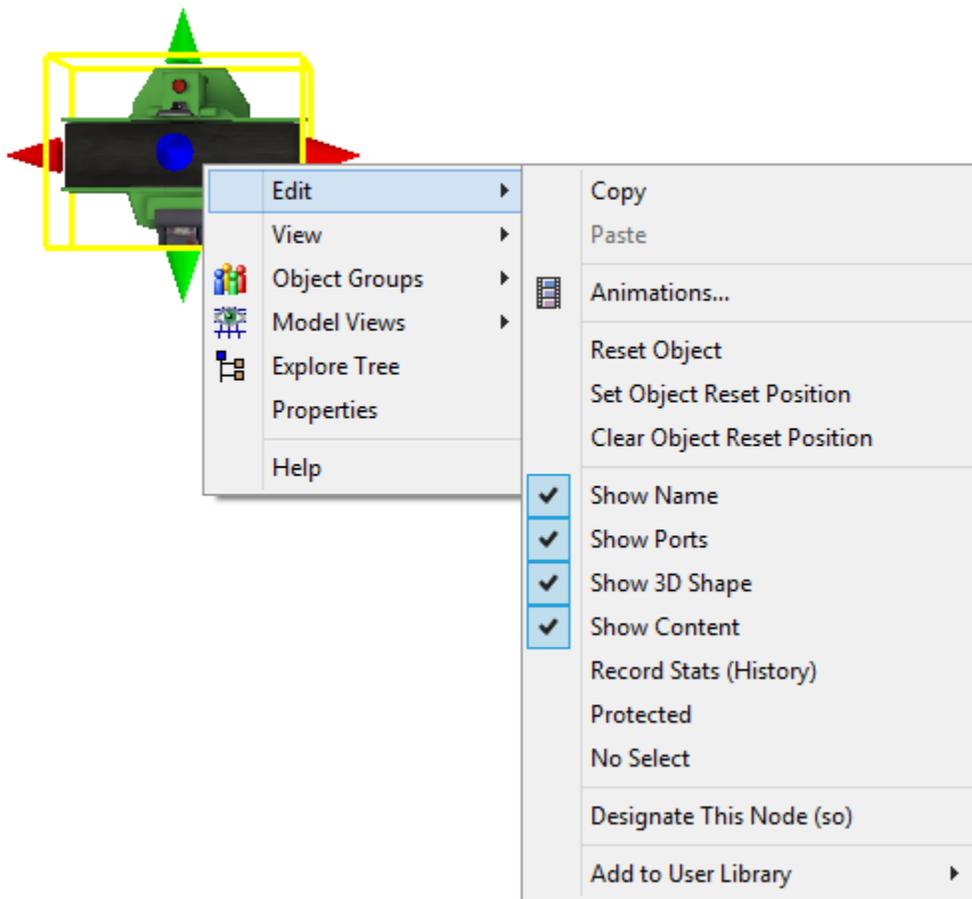


Explore Tree - Abre a janela Tree e mostra os objetos na árvore (Tree).

Properties - Abre a janela de propriedades dos objetos.

Help - Mostra a página de ajuda.

Editar



Copy - Isso copia o item atual para a área de transferência.

Paste - Este cola da área de transferência para o item atual (normalmente o modelo).

Animations... - Abre o Animation Creator para editar a animação dos objetos.

Reset Object - Isso redefine a rotação x / y / z e a localização z do objeto a 0.

Set Object Reset Position - Isso economiza posição atual do objeto, de modo que sempre que você pressione o botão reset, o objeto vai voltar para essa posição (especialmente útil para objetos móveis, como transportadores e operadores).

Clear Object Reset Position - Limpa a posição de resetar o objeto.

Show Name - Quando esta opção for selecionada, o nome de exibição do objeto é sempre visível.

Show Ports – Torna visível as portas e conexões do item.

Show 3D Shape – Torna visível o objeto em 3D.

Show Content – Torna visível os flowitens dentro de um objeto (i.e., flowitens no Queue).

Record Stats (History) – Esta é uma outra maneira de ativar os registros estatísticos que estão sendo gravados de um objeto.

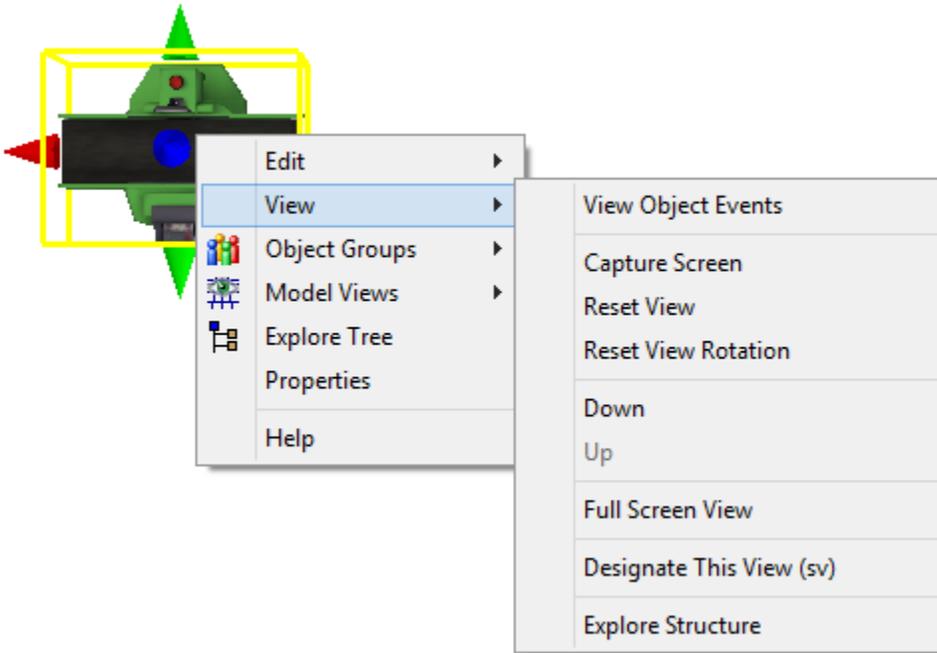
Protected - Isto bloqueia (trava) a localização, tamanho e rotação dos objetos.

No Select - Isto alterna a não seleção dos objetos, fazendo com que os objetos não fiquem acessíveis na superfície de visualização 3D (mas acessível na janela Tree).

Designate This Node (so) - Isto atribui o objeto como "objeto selecionado", que pode ser referenciado no código como (). Normalmente você vai usar essa opção para escrever o código na janela de script. Só pode haver um modo () a qualquer momento.

Add to User Library – Esta opção adiciona o objeto selecionado para a biblioteca de usuário atualmente ativa (currently active user library).

View



View Object Events - Abre a lista de eventos e mostra todos os eventos atuais para aquele objeto.

Capture Screen - Esta opção captura a tela atual, salvando-o como um arquivo de bitmap na pasta de cópias de Flexsim. Isso também pode ser feito selecionando a tela ativa e pressionando a tecla 'P'.

Reset View - Isto devolve a posição da câmara da vista à sua posição inicial, que é focado em 0,0,0.

Reset View Rotation (not on Planar) - Isso retorna a camera do ponto de vista de rotação para sua posição inicial, como se estivesse olhando de baixo.

Down - Esta opção faz com que o ponto de vista volte-se para fora de um objeto selecionado. Se você deslocou a visão para vários níveis Guiaixo ou dentro do objeto, você terá que voltar até o mesmo número de níveis para alcançar o ponto de vista principal.

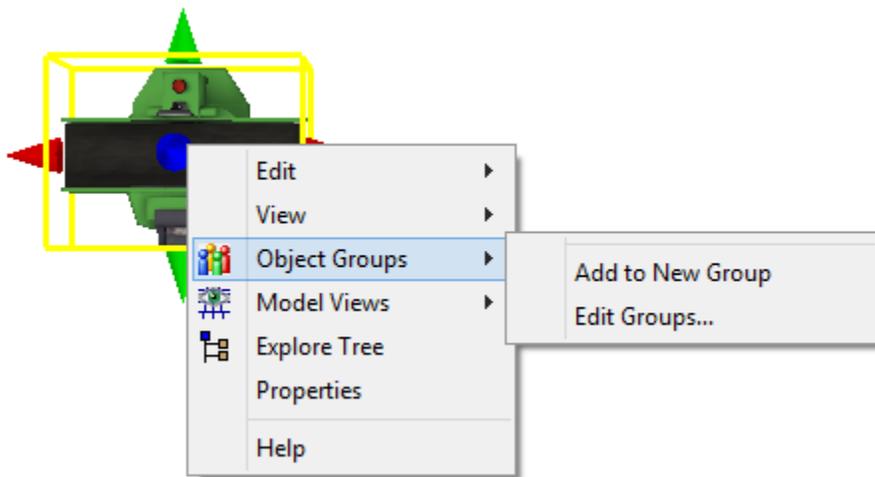
Up – Esta opção faz com que se visualize dentro do objeto selecionado, permitindo que visualize o que está acontecendo com os conteúdos do objeto. A barra de títulos da janela irá mostrar o nome do objeto que você está observando. Este recurso é particularmente útil para os Visual Tools que mantêm outros objetos.

Full Screen View (Perspective only) – Serve para redimensionar a janela de visualização para ocupar a tela inteira. A barra de títulos da janela não será exibida, e qualquer outra tela do FlexSim no monitor, incluindo a tela principal, serão escondidas através deste formato de visualização. Para sair da visualização de tela cheia, clique com o botão direito do mouse e selecione View > Out of Full Screen View.

Designate This View (sv) – Serve para atribuir a janela como “tipo de visualização selecionado”, o qual pode ser referenciado através de código como sv(). Você usualmente usará esta opção para escrever em linguagem de programação na tela de script. Lá pode ser colocado apenas um sv() em qualquer tempo.

Explore Structure – Esta opção traz uma janela tree que explora a estrutura da tree referente a janela de visualização ortográfica.

Object Groups

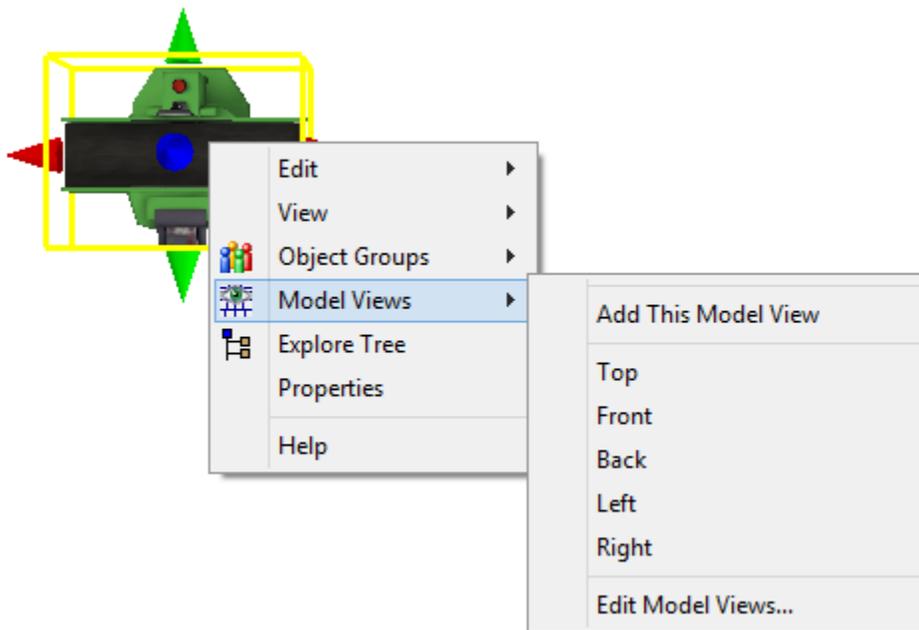


Esta listagem será dinamicamente atualizada conforme você adiciona grupos. Cada nome de grupo será listado, e tornar-se-á selecionável, para que você possa facilmente adicionar objetos a qualquer grupo.

Add to New Group - Isto adiciona objetos para um novo grupo.

Edit Groups... - Isto abre o recurso de grupos de objetos.

Model Views

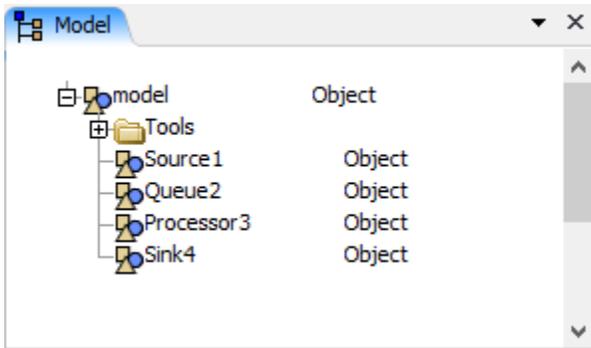


Esta listagem se atualiza dinamicamente para mostrar todas as visualizações disponíveis. As padrões são **Top**, **Front**, **Back**, **Left**, e **Right**. Selecionando uma visualização disponível irá rotacionar e levar o modelo para aquela visualização.

Add This Model View - Abre o utilitário de visualização e adiciona a visualização atual para a listagem de visualização.

Edit Model Views... – Abre o utilitário de visualização

Tree Window



Para informações sobre a estrutura em árvore (tree), veja a página que fala da estrutura em Tree.

A janela Tree pode ser acessada da barra de menu ao clicar em  ou através do menu View.

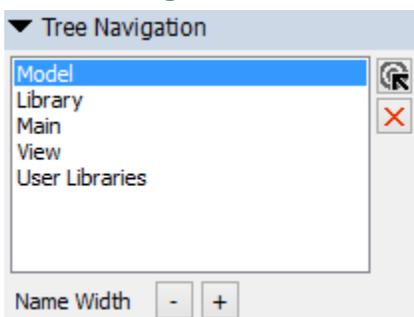
Navegando na Tree

Para movimentar-se na janela tree, clique com o mouse sobre a superfície da janela do tree e arraste para onde quiser. Você também pode usar o botão do mouse ou o teclado do mouse para movimentar-se para cima ou para baixo na janela do tree. O painel Tree Navigation da janela Quick Properties, como descrito Guiaixo, pode ser usado para navegar na Tree (veja logo Guiaixo).

Quick Properties

Quando a janela Tree estiver ativa, a janela Quick Properties será alterada e mostrará o painel Tree Navigation conforme imagem Guiaixo e também o painel de pesquisa (Search panel). Se você clicar sobre um node na tree que representa um objeto sem dados, o painel de propriedades do node será mostrado na janela Quick Properties. Clicando sobre os nodes de um objeto com dados, a janela Quick Properties irá mostrar os mesmos resultados quando visualizamos as propriedades de um objeto na superfície de simulação em 3D.

Tree Navigation



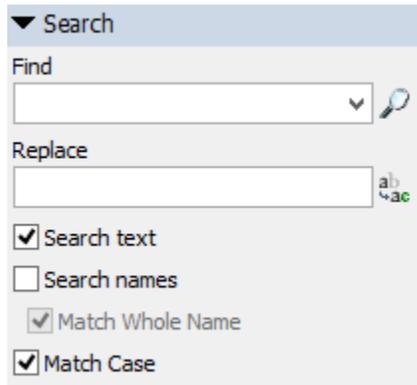
O painel permite você a navegar entre as seções da árvore do FlexSim (Flexsim tree).

- **Model** – Um subconjunto da árvore principal (Main Tree), isto contém todos os objetos usado no modelo atual.
- **Library** - Um subconjunto da árvore principal, isto contém todas as bibliotecas de objetos do FlexSim (encontrados na Library Icon Grid).
- **Main** - A árvore principal contém muitas das funções de nível elevado no FlexSim.
- **View** - A árvore de visualização contém todos os GUIs na interface do FlexSim.
- **User Libraries** – Um subconjunto do the Main Tree, todas as bibliotecas customizáveis estarão dentro deste node.

Você também tem a opção de salvar os tipos de visualizações a partir da árvore. Ao destacar um nó na árvore e pressionar em , o nó será adicionado à lista de pontos de vista predefinidos. Isto permite-lhe saltar rapidamente para trás e para a frente entre os diferentes setores da Árvore. Selecione um ponto de vista e clique em X para removê-lo da lista salva na pasta Tools/TreeNavigation. Visto que são adicionados a partir de qualquer outro lugar no Main Tree ou no View Tree são salvos nos User Prefs e estarão disponível sempre que o Flexsim for aberto sob o seu usuário.

Name Width - Isto controla a largura dos nomes dos nodes na árvore.

Search



O Search permite você pesquisar a partir de textos e nomes dos nodes na árvore (Tree). O Search iniciará pelo node em destaque e ativo na Tree Window. Se o node não estiver destacado, a pesquisa iniciará na parte superior da janela ativa do Tree. O search busca em todos os subnodes e nodes atributos dos objetos.

O campo Replace permite a você substituir todas as ocorrências do texto encontrado. Isto pode ser aplicado para o texto e/ou nome dos nodes.

 - Encontra o texto específico (ou pressione Enter no campo).

 - Encontra o texto acima e o substitui com o texto específico (ou pressione Enter no campo)

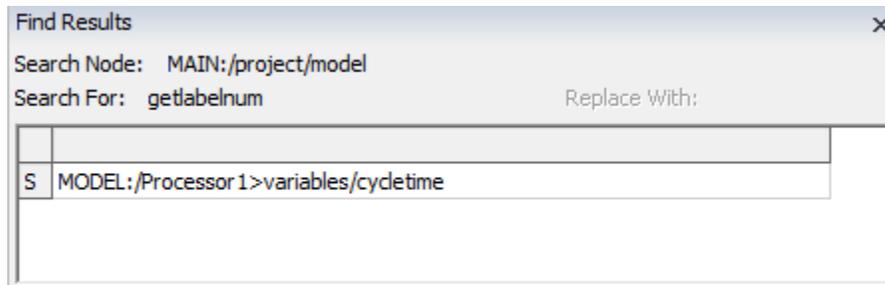
Search text - Pesquisas através de todo o texto sob o node selecionado.

Search names - Pesquisas através de todos os nomes de nodes em nodes selecionado.

Match Whole Name - Se não for controlada, a busca vai retornar qualquer nome do node que contém o texto procurado.

Match Case - Se não for controlada, a busca vai encontrar todo o texto / nomes que contenham o texto da pesquisa, independentemente de estar maiúscula ou não. Por exemplo, se você procurar por "myvariablename", os resultados ainda vão retornar todos os nodes que contêm "MyVariableName".

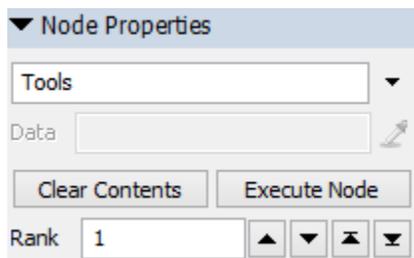
Uma vez que a pesquisa começar, os resultados encontrado irão aparecer:



Dê um duplo clique sobre o resultado para visualizar o texto na janela de edição de código. Você pode clicar com o botão direito do mouse sobre o resultado para editar o código ou explorar a Tree.

Nodes alternado como FlexScript terá um S do lado esquerdo do node. Nodes alternado como C + + irá exibir um C e nodes alternado como DLL vinculada exibirá um D.

Node Properties



Quando um node de um não objeto é destacado na árvore, o Node Properties panel irá aparecer, mostrando o nome e a posição do node. Se o node tiver alguns dados, o campo de dados(Data) irá mostrar o node associado. Use o sampler  para selecionar o node para criar uma associação com ele.



▼ - Isto irá mostrar um menu de nomes de atributos do objeto possíveis. Isto é uma listagem curta e organizada retirada da janela Attribute Hints.

Clear Contents - Elimina todos os subnodes do node destacado.

Execute Node - Chama o nodefunction ou executefsnode do node, executando seus comandos FlexScript, C++ ou código DLL.

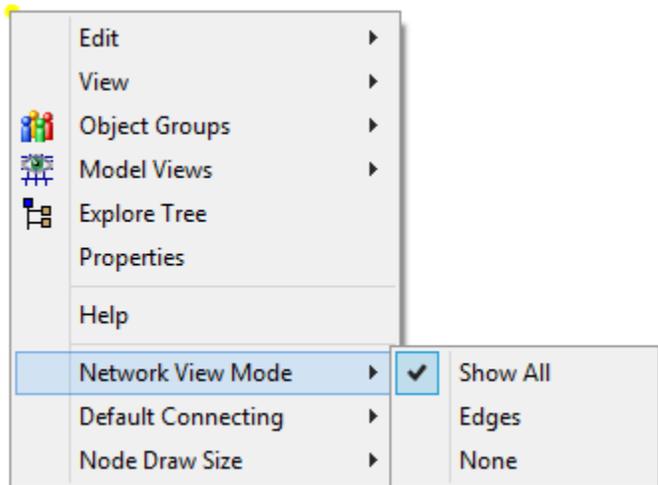
Rank - Isto especifica a posição (rank) do node em destaque em sua árvore de origem. Entre com um novo número (rank number) ou use a seta para cima e para baixo para reorganizar a posição do node.

Utilitário Travel Networks

O utilitário Travel Networks foi removido dessa nova versão e agora pode ser encontrado ao arrastar o Network Node para dentro do modelo e clicando com o botão direito sobre ele.

Modo de visualização para o Network

Para alterar a forma com que os NetworkNode são visualizados na visualização em 3D, clique sobre o botão direito sobre um NetworkNode e escolha Network View Mode.



Alternativamente, você pode manter a tecla X pressionada e clicar com o botão esquerdo sobre o networknode e alternar entre estes três modos.

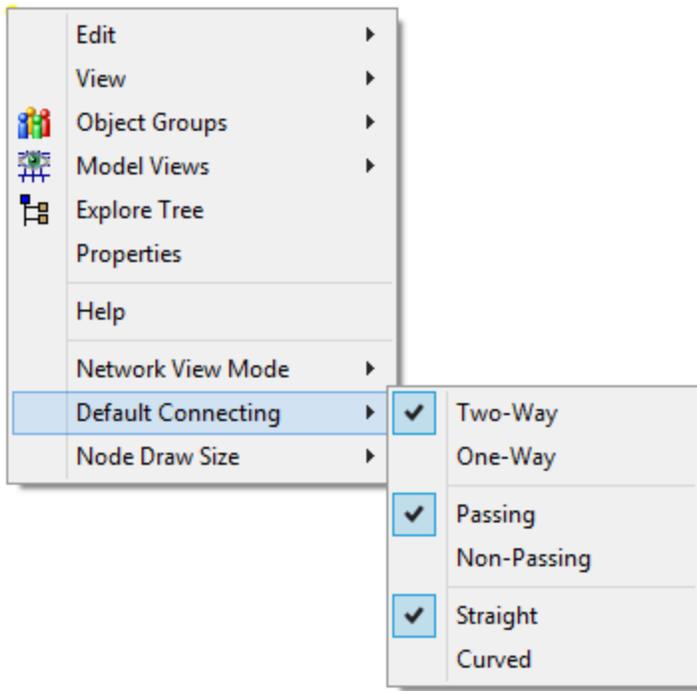
Show All - Mostra todos os NetworkNodes, os limites/ligações entre eles, e as direções das setas de fluxo.

Edges – Exibe os limites entre os NetworkNodes apenas e esconde as setas das direções de movimentação. .

None - Esconde todas as conexões entre os network nodes, as setas de direções de movimentação e mantém apenas o network nodes que você clicou com o botão direito sobre.

Conexão Padrão

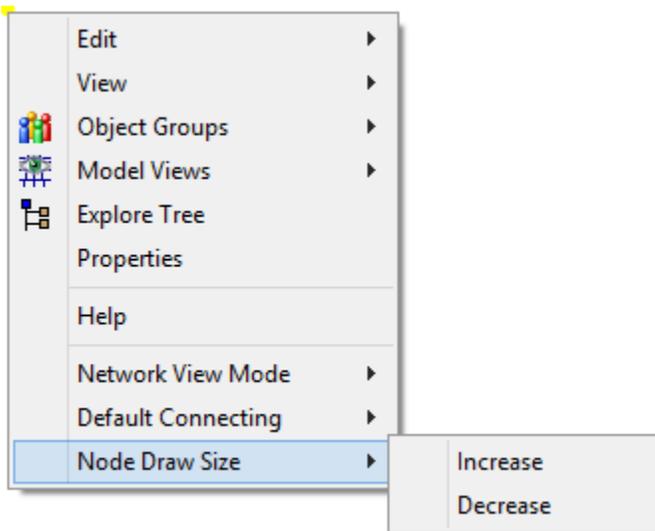
A conexão padrão especifica como as novas conexões serão feitas entre os NetworkNodes quando você fizer uma conexão arrastando com o A pressionado. Estes parâmetros podem ser alterados em cada conexão individual após ela ter sido criada.



Você pode selecionar as conexões padrões de duplo sentido ou sentido único, ultrapassagem (passing) e proibição de sentido (nonpassing), em linha reta (straight) ou em curva (curved).

Tamanho do Node do NN

Você pode também alterar o tamanho do network nodes ao clicar com o botão direito do mouse sobre o NN e escolhendo a opção **Node Draw Size**. Os Network Node são desenhados com tamanho de pixel específico. Eles não são afetados pelo zoom da câmera. Este é o motivo que eles se parecem maior quando você tira o zoom e menores quando você aumenta o zoom em relação aos outros objetos em seu modelo.



Alternativamente, você pode clicar com o botão esquerdo do mouse sobre o NetworkNode e pressionar Ctrl+L ou Ctrl+K para aumentar ou diminuir o tamanho do NetworkNode respectivamente.

Library Icon Grid (Grade da biblioteca de objetos)



A library icon grid (grade da biblioteca de objetos) permite você arrastar objetos para dentro do seu modelo a partir de uma das bibliotecas padrões do FlexSim, ou de uma biblioteca customizada pelo próprio usuário. Se a grade não estiver visível, você pode acessá-la através do menu **View > Drag-Drop Library**.

Os objetos estão divididos dentro de grupos que podem ser expandidos ou escondidos ao pressionar no botão + ou – próximo ao grupo de objetos. Os grupos padrões do FlexSim são Fixed Resources, Task

Executers, Travel Networks, Visual e Fluid. Se você criar sua biblioteca customizada, estas bibliotecas serão exibidas na parte superior e no topo dos grupos.

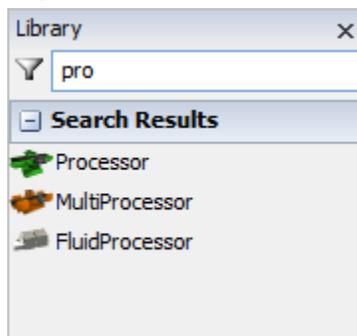
Criando Objetos

Existem dois métodos para criar objetos de uma biblioteca:

- Clique sobre um objeto e mantenha o mouse pressionado e arraste os para a superfície de simulação 3D e solte o botão do mouse na posição que você gostaria de deixar o objeto dentro do modelo.
- Clique e solte um objeto na biblioteca para entrar no modo de criação (Create Mode). Clique em algum lugar na superfície 3D para criar novos objetos. Para sair do creat mode, clique com o botão direito do mouse ou pressione a tecla Escape.

Filtrando

A grade de biblioteca de objetos pode ser filtrado por nome ou entrando com um texto no campo 

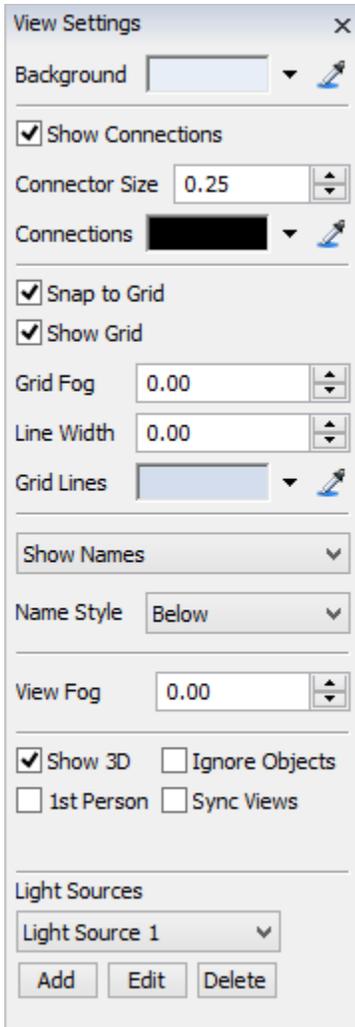


O Sistema de filtragem não diferencia entre maiúscula ou minúscula.

Context Sensitive

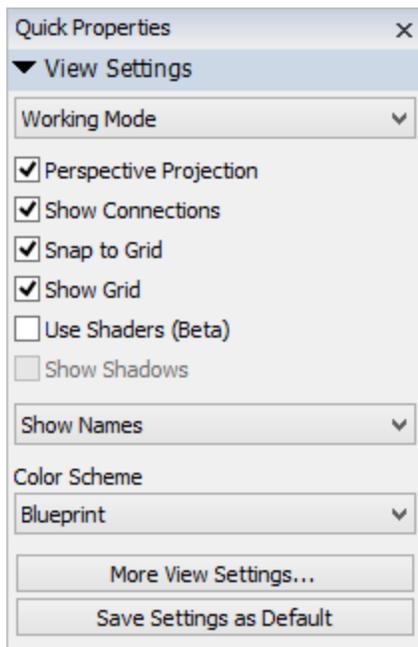
A grade de biblioteca de objetos é context sensitive. Ela será alterada baseada no que o modelador estiver trGuialhando. Irá mostrar a forma 3D quando estiver no FlowItem Bin e Animation Creator, nos blocos code builder enquanto na janela de edição de códigos e janelas do dashboard enquanto estiver editando os Dashboards.

Configurações de Visualização



Esta caixa de diálogo pode ser aberta ao selecionar **View > View Settings** do menu principal ou clicando em **More View Settings...** da janela Quick Properties. Isto permitirá a você configurar o modo de visualização do seu modelo. Estas configurações aplicam-se apenas para a janela que estiver ativada no instante da modelagem e será perdida se esta janela for fechada. Se uma janela diferentemente da visão ortho, perspective, ou planar estiver ativa, esta janela ficará acinzentada.

A janela Quick Properties irá mostrar algumas das mais comuns configurações de visualização (View Settings) sempre que a visualização em 3D estiver ativa e nenhum objeto está selecionado.



Mode - Esta opção dá a você duas formas de visualização pré-configurada. Alternando para o modo de apresentação, esconde as conexões, os grids, e os nomes dos objetos.

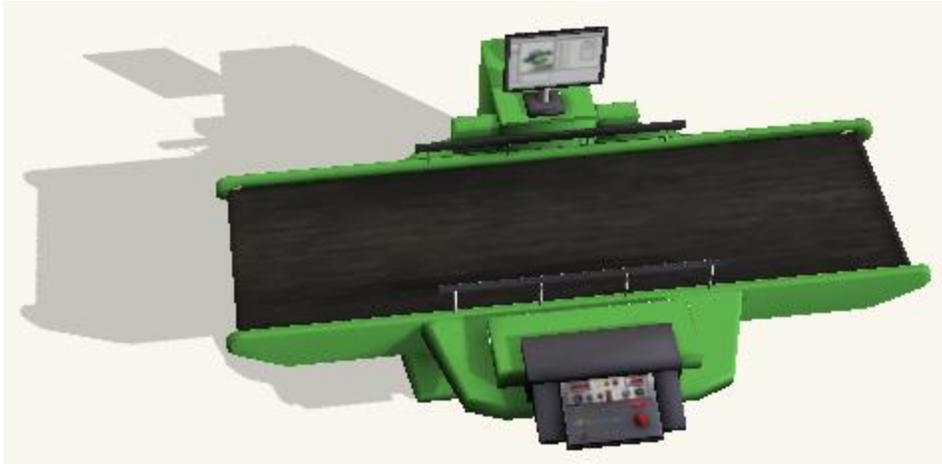
Perspective Projection - Esta opção alterna a visualização em 3D como sendo em vista perspectiva ou sendo em vista ortográfica. A vista em perspectiva dá uma visão mais real de 3D enquanto a ortográfica não dá opção para vista em profundidade.

Snap to Grid - Se esta caixa estiver selecionada, os objetos irão automaticamente movimentar-se para a linha de grade mais próxima quando eles estiverem se movimentando dentro de um modelo. Isto é útil para colocar os objetos em localização precisa. Redimensionando os tamanhos dos objetos irá também encaixá-los no grid que estiver selecionado

Show Grid - Se esta caixa estiver selecionada, o grid será exibido na janela de visualização do modelo.

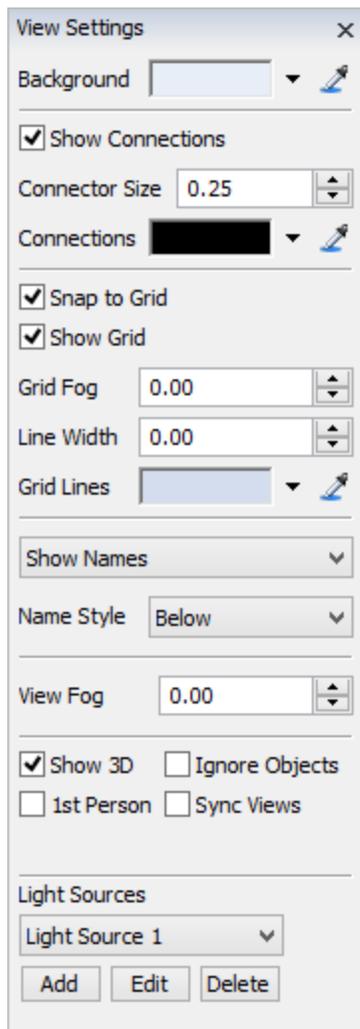
Use Shaders (Beta) - Usa sombra para desenhar objetos. Este recurso ainda encontra-se em fase de testes (Beta) e não é garantido que funcionará exatamente como atualmente no futuro.

Show Shadows - Quando shaders estão ativadas, você também tem a opção de ter objetos projetando sombras em outros objetos. Para fazer com que objetos projetem sombras sobre um "piso", crie uma Visual Tool plana e coloque-o Guiaixo de seus objetos.



Color Scheme - Isto serve para modificar a visualização das cores em 3D.

Save Settings as Default - Como descrito acima, as configurações de visualização em 3D não são guardadas e serão perdidos quando a visualização em 3D estiver fechada. Isto vai salvar suas configurações de visualização atuais como padrão para todas as novas visualizações em 3D que serão abertas.



Cor do Background

Background Color – Esta opção permite você selecionar a cor do background padrão da janela de visualização que estiver selecionada dentro da perspectiva de caixa de diálogo selecionada no formato do Windows.

Conexões

Show Connections - Se esta caixa estiver selecionada, as portas e as linhas de conexão das portas serão exibidas na janela de visualização. Escondendo esta conexão frequentemente torna-se fácil visualizar o que está acontecendo no modelo. Se o modelo estiver rodando devagar, desmarque esta caixa para que ele volte a rodar mais rápido.

Connector Size - Este número configura o quanto mais largo está a conexão da porta sobre um objeto.

Connections Color - Este valor configure a cor das linhas de conexão no modo de visualização.

Grade (Grid)

Snap to Grid – Se esta caixa estiver selecionada, os objetos irão automaticamente movimentar-se para a linha de grade mais próxima quando eles estiverem se movimentando dentro de um modelo. Isto é útil para colocar os objetos em localização precisa. Redimensionando os tamanhos dos objetos irá também encaixá-los no grid que estiver selecionado.

Show Grid – Se esta caixa estiver selecionada, o grid será exibido na janela de visualização do modelo.

Grid Fog - Este valor permite que você tenha a exibição do grid no plano de fundo da tela de modelagem, uma vez que pode aparecer mais longe do ponto de vista. Geralmente isso só é útil em uma vista em

perspectiva. Defina o valor entre 0 e 1, 0 significando que o grid pode ficar aparente, ou até mesmo desaparecer.

Grid Line Width – Este item configura a largura das linhas do grid.

Grid Line Color – Este item configura a cor das linhas do grid.

Nomes

Show Names – Esta listagem de opções permite você escolher se nomes e informações estatísticas dos objetos serão exibidos ou escondidos.

Name Style – Esta listagem de opções permite você escolher se o nome do objeto será mostrado. Escolha para colocar Guiaixo do objeto ou no centro do objeto.

Outras configurações

View Fog – Este valor configura o ambiente para criar um embaçamento dentro do modelo. O View fog faz com que os objetos que estão distantes da posição da câmara ficar quase que obscura dentro do plano de fundo do ambiente de modelagem. Configure o valor entre 0 e 1, significando sem embaçamento algum com 0 e com embaçamento total com 1. Isto é somente aplicável para a vista em perspectiva.

Show 3D Shapes – Se esta caixa for selecionada, as formas 3D (arquivos .3ds ou .wrl) para todos os objetos no modelo serão mostradas na janela de visualização. Alguns objetos não tem arquivos 3D associado com eles, pois são geralmente criados diretamente com OpenGL. Estes objetos não serão afetados por esta seleção de caixa.

Show 2D Bases – Se esta caixa for selecionado, a base 2D dos objetos serão mostrados na janela de visualização. Ao esconder as bases, geralmente faz com que a visualização em 3D tenha uma forma de visualização melhor.

1st Person – Se esta caixa for selecionada, o controle do mouse na janela de visualização estará no modo de primeira pessoa. Isto significa que a tela de visualização irá rotacionar ao redor do ponto de visualização do usuário e não ao redor do ponto no meio da tela. Este modo é útil quando estiver navegando no modo fly-through.

Sync Views – Se esta caixa estiver selecionada, todas as janelas abertas de visualização serão atualizadas automaticamente e ao mesmo tempo. Se não estiver selecionada, algumas janelas não irão atualizar até que a tarefa estiver completada na outra janela diferente. Ao selecionar esta caixa, o software poderá rodar a simulação um pouco mais devagar.

Ignore Objects – Se esta caixa estiver selecionada, o usuário não estará apto a clicar em nenhum objeto na janela de visualização. Isto é útil para navegação ao redor do modelo que estiver finalizado, impedindo que o usuário acidentalmente movimente qualquer objeto.

Fontes de Luz

Este controles permitem ao usuário adicionar, editar e criar fontes de luz para a janela de visualização.

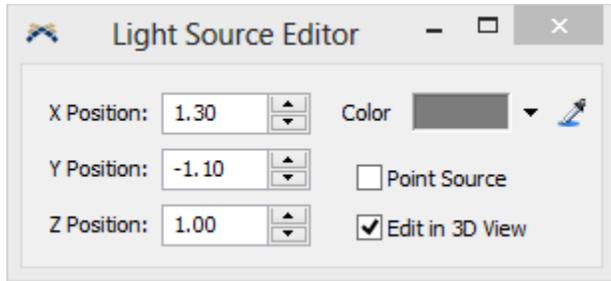
Light Sources – A lista de opções contém todas as fontes de luz que atualmente estão na janela de visualização.

Edit – Este botão abre o editor da fonte de luz, mostrando a cor atual em uma listagem de cores.

Add – Este botão irá criar uma nova fonte de luz na janela de visualização.

Delete – Este botão irá deletar da janela de visualização a fonte de luz atualmente exibida na listagem de opções. Deve existir, pelo menos uma fonte de luz no modelo

Edição de Fonte de Luz



Esta tela pode ser acessado ao ir em **View Settings** e clicando no botão **Edit** na seção **Light Sources**

X Position - Este número é a posição ao longo do eixo X do modelo onde a fonte de luz está localizada. Este número é relativo a posição de cada objeto.

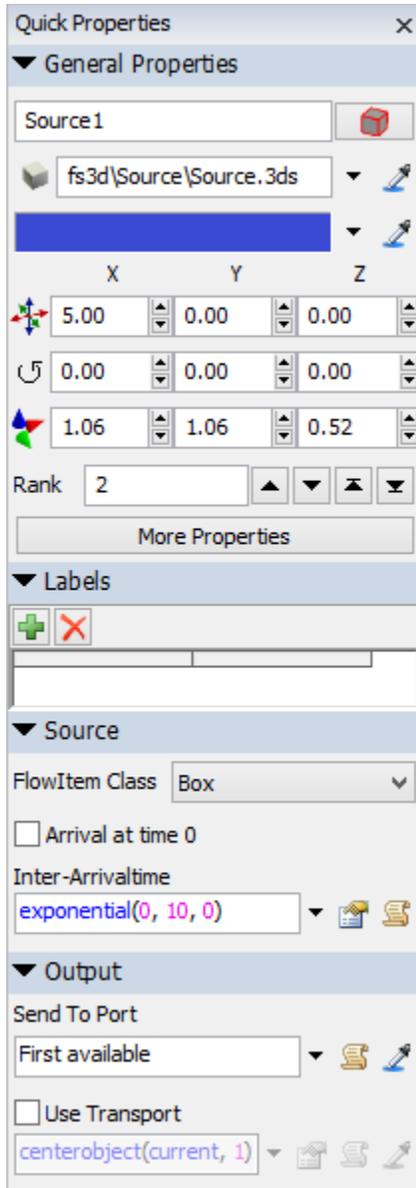
Y Position - Este número é a posição ao longo do eixo Y do modelo onde a fonte de luz está localizada. Este número é relativo a posição de cada objeto.

Z Position - Este número é a posição ao longo do eixo Z do modelo onde a fonte de luz está localizada. Este número é relativo a posição de cada objeto.

Color - Esta opção permite ao usuário escolher a cor da luz que sai da fonte para uma caixa de escolha padrão de cores do Windows.

As point source - Se esta opção estiver selecionada, a luz gerada pela fonte aparecerá vir de um ponto específico, relativo a camera. Se não estiver selecionado, a luz parecerá vir de um determinada direção apenas.

Janela Quick Properties

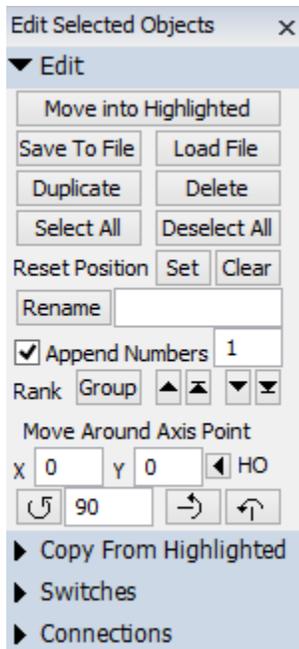


A janela Quick Properties é uma janela context sensitive. Dependendo da janela que estiver ativa no momento e do objeto selecionado, as informações da janela Quick Properties serão alteradas. Nós iremos cobrir este tópico agora. Para mais informações sobre estes itens, leia sobre os seguinte itens:

- Animation Creator
- Dashboard
- FlowItem Bin
- Orthographic/Perspective View
- Tree Window

Se a janela Quick Properties estiver fechada, você pode abri-lá através do menu View.

Edit Selected Objects



O utilitário para edição de objetos selecionados (The Edit Selected Objects Utility) pode ser encontrado através do menu principal **View > Edit Selected Objects**. Ele oferece várias opções que são executadas no conjunto selecionado de objetos na janela de exibição. Para selecionar um grupo de objetos, segure a tecla Shift ou Control e arraste uma caixa de seleção em torno do grupo de objetos. Os objetos do conjunto de seleção terá uma caixa vermelha desenhada em torno deles. O objeto que encontra-se destacado (o último objeto que você clicou) terá uma caixa amarela desenhada em torno dele.

Edit

Move into highlighted object – Esta opção movimenta o objeto em destaque (os que encontram-se com caixa destaque em vermelho) para dentro de outro objeto em destaque (os que encontram-se com caixa destaque em amarelo). Isto permite ao objeto selecionado ser usado como um container.

Save to file – Os objetos selecionado serão salvos como uma extensão .t que pode ser reaberto posteriormente no FlexSim. Isto permite aos usuários a salvar e importar partes do modelo se necessário.

Load from file – Este objeto abre arquivos de extensão .t dentro do objeto selecionado no momento. O objeto selecionado então torna-se um container para o os objetos importados.

Duplicate – Esta opção cria uma cópia idêntica no modelo do conjunto selecionado. Todas as conexões de portas são mantidas intactas.

Delete – Esta opção deleta os objetos selecionados. Isto pode ser feita ao destacar um dos objetos selecionados na tela de visualização e pressionando a tecla Delete ou Backspace.

Select All – Esta opção adiciona todos os objetos do modelo para dentro do conjunto selecionado.

Deselect All – Esta opção leva todos os objetos do modelo para fora do conjunto selecionado.

Set Reset Position – Esta opção reseta a posição de cada objeto do conjunto selecionado para sua posição de origem ou posição de reset. Quando o modelo é resetado, todos os objetos com uma posição inicial gravada será movimentado, rotacionado e redimensionado para esta posição.

Clear Reset Position – Esta opção remove a posição reset de cada um dos objetos do conjunto selecionado.

Rename – Entre um nome e pressione o botão “Rename” para renomear todos os objetos selecionados.

Append Numbers – Se está opção for selecionada, o botão rename também irá acrescentar números para o novo nome a partir de um número específico, incrementando cada objeto renomeado por 1.

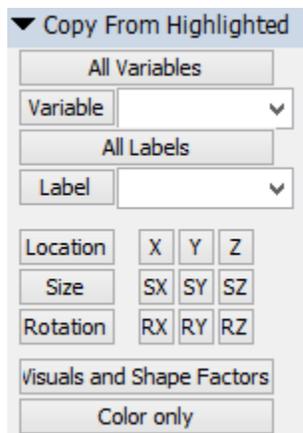
Rank – Esta série de botões deixa você agrupar os objetos juntos em uma árvore, assim como movimentá-las para cima ou para baixo.

Move Around Axis Point - Este conjunto de controles permite que você manipule os locais e as rotações do conjunto de objetos selecionados. Todas as operações são feitas em torno de um eixo de localização no modelo. Você pode entrar com os valores diretamente ou destacar um objeto no modelo e clicar no botão "< HO" para dar aquela localização ao objeto. Entre com um ângulo de rotação em graus e pressione o botão  para rotacionar o conjunto selecionado ao redor do ponto do eixo. Clique nos botões  ou  para virar os objetos selecionados sobre o ponto do eixo verticalmente ou horizontalmente, respectivamente.

Copy from Highlighted

Estas opções copiam informações dos objetos destacados (os de destaque em amarelo) para todos os objetos do conjunto selecionado (os de destaque em vermelho).

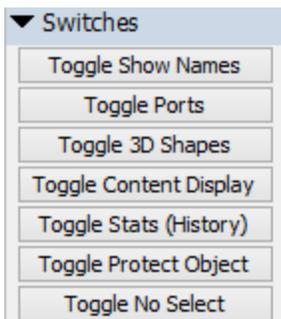
Se qualquer um dos objetos selecionados tiver triggers que são que são alternados como funções C + +, será necessário recompilar o modelo antes de rodá-lo novamente



Os seguintes dados pode ser copiado:

- Todas as variáveis (incluindo todas as triggers);
- Todas labels;
- Variável simples ou label. Digite o nome da variável ou label a ser copiada e pressione o botão "Variable" ou "Label". Ao invés de digitar o nome explicitamente, você pode selecionar uma opção da lista de opções drop down que mostra a você todas as variáveis ou labels do objeto em destaque. Se você não tiver uma janela de visualização ativa ou um objeto em destaque dentro da tela de visualização atual, a listagem drop down estará vazia. A listagem drop down de label também estará vazia se o objeto em destaque não tiver labels associado a ele.
- Spatial values. Estes botões permitem a você copiar os valores espaciais do objeto em destaque. Você pode copiar completamente o tamanho/localização/rotação em x/y/z com uma operação , ou copiar individualmente os atributos de x, y ou z.
- Visuals and Shape factors. Este botão copia as informações visuais como os formatos em 3D, 2D e cores. Isto também copia os fatores de forma do objeto em destaque.

Switches – Esta opção permite ativar e desativar várias configurações do objeto selecionado. Para todos os objetos do conjunto selecionado. Para todos os objetos no conjunto selecionado, a configuração escolhida será revertida (alternada). Ou seja, se é sobre ele será desligado e, se estiver desligado ele será ligado.

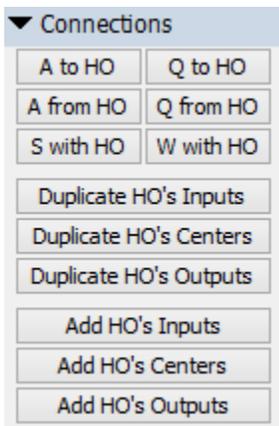


As configurações disponíveis para alternar ou ativar são:

- **Show Names** - Mostrar nomes e estatísticas dos objetos
- **Ports** - Mostrar as portas dos objetos e as conexões de portas.
- **3D Shapes** - Mostrar as formas 3D dos objetos. Note que o Custom Draw Code trigger não será ativado para objetos com formas em 3D escondidos dessa forma.
- **Content Display** - Apresentar o conteúdo dos objetos. Se estiver desligada, os objetos dentro do objeto principal não será visto
- **Stats (History)** - Ativar as estatísticas de coleta para os objetos
- **Protect Object** - Protege os objetos. Isto bloqueia (trava) a localização dos objetos, tamanho e rotação.
- **No Select** - Faz com que o objeto fique não clicável da visualização em 3D (objeto será acessado através da janela Tree)

Connections

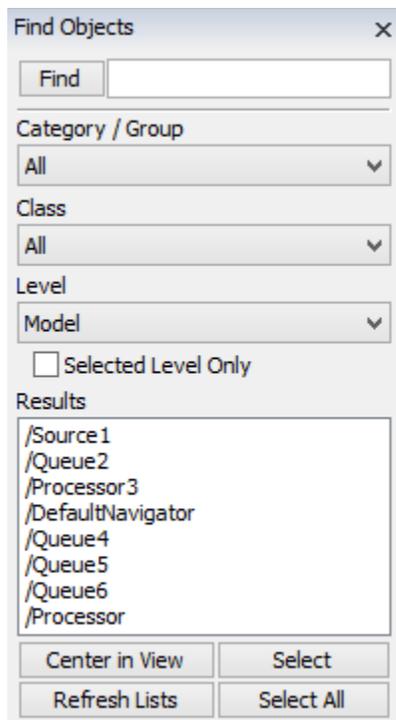
Este sub-menu permite a você faça conexões de arraste entre o objeto de destaque e do conjunto de seleção.



- **A to HO** – esta opção faz com que aconteça uma conexão de arraste semelhante ao manter a tecla A apertada, partindo de todos objetos do conjunto selecionado para o objeto em destaque.
- **Q to HO** – esta opção faz com que seja desfeita uma conexão de arraste semelhante ao manter a tecla Q apertada, partindo dos objetos do conjunto selecionado para o objeto em destaque.
- **A from HO** – esta opção faz com que aconteça uma conexão de arraste semelhante ao manter a tecla A apertada, partindo do objeto em destaque para todos os objetos do conjunto selecionado.
- **Q from HO** – esta opção faz com que seja desfeita uma conexão de arraste semelhante ao manter a tecla Q apertada, partindo do objeto em destaque para todos os objetos do conjunto selecionado.
- **S with HO** – Esta opção faz com que seja feita uma conexão de porta central semelhante a manter a tecla S apertada, partindo de todos os objetos do conjunto selecionado para o objeto em destaque.

- **W with HO** – Esta opção faz com que seja desfeita uma conexão de portal central, semelhante a manter a tecla 'W' apertada, partindo do objeto em destaque para os objetos do conjunto selecionado.
- **Duplicate HO's Inputs** – esta opção destrói todas as portas de entrada dos objetos selecionados e cria cópias das conexões de portas de entrada dos objetos em destaque para todos os objetos do conjunto selecionado.
- **Duplicate HO's Centers** – esta opção destrói todas as portas centrais dos objetos selecionados e cria cópias das conexões de portas centrais dos objetos em destaque para todos os objetos do conjunto selecionado.
- **Duplicate HO's Outputs** – esta opção destrói todas as portas de saída dos objetos selecionados e cria cópias das conexões de portas de saída dos objetos em destaque para todos os objetos do conjunto selecionado.
- **Add HO's Inputs** – esta opção adiciona cópias das conexões de porta de entrada dos objetos em destaque para todos os objetos dentro do conjunto selecionado.
- **Add HO's Centers** – esta opção adiciona cópia das conexões de portas centrais dos objetos em destaque para todos os objetos dentro do conjunto selecionado.
- **Add HO's Outputs** – esta opção adiciona cópia das conexões de portas de saída dos objetos em destaque para todos os objetos dentro do conjunto selecionado.

Find Objects



Este recurso permite que você encontre rapidamente objetos em seu modelo com base no nome ou tipo de objeto. Ele pode ser acessado a partir da opção do menu principal em **View > Find Objects**.

Encontre pelo nome

Digite todo ou parte do nome no campo de texto na parte superior e pressione Find para encontrar um objeto nas opções de visualização Ortho / Perspectiva / Tree. Este campo é case-sensitive, portanto, "Queue" é diferente de "queue". Ao clicar em Find ele irá buscar todos os objetos em seu modelo para encontrar um objeto em que o nome contém um texto específico. Exemplo: buscando pelo "Processor" ele irá encontrar ambos "Processor3" e "FluidProcessor4".

Encontre pela categoria/Grupo/Classe

Você também pode selecionar a categoria de biblioteca e / ou o tipo de classe, e / ou um layer específico no modelo que você quiser encontrar os objetos que fazem parte. Quando você selecionar uma opção no menu drop-down, a listagem na parte inferior será repovoada (atualizada). Selecione um objeto na listagem e pressione Center in View para ir para este objeto nas opções de visualização 3D/ Tree.

Selected Level Only - Marque esta caixa para mostrar somente objetos no nível selecionado. Por exemplo, se você tem uma Visual Tool no modelo que contém três equipamentos (processors) e você seleciona o nível como modelo, estes três processors não serão mostrados na lista de resultado.

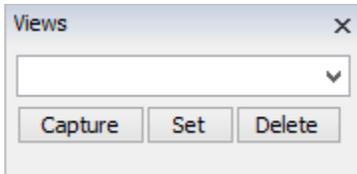
Center in View - Centraliza os objetos selecionados da lista de resultados na visualização em 3D e na Tree.

Refresh Lists - Atualiza todas as listas incluindo a lista de resultados (útil quando você adiciona um objeto para o seu modelo).

Select - Seleciona (caixa vermelha) o objeto selecionado da lista de resultados.

Select All - Seleciona (caixa vermelha) todos os objetos da lista de resultados.

Views



Esta janela pode ser acessada através do menu View em **View menu > Views**.

Este recurso de Visualizações permite que você capture uma posição atual da câmara de vista e adicione a uma lista de pontos de vista.

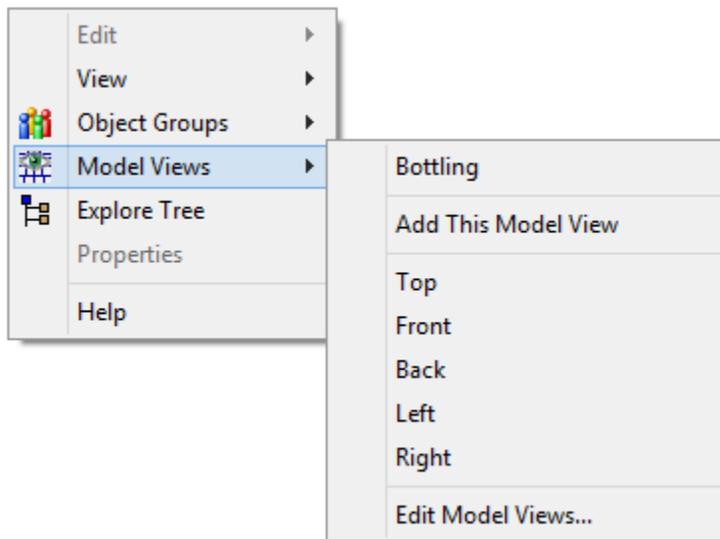
View List- Selecione uma visão a partir da lista drop-down para girar o modelo para essa visão

Capture - Pressione Capture para capturar o ponto de vista atual. Quando uma posição de exibição é capturado, é adicionado a uma listagem na caixa drop-down.

Set - Esta opção irá atualizar a vista atualmente selecionada para a posição/rotação atual da câmera.

Delete - Remove the selected view from the list. Remova a visualização selecionada da lista.

As vistas podem também ser acessada através de um clique com o botão direito do mouse na superfície de simulação 3D.



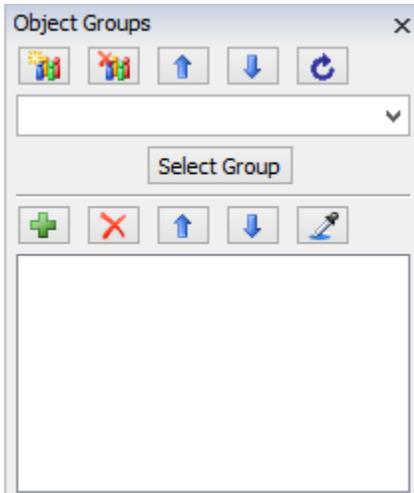
Visualizações customizadas, as que são criadas através da janela Views, irão aparecer no topo do menu.

Add This Model View - Captura a visualização atual e adiciona para a listagem de visualizações do modelo.

Top, Front, Back, Left, Right - Pré-configura as visualizações do modelo.

Edit Model Views... - Abre a janela de visualizações.

Groups



O utilitário groups permite ao usuário criar e editar diferentes grupos em seu modelo.

Informações do grupo é armazenado tanto em /Tools/Groups folder e em cada um dos objetos individuais. Um node é adicionado aos atributos dos objetos chamados Grupos com um acoplamento ao grupo associado. Objetos podem ser incluídas em vários grupos.



- Cria um novo grupo vazio



- Remove o grupo selecionado atualmente.



- reorganize o grupo.



- Atualiza a listagem de grupos

Group Dropdown Menu - Todos os grupos estão listados neste menu drop-down. O nome do grupo selecionado atualmente é editável.

Select Group – Seleciona todo os objetos pertencentes ao grupo (caixa vermelha).



- Abre um menu que permite que você adicione itens atualmente selecionados para o grupo ou defina o grupo para combinar com o atual modelo selecionado.



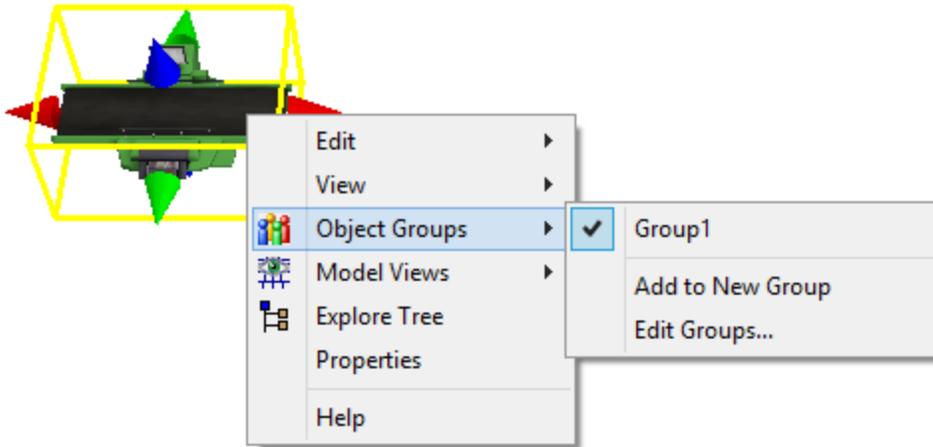
- Remove os objetos selecionados do grupo.



- reorganiza os membros de um grupo.



Clique para entrar no modo “Sample” e então selecione um objeto na superfície de simulação em 3D para adicioná-lo ao grupo. Você pode querer adicionar objetos em certos grupos, clicando com o botão direito do mouse na visualização em 3D.

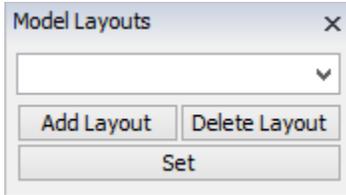


A lista atual de grupos irá aparecer no topo do menu. Cada grupo de nome será acessível, então você poderá facilmente adicionar objetos em qualquer grupo.

Add to New Group - Crie um novo grupo e adicione o objeto ao grupo. Se vários objetos são selecionados (caixa vermelha) isto irá adicionar estes objetos para o grupo também.

Edit Groups... - Abra a janela Groups.

Model Layouts



Esta janela pode ser acessada através do **View menu > Model Layouts**.

Este utilitário permite que você crie e edite certos layouts para o seu modelo. Por exemplo, você pode querer salvar um layout que representa um fluxograma da esquerda para a direita das etapas em seu modelo, onde o fluxo do modelo procede sempre da esquerda para a direita. Então você pode querer salvar outro layout que representa uma disposição real de fábrica. Esta barra de ferramentas permite que você faça isso. Ele também é útil para testar diferentes layouts possíveis de suas instalações.

A informação de layout é armazenada em cada objeto individual. Um node é adicionado ao atributo do objeto chamado Layouts. Para cada layout criado, uma cópia da informação espacial atual do objeto (posição, tamanho e rotação) é armazenada.

Layout List - Mostra a listagem de layouts atuais. Selecione um layout para imediatamente atualizar seu modelo a ao layout.

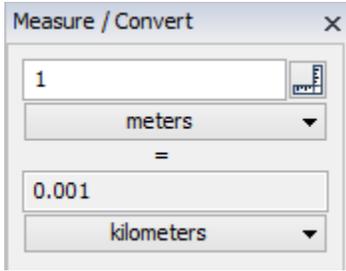
Add Layout - Pressione Add Layout para adicionar um layout. Por padrão, quando você adiciona um layout, ele vai salvar o layout atual do modelo

Delete Layout – Remove o layout selecionado.

Set - Isto irá atualizar o layout do modelo selecionado para atender o layout atual do modelo.

c

Measure / Convert



Esta janela pode ser acessada através do **View menu > Measure Convert**.

A ferramenta Measure/Convert permite você medir distâncias em seu modelo assim como converter unidades de distância/tempo. Por padrão, as unidades de medida serão as unidades do modelo, conforme definido na janela Model Settings.

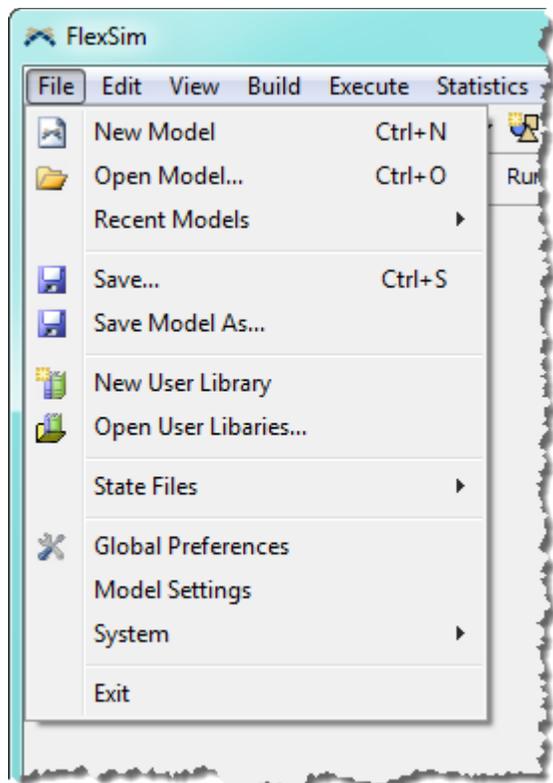
 - Clique neste botão para entrar no modo "Measure". Você pode então clicar em qualquer lugar na superfície de visualização 3D para definir seu ponto de início, e clicar novamente para definir seu ponto final. O resultado da distância será mostrado do lado esquerdo.

Na primeira listagem drop down (mostrado acima como metros), você pode selecionar as unidades para medir "Measure" ou selecionar um comprimento, tempo ou comprimento/tempo para realizar a conversão. Você pode selecionar o valor que quiser no campo localizado na parte superior para realizar a conversão. O campo na parte inferior mostrará o resultado. O botão drop down pode ser alterado para converter em diferentes unidades.

Menu Principal e Barra de Ferramentas

1. File Menu
2. Edit Menu
3. View Menu
4. Build Menu
5. Execute Menu
6. Statistics Menu
7. Tools Menu
8. Debug Menu
9. Help Menu
10. FlexSim Toolbar
11. Simulation Run Panel

File Menu



New Model - Esta opção limpa o modelo atual para que um novo possa ser criado. Não afeta o layout de visualização ou biblioteca. Um aviso será emitido e se você continuar, irá perder todos os objetos em seu modelo. O resultado é um modelo em branco, exceto que todas as bibliotecas customizadas irão permanecer intactas.

Open Model... - Esta opção permite ao usuário escolher o arquivo de modelo do FlexSim (extensão .fsm) para editá-lo. Qualquer mudança que for realizada no modelo atual e que não foram salvas, serão perdidas.

Recent Models - Mostra uma lista dos últimos modelos do FlexSim que foram abertos em seu computador, com os modelos abertos mais recentemente no parte superior da lista.

Save... - Esta opção salva o arquivo do modelo atual. Qualquer mudança que for realizada no modelo atual será salva

Save Model As... - Esta opção permite ao usuário salvar o modelo em um arquivo. O arquivo que é criado tem uma extensão .fsm. O conteúdo inteiro do modelo será salvo. As duas outras opções de salvar um arquivo são em .XML (.fsx – para mais informações veja FlexSim XML) e em extensão .t (para arquivos Tree e efetivamente idêntico ao arquivo .fsm)

New User Library - Esta opção cria uma nova biblioteca de usuário e adiciona-a ao painel de biblioteca. Para mais informações sobre bibliotecas customizadas, veja Creating Custom Libraries.

Open User Libraries... - Esta opção abre uma ou mais bibliotecas salvas, adicionando elas a grade de biblioteca de objetos. Se a biblioteca contém componentes para instalação automática, uma mensagem irá aparecer perguntando se você quer instalar estes componentes. Pressione OK para instalar estes componentes. Para mais informações sobre bibliotecas customizadas, veja Creating Custom Libraries.

State Files - Esta opção permite ao usuário salvar o estado (modelo atual que está rodando) do modelo, ou carregar um modelo salvo para rodar posteriormente. Salvar o estado de simulação do modelo pode ser útil quando o seu modelo estiver no meio de uma simulação e você precisar salvar seu estado atual (todos os flowitem permanecem ondem eles estão e os recursos permanecem no estado de operação atual), e então posteriormente carregar o estado do modelo para continuar a rodar o modelo a partir do ponto salvo.

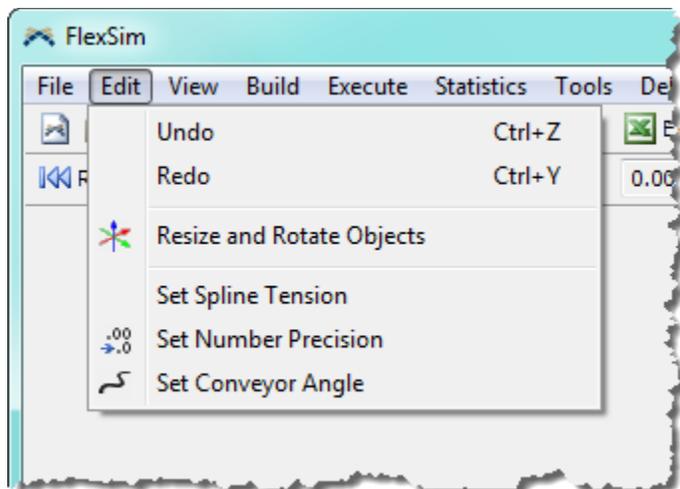
Global Preferences - Esta janela o leva para a janela de preferências, permitindo você configurar as preferências no FlexSim. Você pode também configurar as propriedades dos textos em destaque aqui.

Model Settings - Abre a janela de configuração do modelo, com as configurações atuais do modelo atual do FlexSim. Para mais informações, veja Model Settings.

System - Esta opção permite a você manualmente carregar mídias, assim como desconectar qualquer DLL que foi conectado aos DLL toggled nodes.

Exit - Esta opção irá fechar o FlexSim sem salvar nenhum arquivo. Se o usuário desejar salvar qualquer mudança, o arquivo apropriado deve ser salvo antes desta opção ser selecionada.

Edit Menu



Undo – Apaga as últimas mudanças realizadas no modelo, revertendo ao estado imediatamente anterior à mudança.

Redo – Anula o efeito Undo.

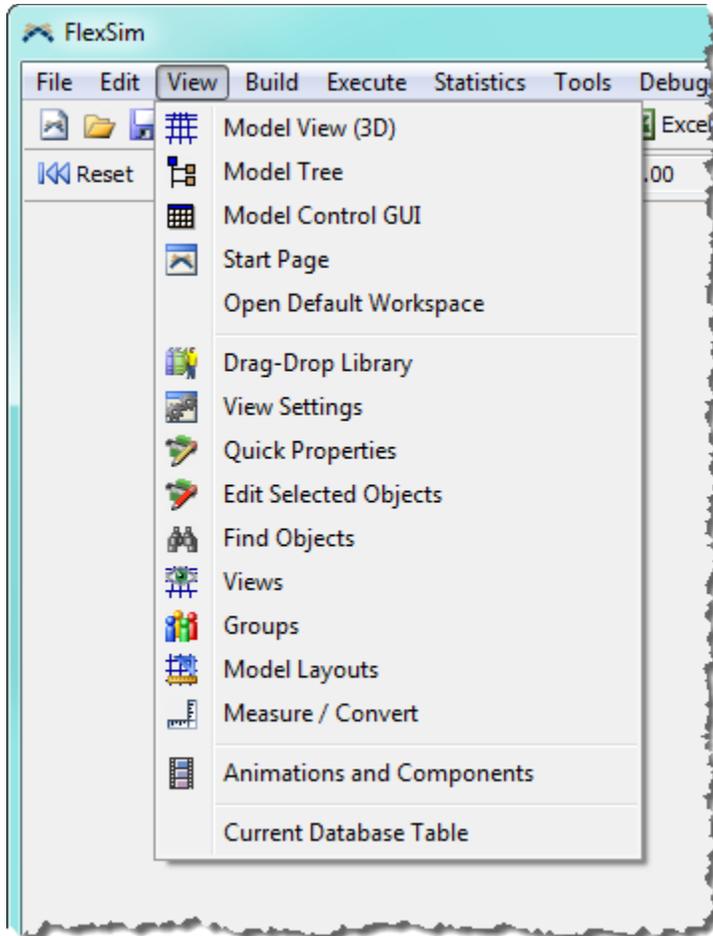
Resize and Rotate Objects - Ao selecionar esta opção, você verá os eixos dos objetos no formato de visualização em perspectiva e ortho para redimensionamento e rotação.

Set Spline Tension - Esta opção traz uma caixa de diálogo para configurar a tensão dos splines no modelo. Este valor deve ser entre 0 e 1. O valor de 0 fará com que todos os splines passem direto em sua totalidade através dos nodes, e o valor de 1 maximiza a curvatura dos splines.

Set Number Precision - Esta opção faz com que a tela para configurar o número de precisão apareça. Isto é para configurar como irão aparecer os pontos decimais para os números na seção atual do FlexSim.

Set Conveyor Angle - O ângulo do conveyor determina o quão detalhado as curvas do conveyor estão desenhadas. Um ângulo pequeno significa uma curva suave. Este valor altera apenas a performance gráfica e não os resultados do modelo.

View Menu



Model View (3D) – abre uma nova janela de vista em perspectiva do modelo.

Model Tree - Esta opção abre a janela de visualização que mostra a pasta dos modelos. Isto mostra todos os objetos que estão no modelo. A tree pode ser manipulada deste janela.

Model Control GUI - Abre o control GUI do modelo. Esta é uma janela que usuário pode criar por conta própria para fazer operações que são específicas ao seu modelo. Veja o Graphical User Interfaces para mais informações.

Start Page - Abre a página inicial do FlexSim, o qual é visto quando você abre pela primeira vez o software FlexSim. A página inicial oferece a opção para criar um novo modelo ou abrir um modelo existente, acessar o Getting Started Guide e o manual do usuário, ajustando global preference e ter informações de licenciamento. Também mostra os últimos modelos acessados, informações sobre a licença que está ativa atualmente com a versão atual do FlexSim, e algumas informações sobre o conteúdo de internet ativado.

Open Default Workspace - Fecha todas as janelas ativas do windows no FlexSim e então abre qualquer janela salva como área de trabalho padrão da guia Environment da janela Global Preferences.

Modeling Utilities

As seguintes opções irão abrir vários utilitários de modelagem em uma janela no FlexSim. Por padrão, a biblioteca para arrastar e soltar objetos e a janela Quick Properties já estão abertas.

Drag-Drop Library - Abre o utilitário de bibliotecas. Em muitos casos, esta biblioteca atua como uma biblioteca de objetos, com uma grade contendo ícones de objetos que podem ser arrastados para dentro do modelo.

View Settings - Esta opção abre o utilitário de edição da forma de visualização do modelo..

Quick Properties - Abre o utilitário Quick Properties. Este utilitário fornece um rápido acesso as propriedades dos objetos e também dos flowitems que antigamente estavam disponíveis apenas na janela de propriedades.

Edit Selected Objects - Abre o utilitário Edit Selected Objects.

Find Objects - Abre o utilitário Find Objects.

Views - Abre o utilitário Views.

Groups - Abre o utilitário Object Groups.

Model Layouts - Abre o utilitário Model Layouts.

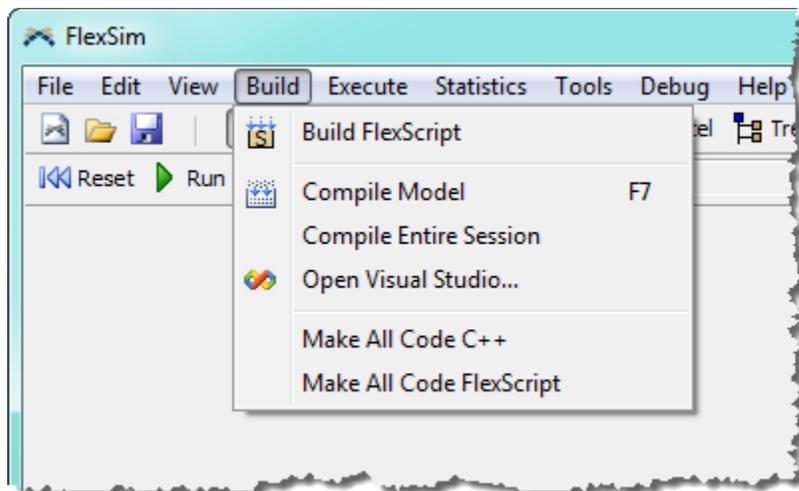
Measure/Convert - Abre o utilitário Measure/Convert.

Animations and Components - Abre o utilitário Animations and Componentes. Este é o botão da janela do Animation Creator. Clique com o botão direito sobre um objeto e selecione **Edit > Animations** para abrir o Animation Creator.

DatGuiase

Current DatGuiase Table - Esta opção abre uma janela que mostra a tabela do banco de dados atualmente ativa que foi aberta ou que estava em standby com os comandos dbopen(), dbchangetable(), dbsqlquery(). Acesse o sumário de comandos para mais informações sobre estes comando

Build Menu



Build FlexScript – Estrutura todo o código do FlexSim.

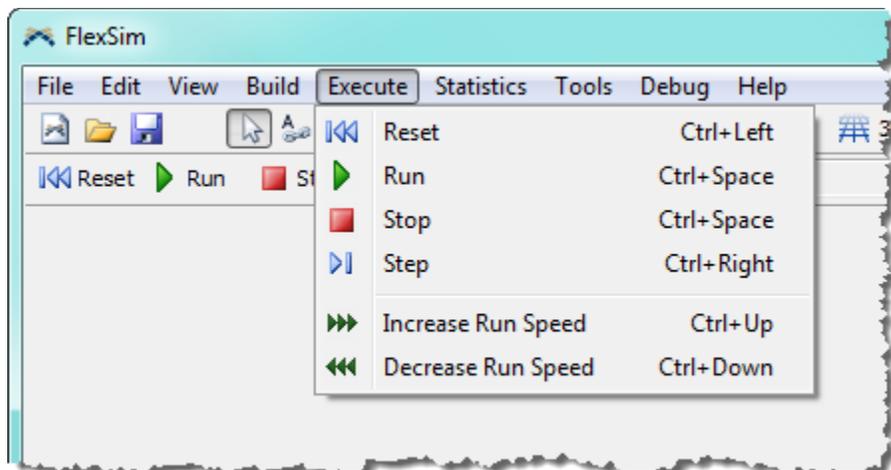
Compile Model – Compila todo código C++ no modelo. Este recurso pode também ser acessado a qualquer tempo, clicando em F2.

Compile Entire Session – Compila todo código C++ na janela principal. Você em geral não precisará fazer isso, a menos que esteja desenvolvendo sua própria aplicação com o FlexSim.

Open Visual Studio... – Abre o Microsoft Visual Studio.

Make all Code C++/Flexscript - Existem duas opções de fazer todos os códigos em C++ ou Flexscript. Nós fornecemos estas opções para que os modeladores possam escolher a melhor opção e a mais fácil para o uso do Flexscript (códigos funcionam imediatamente quando editados no Flexscript, sem a necessidade de compilar), assim como o run-speed do C++(uma vez que é compilado, roda muito mais rápido que o Flexscript). Enquanto que na fase de construção do modelo você pode usar o Flexscript, para que o seu código seja interpretado imediatamente após você escrevê-lo. Então, uma vez que o seu modelo estiver pronto para rodar, você pode escolher a opção Build > Make All Code C++, compilar, e rodar para conseguir a velocidade desejada para o C++.

Execute Menu



Reset - Esta opção reseta o modelo atual e prepara ele para rodar novamente. Funciona da mesma forma que clicar no botão Reset no painel de simulação. Também disponível ao manter a tecla Ctrl pressionada juntamente com a seta esquerda.

Run - Esta opção roda o modelo atual. Funciona da mesma forma que apertar o botão Run no painel de simulação. Também disponível ao manter o Ctrl e a barra de espaço pressionado para alternar entre rodar e pausar o modelo.

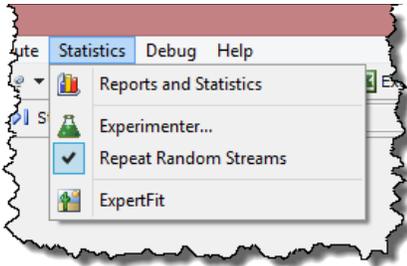
Stop - Esta opção para de rodar o modelo atual. Funciona da mesma forma que pressionar o botão Stop no painel de simulação.

Step - Esta opção leva o modelo forçadamente para o próximo evento dentro da listagem de eventos do modelo. Funciona da mesma forma que clicar no botão Step no painel de simulação do modelo. Também disponível ao manter pressionado o Ctrl e a seta para direita ao mesmo tempo.

Increase Run Speed - Esta opção aumenta a velocidade de simulação do modelo. Funciona da mesma forma que movimentar ou deslizar a barra para a direita no painel de simulação. Também disponível ao manter o Ctrl e a flecha para cima do teclado pressionados.

Decrease Run Speed - Esta opção diminui a velocidade de simulação do modelo. Funciona da mesma forma que movimentar ou deslizar a barra para esquerda do painel de simulação. Também disponível ao manter o Ctrl e a flecha para baixo do teclado pressionados.

Statistics Menu



Reports and Statistics - Esta opção abre a caixa do Reports and Statistics. Permite ao modelador criar vários relatórios baseados nas estatísticas capturadas durante o andamento da simulação do modelo. Estes relatórios podem incluir informações sobre a produtividade, tempo de permanência, estado dos flowitems e outros dados que o modelo pode selecionar ou customizar.

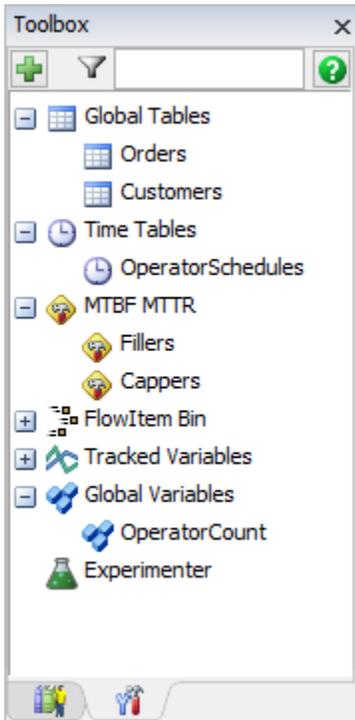
Experimenter - Esta opção abre a janela do Experimenter. Funciona da mesma forma que pressionar o botão do Experimenter na parte direita inferior da janela principal. O Experimenter é usado para realizar análises de vários cenários dentro da simulação para o seu modelo. Esta também é a forma de acessar o **FlexSim's Optimizer** na nova versão do FlexSim, o qual é usado para encontrar os valores ótimos para um conjunto de variáveis de decisão definida pelo usuário, seja para maximizar ou minimizar uma função objetivo também definida pelo usuário e que atenda todas as restrições. O Optimizer, utilizando os conhecimentos da **OptQuest**, requer licença especial e separada além de ser vendido separadamente da licença básica do Flexsim.

Repeat Random Streams - Esta opção repete os fluxos aleatórios cada vez que você reinicia e executa o modelo. Para fazer cada rotação diferente, desmarque esta opção.

ExpertFit - Esta opção, irá abrir o software ExpertFit desenvolvido por Averill Law & Associates. Um manual completo está disponível online com o software. Este programa é usado para determinar a melhor distribuição de probabilidade estatística que se ajusta aos seus dados de entrada.

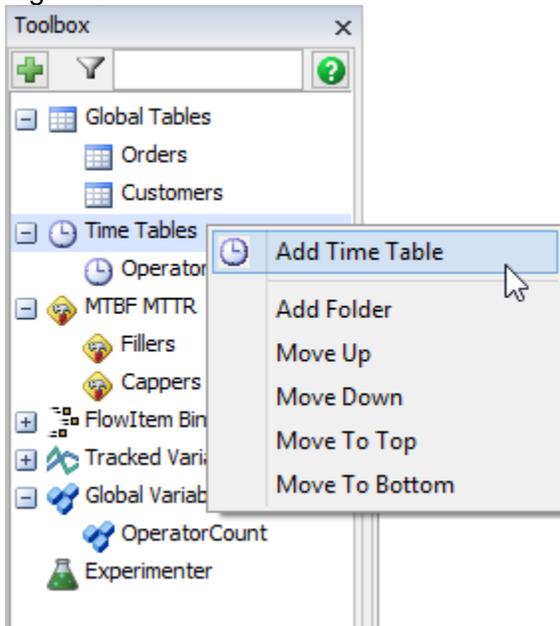
FlexSim Toolbox

FlexSim's Toolbox window is the central location where you manage your model's tools, such as Global Tables, Time Tables, Dashboards, etc. You access the Toolbox by choosing View > Toolbox from the main menu, or by choosing the  Tools button from the main toolbar.

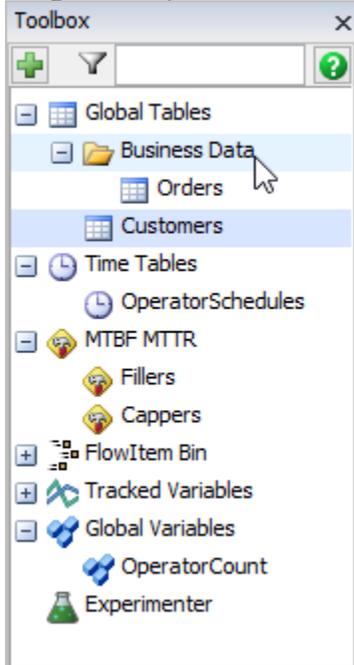


In the Toolbox you can:

- Click on the  button at the top of the window to add a tool to the model.
- Double-click on an item to go to its properties.
- Right-click on an item or folder to re-order it in the list, add a new folder, or add a new tool.



- Drag items up and down to reorder them, or drag them into folders to organize them hierarchically.

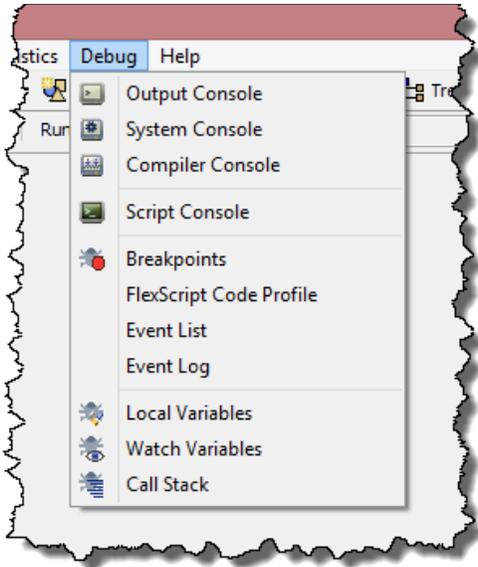


- Select an item and hit Delete to delete it from the model.
- Click in the filter edit box at the top of the window and type in the name of a tool you are looking for to filter by that name.
- Slow-lick twice on the name of an item to rename.

Although add-on modules may include additional tools, by default you can manage the following tools from the Toolbox:

- AVI Maker
- Dashboards
- Excel Import/Export
- Experimenter
- FlowItem Bin
- Global Tables
- Global Task Sequences
- Global Variables
- Graphical User Interfaces
- Model Background
- Model Triggers
- MTBF/MTTR
- Presentation Builder
- Reports and Statistics
- Time Tables
- Tracked Variables
- User Events
- User Commands
- Visio Import

Debug Menu



Output Console - Esta opção abre o painel onde as informações de saída são mostradas. Você pode fazer com que seja mostrado sua própria informação no console dos dados de saída usando os comandos: pt(), pr(), pd(), pf(), etc..Para mais informações sobre estes comandos, veja o sumário de comandos.

System Console - Esta opção abre o painel onde a informação sobre o status da arquitetura do programa é mostrado. Qualquer erro ou exceção será mostrado neste console.

Compiler Console - Esta opção abre o painel onde a informação é mostrado enquanto o modelo é compilado. Isto mostra o status de cada etapa do processo de compilação. Este console também mostra qualquer erro FlexScript quando você desenvolve o FlexScript.

Script Console - Esta opção abre o console scripting. Este painel é uma janela interativa para executar o comando FlexScript manualmente. Veja o console Scripting para mais informações.

Breakpoints - Esta opção abre a janela de pontos de interrupção. Para mais informações sobre esta janela e sobre os pontos de interrupção, consulte Step Debugging.

FlexScript Code Profile - Esta opção abre a janela Flexscript Code Profile, que lista todas as funções flexscript definidos no modelo e ordenados por quanto tempo o modelo passou executando-o. Esta é uma excelente maneira de descobrir como o computador está gastando o seu tempo e pode levar a ideias sobre a forma de tornar o modelo mais rápido.

Event List - Esta opção abre a janela da listagem de eventos. A lista de eventos é uma lista ordenada de todos os eventos pendentes. Veja o Event List Viewer.

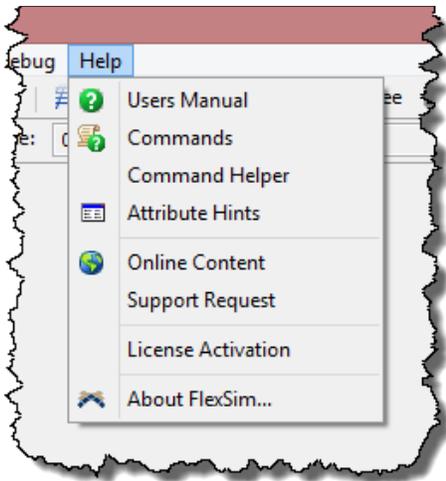
Event Log - Esta opção abre a janela de log de eventos. O log de eventos é uma lista ordenada de todos os eventos que acontecem. Veja o Event Log.

Local Variables - Esta opção abre o painel de variáveis locais (Local Variables), o qual mostra a você os valores atuais de qualquer variável localmente definida. Veja Local Variables para mais informações.

Watch Variables - Esta opção abre o painel Watch Variables, o qual permite a você especificar outras variáveis ou expressões que você quer ver, como por exemplo, variáveis globais. Veja Watch Variables para mais informações.

Call Stack - Esta opção abre o painel Call Stack, o qual mostra a atual pilha de chamadas, um histórico das funções de chamadas. Veja o Call Stack para mais informações.

Help Menu



Users Manual - Esta opção abre o manual do Flexsim.

Commands - Esta opção abre a lista de comandos em seu web browser padrão. Esta lista resume os parâmetros e o uso de cada comando disponível na programação do Flexsim.

Command Helper – Abre o painel de ajuda de comandos que permite você referenciar qualquer comando do FlexSim. Fornece detalhes, parâmetros e exemplos de cada comando.

Attribute Hints - Esta opção abre a janela Attribute Hints. Esta lista mostra todos os atributos do Flexsim e seu significado.

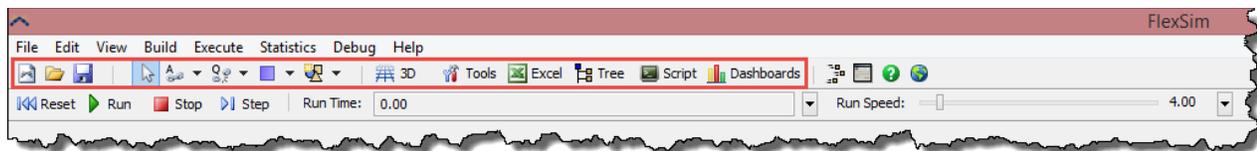
Online Content – usa um browser interno para visualizar e fazer o download online de conteúdos para o FlexSim, incluindo módulos, desenhos 3D, imagens e exemplos de modelos.

Support Request – Direciona para a página de solicitação de suporte, permitindo criar um ticket de suporte.

License Activation - Esta opção abre a janela de Ativação de Licenças.

About FlexSim... - Esta opção abre uma tela mostrando as informações sobre o FlexSim. Isso mostra o status da licença, a versão da FlexSim atualmente sendo executada, gráficos e informações de contato.

FlexSim Toolbar



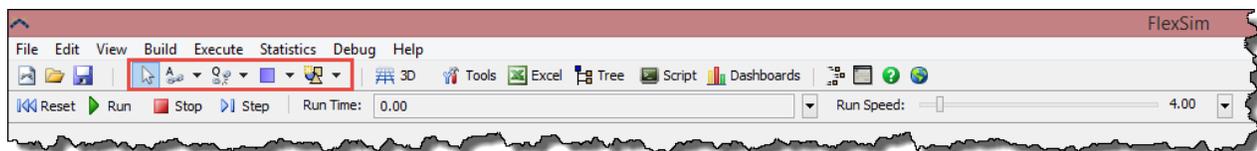
 **New:** Fecha o modelo atual e permite a você a iniciar a construção de um novo modelo. Uma caixa de diálogo irá aparecer perguntando se você gostaria de salvar antes de criar um novo modelo.

 **Open:** Permite a você abrir um modelo anteriormente salvo (arquivo .fsm). Uma caixa de diálogo irá aparecer perguntando se você gostaria de salvar antes de criar um novo modelo.

 **Save:** Permite a você salvar o modelo atual. Se o modelo nunca foi salvo anteriormente, será perguntado a você onde você deseja salvar o arquivo.

Mode Toolbar

A barra de ferramentas (mode toolbar) permite-lhe alternar entre diferentes modos na visualização em 3D. Você entra em um modo pressionando o botão de modo apropriado e pode sair do modo pressionando a tecla Esc ou pressionando um botão diferente. Alternativamente, você pode manter pressionado algumas teclas conforme definido na página de interação de teclas de atalho.



 **Standard Mode** - Este modo é padrão para visualização. Neste modo, você pode movimentar objetos, dimensioná-los e outros. Você pode retomar a este modo a qualquer instante, simplesmente, pressionando a tecla Esc.

 **Connect Objects** - Para conectar dois objetos neste modo, clique com o mouse sobre um objeto, arraste o mouse para o outro objeto, e solte o botão do mouse sobre este outro objeto. Você pode também criar múltiplas conexões em série neste modo. Para fazer isso, clique sobre um objeto, então clique sobre o outro objeto, então clique sobre o outro objeto e assim por diante. Isto funciona da mesma forma que manter a tecla A apertada

 **Connect Center Ports** - Este modo funciona da mesma forma que conectar dois objetos, exceto que existe a conexão de portas centrais ao invés de portas de entrada/saída. Este modo funciona ao manter a tecla S apertada

 **Extended Connect** - Este modo funciona da mesma forma que se conecta dois objetos, exceto que cria uma conexão estendida que varia dependendo dos objetos clicados. Este modo funciona, mantendo a tecla D apertada. Para mais informações sobre conexão estendida (também chamada de conexão "context sensitive"), veja a seção de interação com o teclado do computador.

 **Disconnect Objects** - Para desconectar dois objetos neste modo, clique sobre um objeto, arraste o mouse na direção do outro objeto, e solte o botão do mouse sobre este objeto. Você também pode deletar múltiplas conexões em série neste modo. Para fazer isso, clique em um objeto, e clique sobre o outro objeto e assim por diante. Isto funciona da mesma forma que manter a tecla Q do teclado pressionada.

 **Disconnect Center Ports** - Este modo funciona da mesma forma que o modo de desconectar objetos, exceto que ele desconecta portas centrais ao invés de portas de entrada e de saída. Isto funciona da mesma forma que manter a tecla W pressionada.



Extended Disconnect - Este modo trabalha da mesma forma que desconectar objetos, exceto que realiza a desconexão de extensões que pode variar dependendo do objeto clicado. Isto funciona da mesma forma que manter a tecla E apertada. Para mais informações sobre desconectar extensões (também chamada de conexão “context sensitive”), veja a seção de interação com o teclado do computador.



New Selection - Você pode criar conjuntos de seleção para ter operações sendo aplicadas a todo um conjunto de objetos. Para selecionar objetos neste modo, arraste uma caixa ao redor dos objetos que você deseja selecionar. Para selecionar nenhum, clique no campo onde se constrói o modelo. Isso funciona da mesma forma que segurando a tecla Shift.



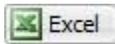
Toggle Selection - Você pode criar conjuntos de seleção para ter operações aplicadas a todo um conjunto de objetos. Para selecionar objetos neste modo, arraste uma caixa ao redor dos objetos que você deseja selecionar. Todos os objetos selecionados na caixa será desmarcada. Todos os objetos não selecionados na caixa ficará selecionada. Isso funciona da mesma forma que segurar a tecla Ctrl



Create Objects - Uma vez que você está neste modo, você pode simplesmente clicar sobre um objeto na biblioteca e, em seguida, cada vez que você clica no botão esquerdo do mouse no modo de visualização ortho, uma nova instância daquele objeto será criada. Isso funciona da mesma forma que pressionar a tecla F.



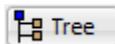
Create and Connect Objects - Este modo é semelhante ao modo anterior, exceto que quando um novo objeto é criado, ele será conectado com o objeto anterior, que foi criado com ('A'). Isso funciona da mesma forma que pressionar a tecla R.



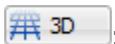
: Abre a caixa de diálogo de interface com o Excel.



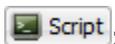
: opens the Toolbox window.



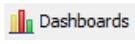
: Abre a visualização da árvore (tree) do modelo. Isto é útil para ver todos os objetos que atualmente estão no modelo, mesmo aqueles que não podem ser vistos através do Model View. Os atributos dos objetos podem ser editados aqui, no entanto, é recomendado que as propriedades dos objetos sejam alterados através da janela de propriedades dos objetos ou através do Quick Properties, e não através do tree.



: Abre uma nova visualização em perspectiva do modelo.



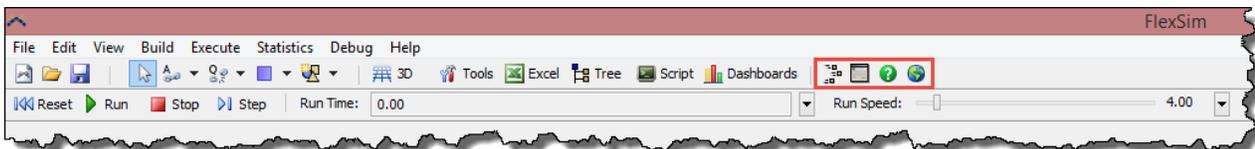
: Abre o console script. Aqui você pode executar os comandos FlexScript para alterar ou conseguir informações de seu modelo.



: opens a dashboard. Here you can gather statistics from your model.

Custom Toolbar

Esta seção da barra de ferramentas pode ser customizada com qualquer opção do menu principal. Para customizar esta seção, vá em **File > Global Preferences** e selecione a guia Customize Toolbar. A seguir algumas opções que podem ser adicionados a barra de ferramentas:



Flowitem Bin: Abre o editor do FlowItem Bin. Nesta janela, o usuário pode editar os tipos de flowitems atualmente disponíveis no FlowItem Bin. Atributos dos flowitems, como tamanho, forma, cor, podem ser alterados; método de embalagem podem ser especificados; e animações podem ser adicionados. Novas classes de flowitem podem também ser criados, e os não usados pode ser deletados.



Model Control GUI: Abre o control GUI do modelo. Esta é uma janela que pode ser criado pelo próprio usuário para fazer as operações que são específicas para o seu modelo. Para mais informações, veja o Graphical User Interfaces.

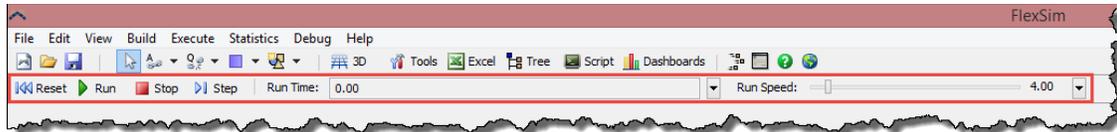


Getting Started: Abre a seção de ajuda do FlexSim Getting Started.



Sample Models: **Abre a seção de downloads de modelos do website da FlexSim em um web browser.**

Simulation Run Panel



Este painel é utilizado para controlar a simulação. Muitas das funções estão disponíveis através do menu Execute no menu principal. O painel para rodar a simulação é encontrado na parte superior da janela principal do FlexSim, e nós iremos descrevê-lo aqui em três seções:

Buttons

Reset: reseta o modelo. A função OnReset é chamada para cada objeto no modelo. Isto deve sempre ser selecionado antes de rodar o modelo.

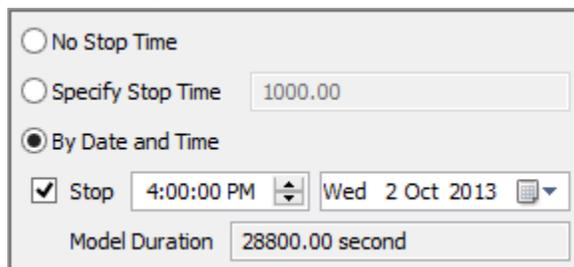
Run: Inicia a simulação do modelo. O relógio de simulação do modelo irá continuar a avançar continuamente até que o modelo é parado ou a lista de evento esteja vazia.

Stop: Para o modelo enquanto ele estiver rodando. Também atualiza os estados de todos os objetos no modelo. O modelo não está resetado. Pode ser iniciado novamente do mesmo ponto no tempo em que foi parado.

Step: Configura o relógio do modelo para o tempo do próximo evento que irá acontecer. O próximo evento então acontece. Isto permite aos usuários passarem para a execução de um evento a um determinado tempo. Quando muitos eventos são programados para acontecer no mesmo tempo, podem não ser visível no modelo quando esta opção é selecionada.

Run Time: Mostra o tempo de simulação atual do modelo.

Time/Stop Time

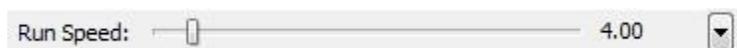


No Stop Time: Selecione esta opção para o modelo rodar indefinidamente. O modelo ainda pode ser parado se o botão “Stop” for clicado ou se a lista de eventos estiver vazia.

Specify Stop Time: Selecione esta opção para especificar a unidade de tempo que o modelo irá rodar antes de parar. O modelo ainda pode ser parada a um certo tempo usando a opção picklist de qualquer objeto relevante no modelo. Nas trigger “OnEntry” ou “OnExit”, a opção “Stop the ModelRun” pode ser usado para dinamicamente parar a simulação do modelo.

By Date and Time: Se esta opção é selecionada, o campo do tempo de simulação irá mostrar a data e tempo atual do modelo. O tempo e data do início da simulação é definido em seu Model Settings. Especifique uma duração para simular baseado em uma data e tempo específico. Se a caixa “Stop” não estiver selecionada, o modelo irá rodar indefinidamente do tempo de início; se o tempo de parada for determinado, a duração do modelo será a diferença entre os dois tempos. O recurso “By Date and Time” irá trabalhar com as Time Table que estão sendo usadas em Graphical Time Table.

Speed

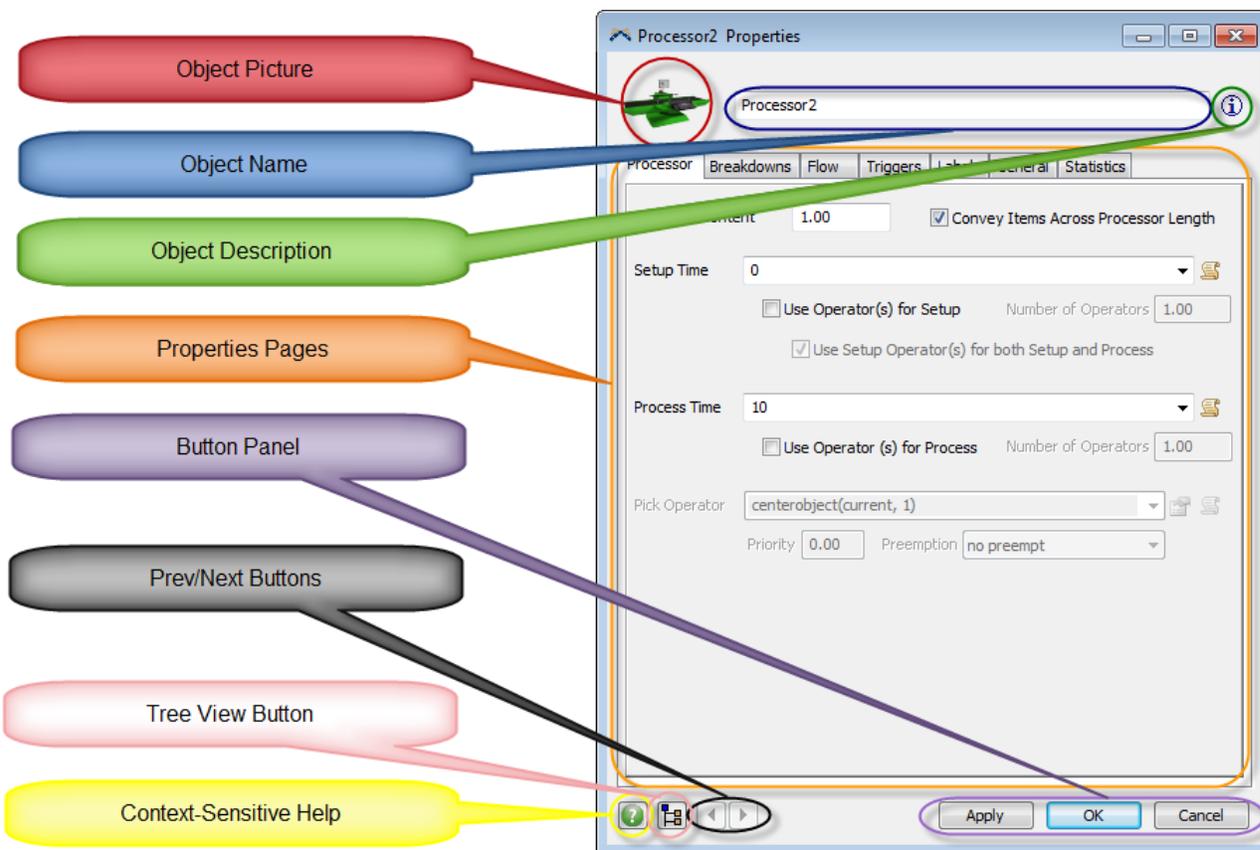


Speed slider: Esta barra define o número de unidades do tempo do modelo que o Flexsim irá tentar calcular por segundo do tempo real. O resultado no momento podem ser próximo deste valor conforme o modelo vão necessitando de mais processamento em cada evento. O valor pode ser alterado ao clicar na flecha em preto do lado direito e entrando com o valor manualmente.

Object Properties Windows Overview

A janela Propriedades permite que você configure os atributos que são dependentes do tipo de objeto que você está editando. Por exemplo, uma fila tem um máximo de teor de atributo, enquanto que um combinador tem uma lista de componentes para receber flowitems. A janela de propriedades da fila é, portanto, diferente da janela Propriedades do combinador. No entanto, muitas vezes há semelhanças entre janelas Properties. Por exemplo, tanto a fila e o combinador pode ter uma estratégia de envio para. Objetos que têm em comum geralmente contêm partes de suas janelas de propriedades que são do mesmo modo que você possa mais rapidamente aprender a usar essas janelas.

Para abrir página de Propriedades de um objeto, clique duas vezes sobre o objeto ou botão direito do mouse no objeto e selecione a opção "Propriedades" no menu pop-up.



Object Picture

Esta imagem mostra o tipo de objeto que você está editando.

Object Name

Aqui você pode dar ao objeto um nome diferente. Ao especificar o nome do objeto, não utilize caracteres especiais, como >, <, *, -, (,), etc Isso pode causar Flexsim para não se comportar corretamente. Espaços e sublinhados devem ser os únicos caracteres não alfanuméricos utilizados. Além disso, não começar o nome com um número. Você deve pressionar Apply ou OK para que esta alteração seja aplicada. É aconselhável dar a cada objeto em seu modelo de um único nome para o relatório e para outros fins. Uma mensagem de erro aparecerá quando você pressionar Apply se outro objeto tem o mesmo nome que o nome que você digitou para este objeto.

User Description

Este botão abre um campo de edição onde você pode digitar uma descrição ou notas sobre o objeto. Esta descrição é salvo com o objeto e pode ser visto em um momento posterior da mesma forma que é editado: ao pressionar este botão. Clique em qualquer lugar na janela para o campo de edição Descrição do Usuário de aplicar e desaparecer.

Properties Pages

Propriedades das janelas são constituídos por um conjunto de páginas de guia, dependendo do objeto. Muitas páginas são usadas por vários objetos. Esta referência fornece a documentação para cada página de propriedades. As páginas de propriedades estão listados a seguir. Dentro de cada página de ajuda estão os links para cada objeto que utiliza essa página.

Páginas compartilhadas - Estas páginas são utilizadas por vários objetos

Fixed Resource Pages – Estas páginas são usadas pelos objetos Fixed Resourced

- **BasicFR Advanced**
- **Breakdowns**
- **Combiner**
- **Conveyor**
- **Decision Points**
- **Flow**
- **Layout**
- **MergeSort Flow**
- **MultiProcessor**
- **Photo Eyes**
- **Processor**
- **ProcessTimes**
- **Queue**
- **Rack**
- **Separator**
- **Sink**
- **SizeTable**
- **Source**

Task Executer Pages – Estas páginas são usadas pelos objetos Task Executer.

- **ASRSvehicle**
- **BasicTE**
- **Breaks**
- **Collision**
- **Crane**
- **Dispatcher**
- **Geometry**
- **Robot**
- **TaskExecuter**
- **Transporter**

Fluid Pages – Estas páginas são usadas pelos objetos fluído.

- **Blender**
- **FluidConveyor**
- **FluidLevelDisplay**

- FluidProcessor
- FluidToltem
- Generator
- Initial Product
- Inputs / Outputs
- ItemToFluid
- Layout
- Marks
- Mixer
- Percents
- Pipe
- Recipe
- Sensors
- Splitter
- Steps
- Tank
- Terminator
- Ticker

Shared Pages – Estas páginas são compartilhadas pelos objetos múltiplos.

- General
- Labels
- Statistics
- Triggers

Other Pages

- Container Functionality
- Display
- NetworkNode
- NetworkNodes
- Speeds
- Traffic Control

Button Panel

Apply - Aplica as alterações feitas na janela do objeto.

OK - Aplica-se as alterações feitas na janela com o objeto e fecha a janela.

Cancel - fecha a janela sem aplicar quaisquer alterações feitas na janela. Algumas propriedades do objeto são atualizadas imediatamente quando mudá-los na janela. Essas mudanças ainda serão aplicadas e não serão desfeitas quando pressionar Cancelar. Exemplos dessas propriedades imediatamente mudanças são avarias, etiquetas, aparência, posição, rotação, tamanho e portos.

Prev/Next Buttons

Estes botões irão aplicar quaisquer alterações feitas na janela para o objeto e mudar o foco da janela para o objeto anterior ou seguinte do mesmo tipo de classe na árvore. Isso é útil para rapidamente modificar as propriedades de vários objetos com o mesmo tipo de classe.

Tree View Button

Este botão abre a janela de edição de árvore que mostra apenas este objeto

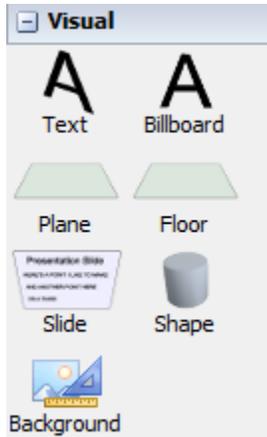
Context-Sensitive Help

Este botão irá abrir o Manual do Usuário para a página que descreve a guia selecionada no momento.

VisualTool

1. Overview
2. Exemplo

VisualTool Overview



Visual Tools são usados para implementar algo útil para o usuário no espaço de modelagem, como por exemplo, textos, cenários, slides de apresentação para dar mais realidade ao modelo. Ele pode ser algo bem simples como uma caixa colorida, um background, ou alguma coisa bem elaborada como um gráfico em 3D ou slide de apresentação. Outra utilização de uma Visual Tool é uma superfície contenedora para outros objetos no modelo. Quando usado desta forma, o Visual Tool se torna uma ferramenta hierárquica para organizar seu modelo. Esta superfície contenedora pode ser salva em uma biblioteca de usuário atuando como um bloco de construção básico no desenvolvimento de modelos futuros.

Detalhes

Visto que o VisualTool trabalha diferentemente de outros objetos do Flexsim, uma descrição detalhada de como ela é usada será feita. Existem muitas formas que você pode usar o objeto Visual Tool em seu modelo.

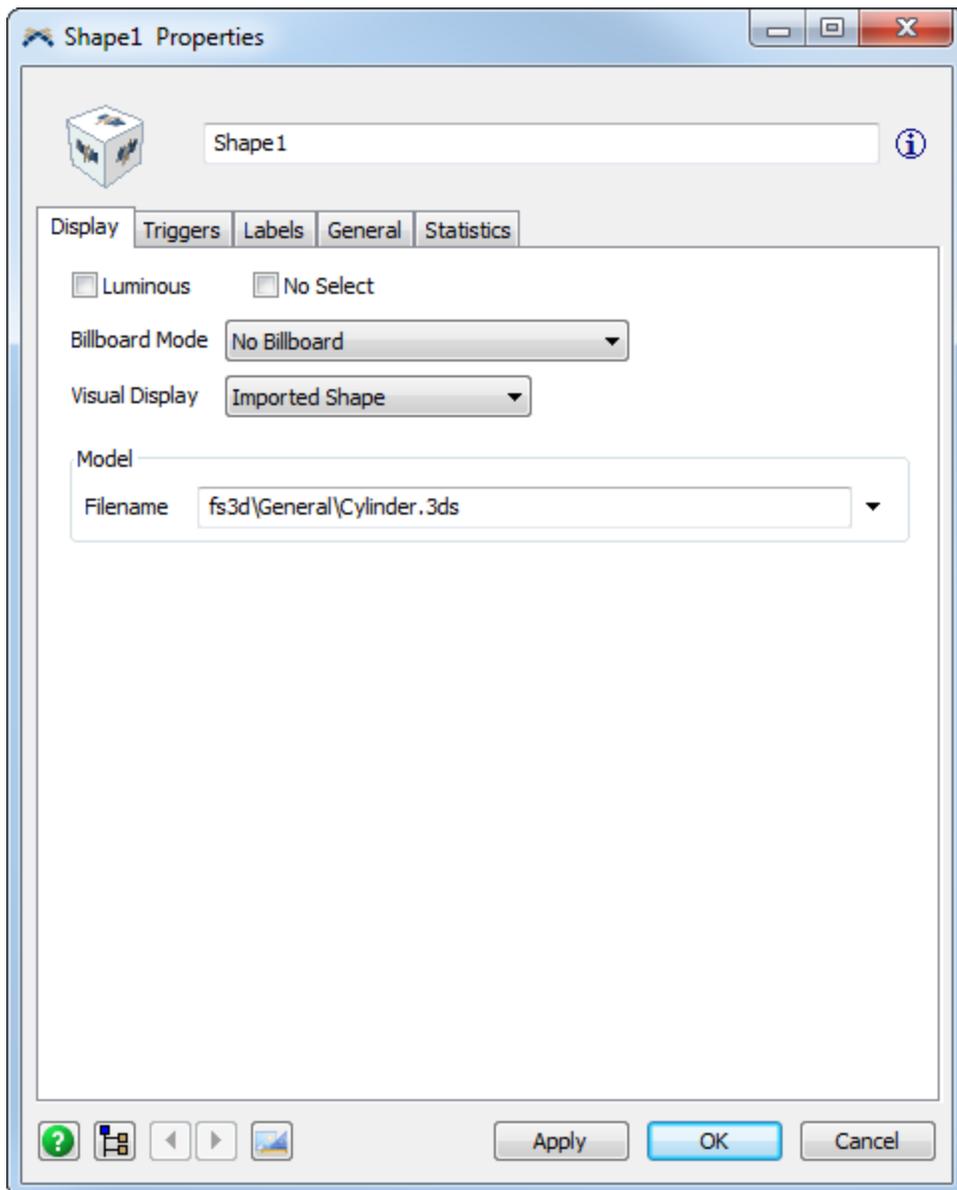
- Como um container ou submodelo
- Como um plano, cubo, coluna ou esfera
- Como uma forma importada
- Como um texto
- Como um slide de apresentação

Usando o VisualTool como uma superfície contenedora

A configuração básica para o Visual Tool é um plano. Quando colocado no modelo, a VisualTool irá mostrar um plano em cinza. O tamanho e a localização do lugar pode ser configurado graficamente na janela de visualização ortográfica ou usando a guia Display do Visual Tool (o uso da guia Display é feito na seção Usando o Visual Tool como um Plano, Cubo, Coluna ou esfera). Quando usando o VisualTool como uma superfície contenedora é recomendado que você use a visualização padrão (um Plano), configurando quando você iniciar o uso dela. Você pode alterar o visual de representação no futuro. Para este exemplo, nós iremos construir uma superfície contenedora que tem 1 queue e 2 processors. Os Flowitems irão passar dentro da superfície contenedora através de um source que está localizado fora da superfície. Os processors irão enviar flowitems para um sink localizado fora da superfície contenedora. Estatísticas sobre a superfície contenedora podem ser vistas através da guia estatísticas na janela de propriedades do Visual Tools.

Usando o VisualTool como um Plano, Cubo, Coluna ou Esfera.

Para usar o Visual Tool como uma ferramenta visual em seu modelo, é um processo simples. Na guia Display, escolha o tipo de recurso que você quer com o que o Visual Tool se pareça e definir as propriedades.



Plano

Um plano pode ser definido como um background para importar por exemplo um layout de Autocad, uma textura ou imagem, ou um conjunto de cores para uma seção específica do seu modelo. O plano é uma vista padrão para a visual tool. Você simplesmente configura o tamanho do plano e então escolhe a textura. A textura pode ser repetida em ambas as direções vertical e horizontal.

Cubo, Coluna ou Esfera

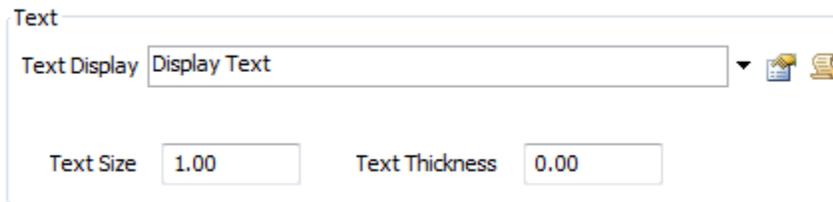
O cubo, coluna ou esfera são formas simples que podem ser atribuídas uma textura muito parecido ao plano.

Usando o Visual Tool como uma forma Importada

Quando usamos o Visual Tool para uma forma importada você precisa ter um modelo 3D ou objeto que você queira trazer para dentro do modelo. O Flexsim suporta uma ampla variedade de formatos de arquivos em 3D como por exemplo, o 3D Studio Max (.3ds), Google SketchUp (.skp), VRML 1.0 (.wrl), AutoCAD DXF (.dxf) e Stereo Lithography (.stl), além de outros.

Usando o VisualTool como um Visual Text

Um texto com visual em 3D pode ser adicionado em seu modelo para mostrar as labels, estatísticas, ou outras informações em seu modelo. Quando o display visual é configurado para “Text” será apresentado uma opção de picklist para configuração de um texto.

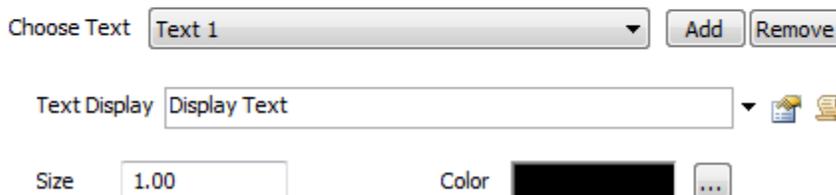
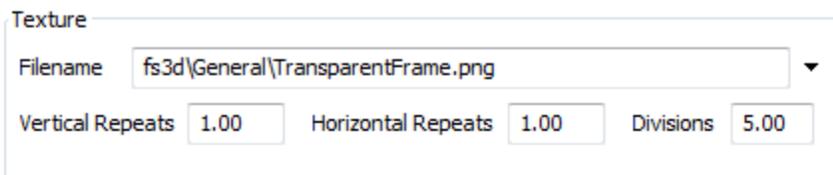


Os pick options incluem tempo de simulação, conteúdo, estado, outputs, inputs, e muitos outros. Se nenhuma estatística estiver selecionado nos picklist, o usuário deve conectar a porta central do Visual Tool ao objeto que você quer que a informação seja exibida. O texto pode ser editado ao selecionar o botão do template de código .

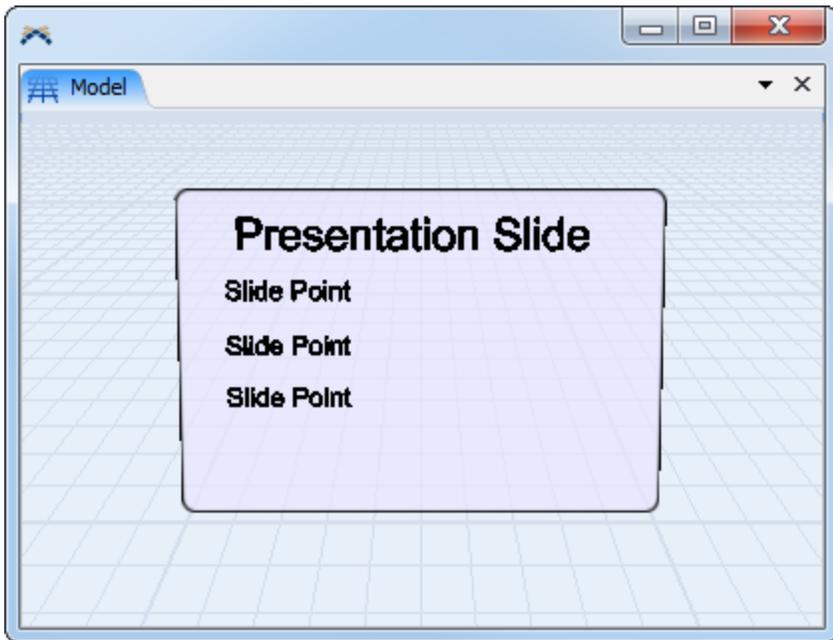
Usando O Visual Tool como um slide de apresentação.

O Visual Tool pode ser usado como um slide de apresentação muito parecido com o que você usa como um slide para criar apresentações em PowerPoint. Os slides de apresentação são colocados em seu modelo para apresentar dados, buscar algo no modelo e pontos de apresentação. Você pode desenvolver uma sequência “FlyTo” para visualizar os slides ao usar o Presentation Builder encontrado no menu Tools>Presentation>Presentation Builder.

Quando o display visual é configurado para “Presentation Slide” opções adicionais irão aparecer na guia Display.



Pressione o botão Add para adicionar uma nova linha de texto para o slide. A primeira linha do texto é rotulada “Text 1” e é um cabeçalho do slide. Textos adicionais são adicionados aos slides como itens de linha. Por exemplo, se você for adicionar 4 linhas no slide de apresentação, você verá o seguinte:



A cada texto é dado a localização padrão do slide conforme mostrado. Ao selecionar o texto desejado na caixa drop down, você pode alterar o texto que é mostrado usando a listagem Text Display, o tamanho do texto e a cor.

Você pode aplicar um background ao selecionar uma textura na guia Display ou alterar a cor do background ao selecionar a cor na guia General.

Páginas de propriedades

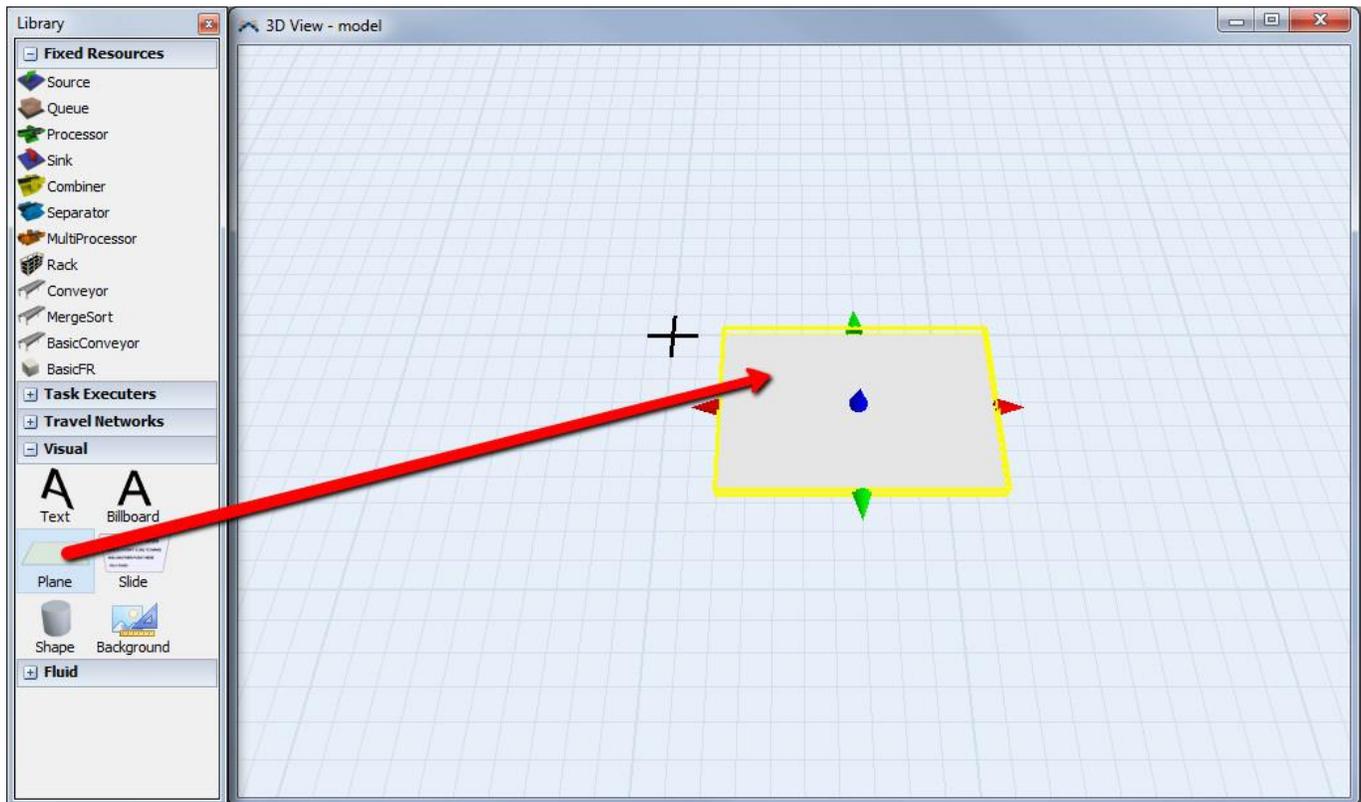
- Display
- Container Functionality
- Triggers
- Labels
- General
- Statistics

Exemplo de utilização da Visual Tool

Usando o Visual Tool como uma superfície contenedora

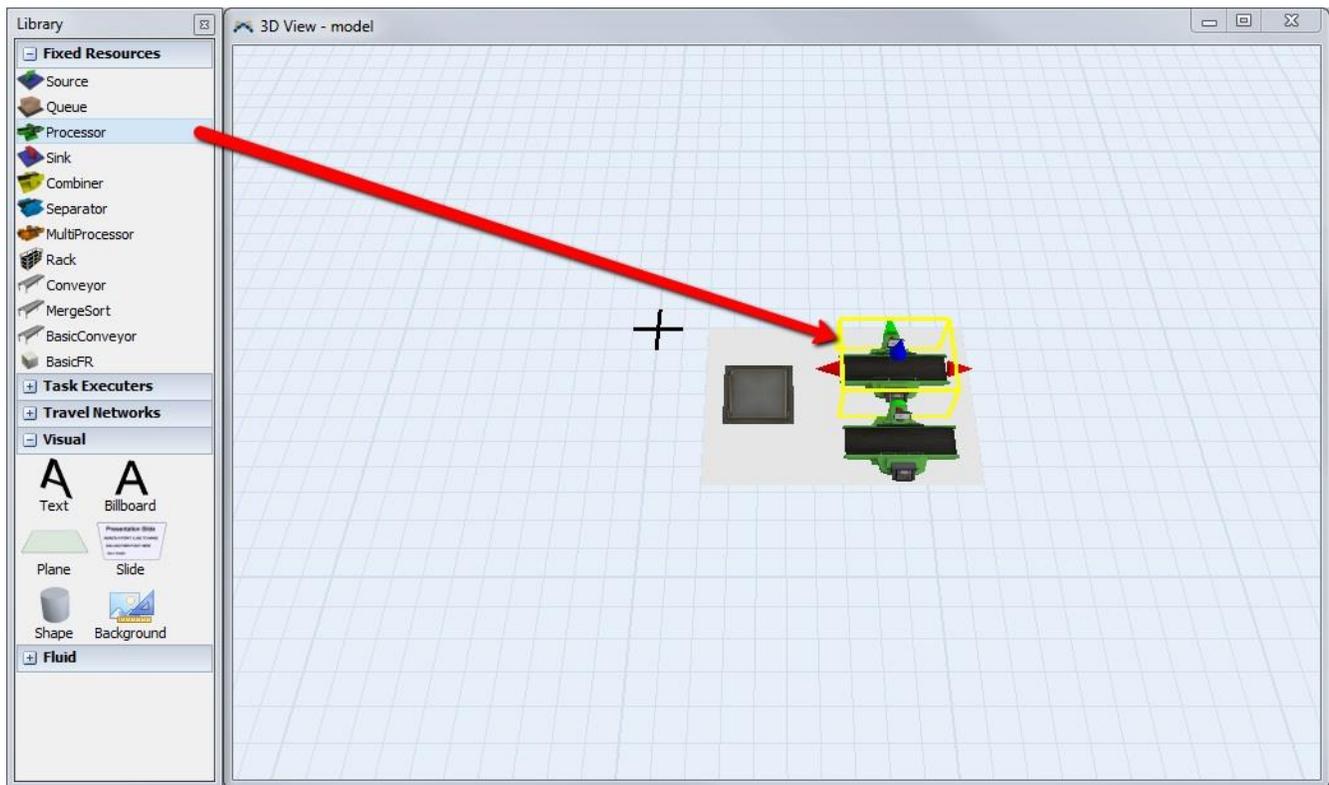
Etapa 1: Coloque uma Visual Tool dentro da superfície de simulação

Para adicionar uma VisualTool para o modelo, simplesmente arraste-o da biblioteca para dentro do modelo.



Etapa 2: Arraste e solte um Queue e 2 Processors em cima da Visual Tool.

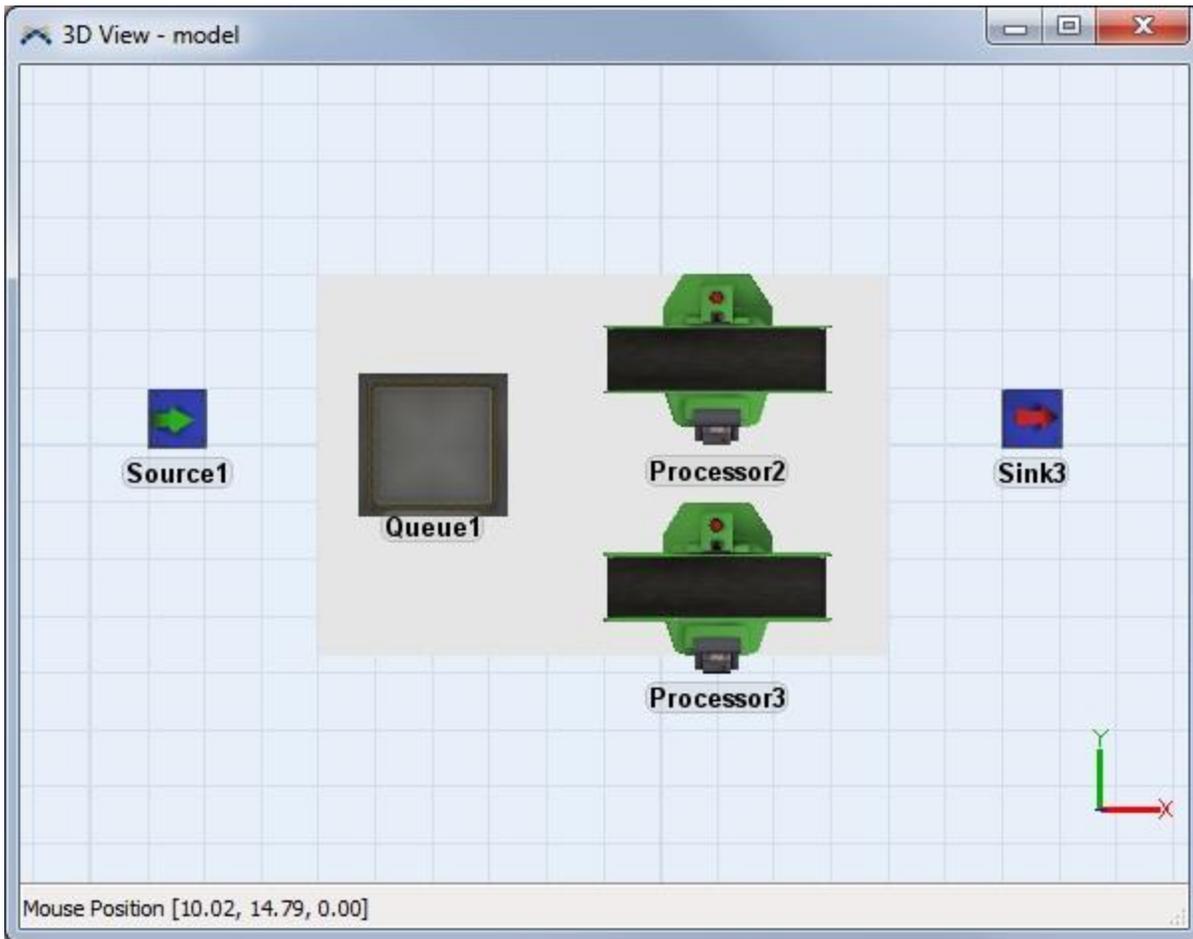
Para adicionar objetos dentro da superfície contenedora, simplesmente arraste os da biblioteca e coloque-os sobre o objeto VisualTool.



Quando você coloca um objeto sobre uma Visual Tool ele automaticamente será colocado dentro de um objeto Visual Tool. Você pode testar isto, selecionando a Visual Tool e movendo ela de localização com o seu mouse. Quando você movimenta o Visual Tool os objetos dentro irão se movimentar também.

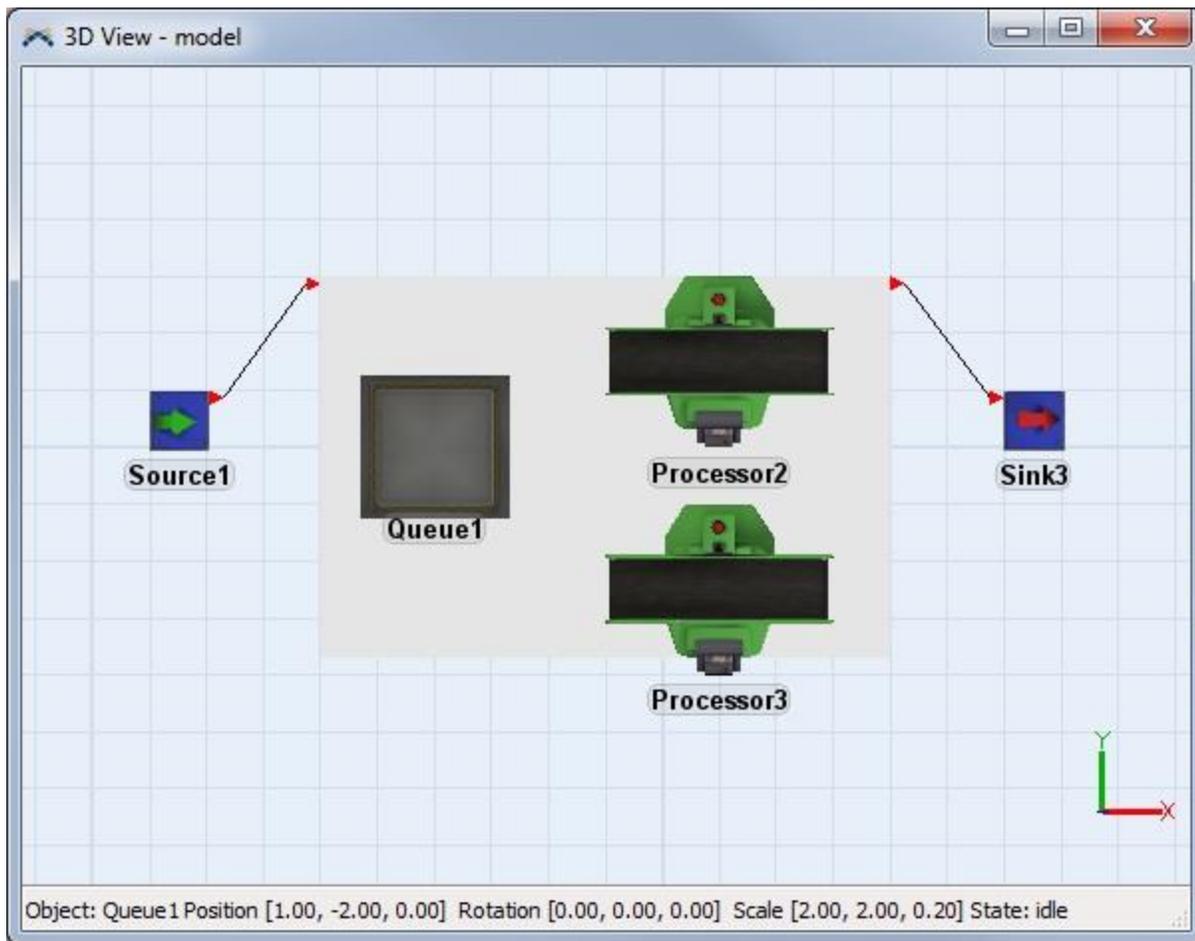
Etapa 3: Arraste um 1 Source e 1 Sink para dentro da superfície de simulação

Quando você colocar o Source o Sink dentro do modelo, assegure-se que você não os colocou dentro do VisualTool, você precisa assegurar-se que eles estão fora do Visual Tool.



Etapa 4: Conecte o Source ao Visual Tool e o Visual Tool para o Sink.

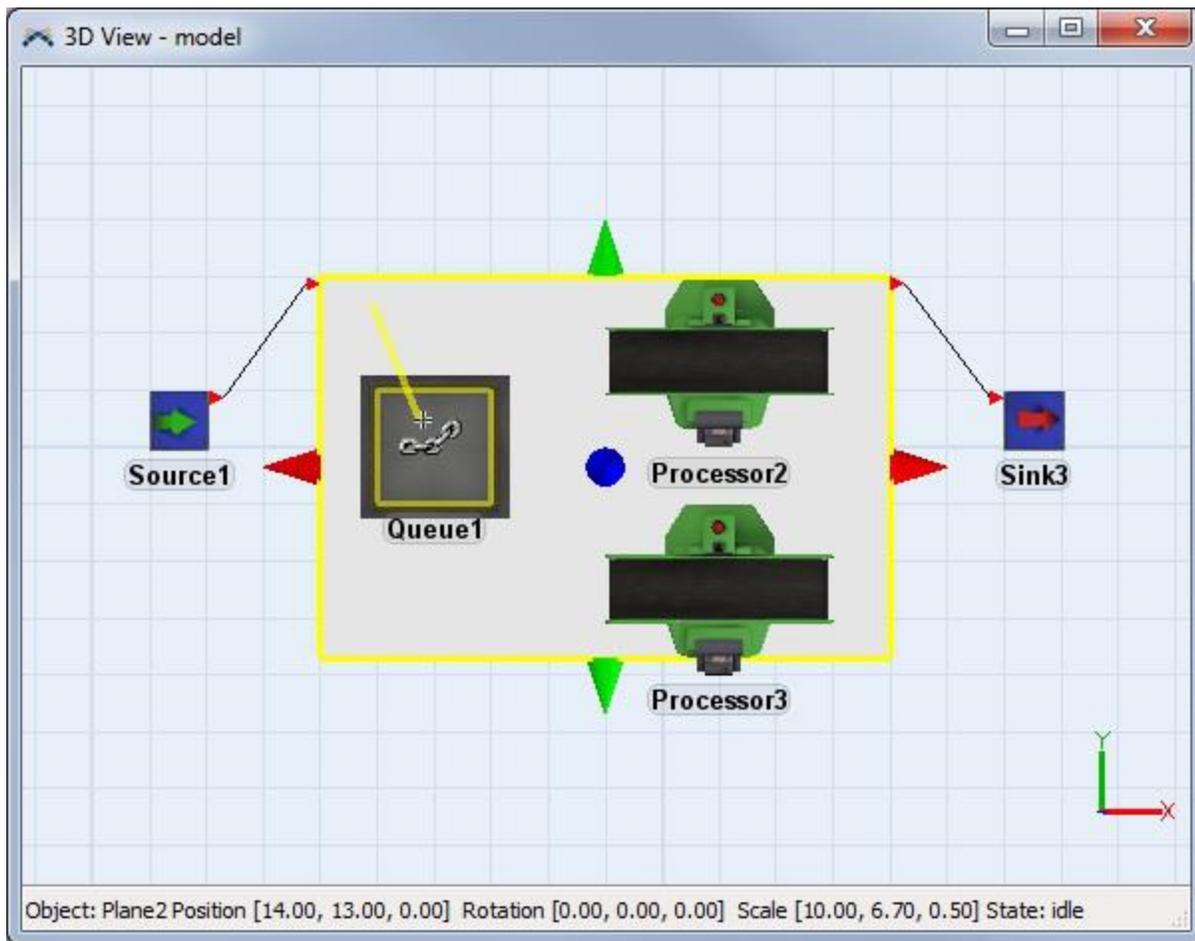
Enquanto pressiona a tecla "A" do teclado, clique e arraste a conexão do Source para a Visual Tool (não ao Queue). Quando você soltar o botão esquerdo do mouse você verá uma conexão feita entre o Source e a Visual Tool. Faça a mesma coisa para fazer uma conexão entre o Visual Tool e o Sink.



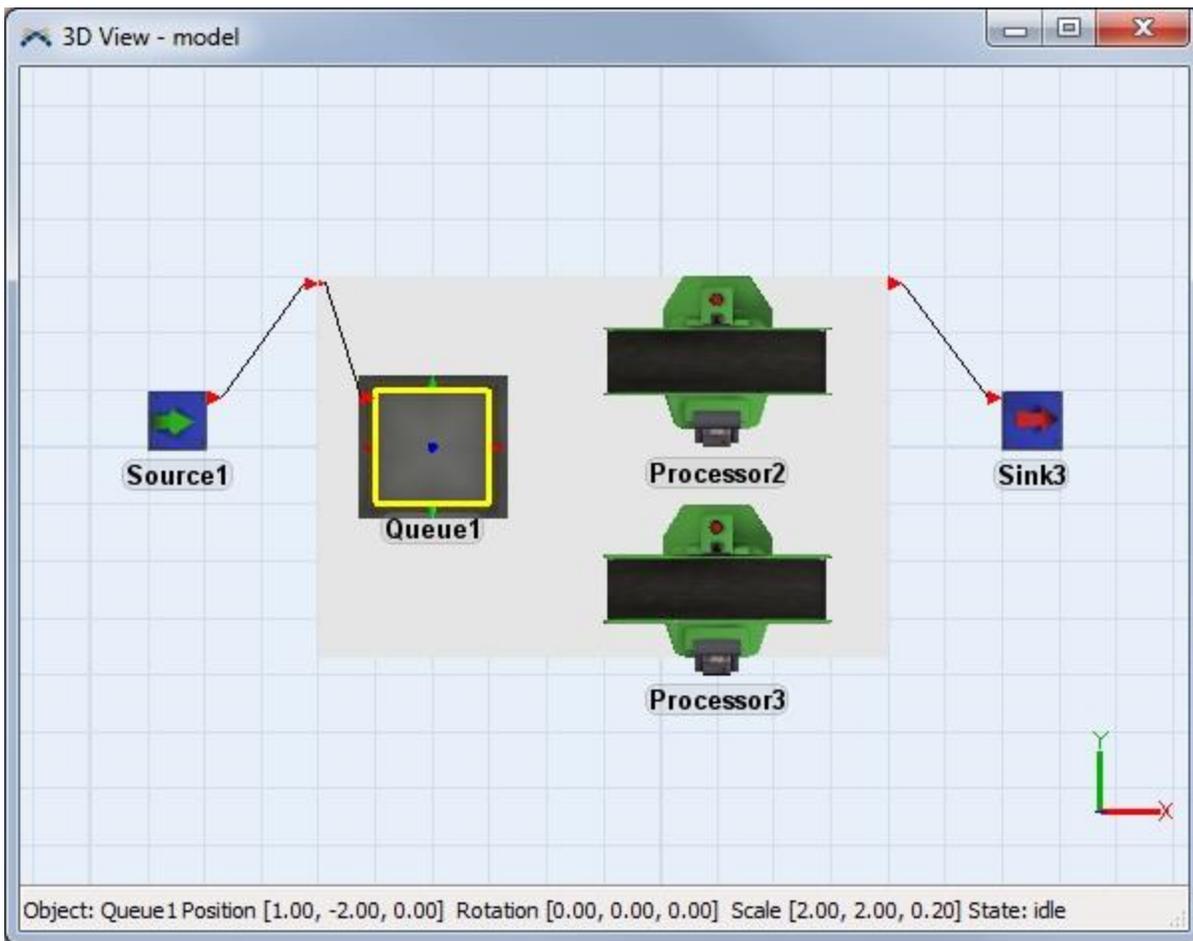
Neste momento, o Source e o Sink são conectados a superfície contenedora (Visual Tool). Agora, nós iremos conectar a superfície contenedora para o restante do modelo.

Etapa 5: Conecte a superfície contenedora ao Sink

Arraste uma conexão entre a superfície contenedora e o Sink.

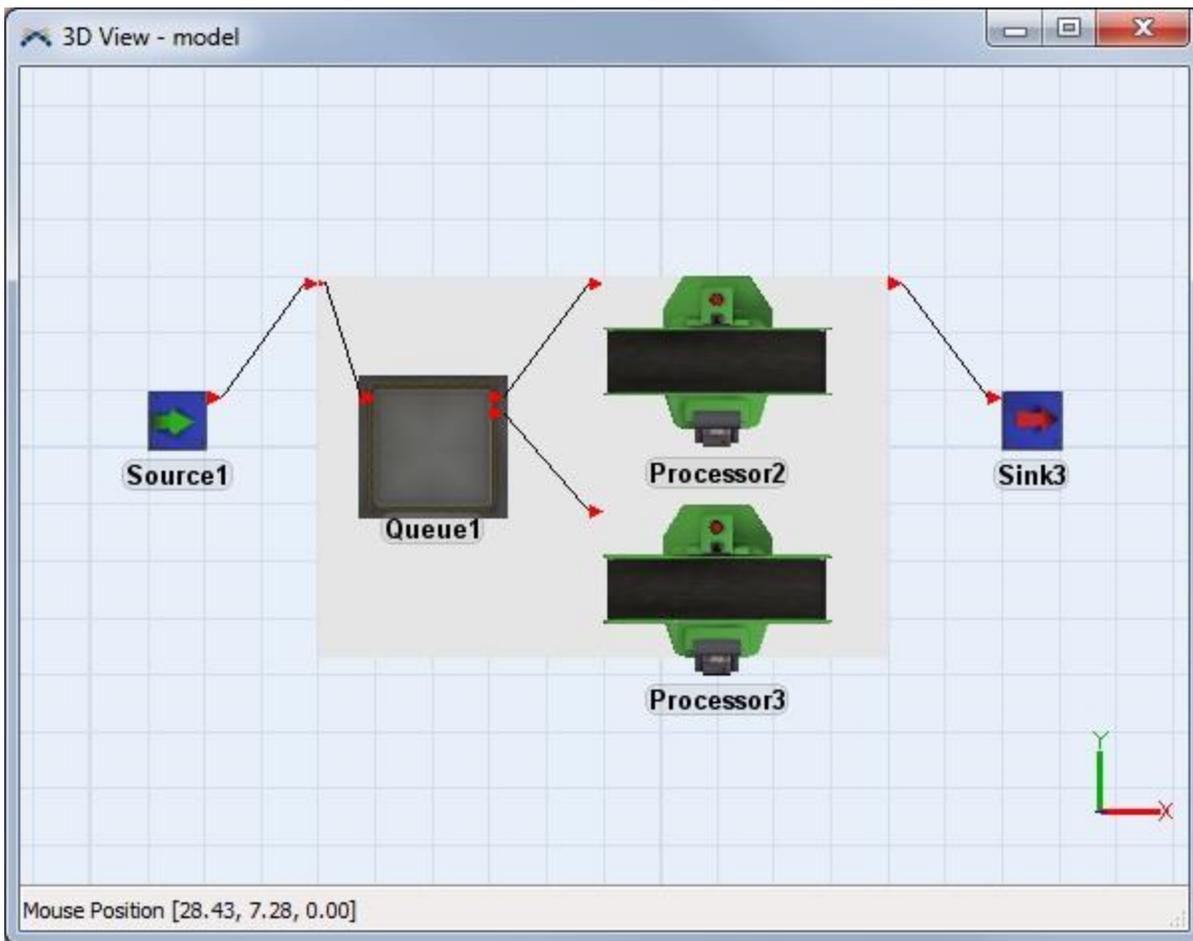


Quando você solta o botão esquerdo do mouse, você verá uma conexão da porta interna (uma seta menor) para o Queue.



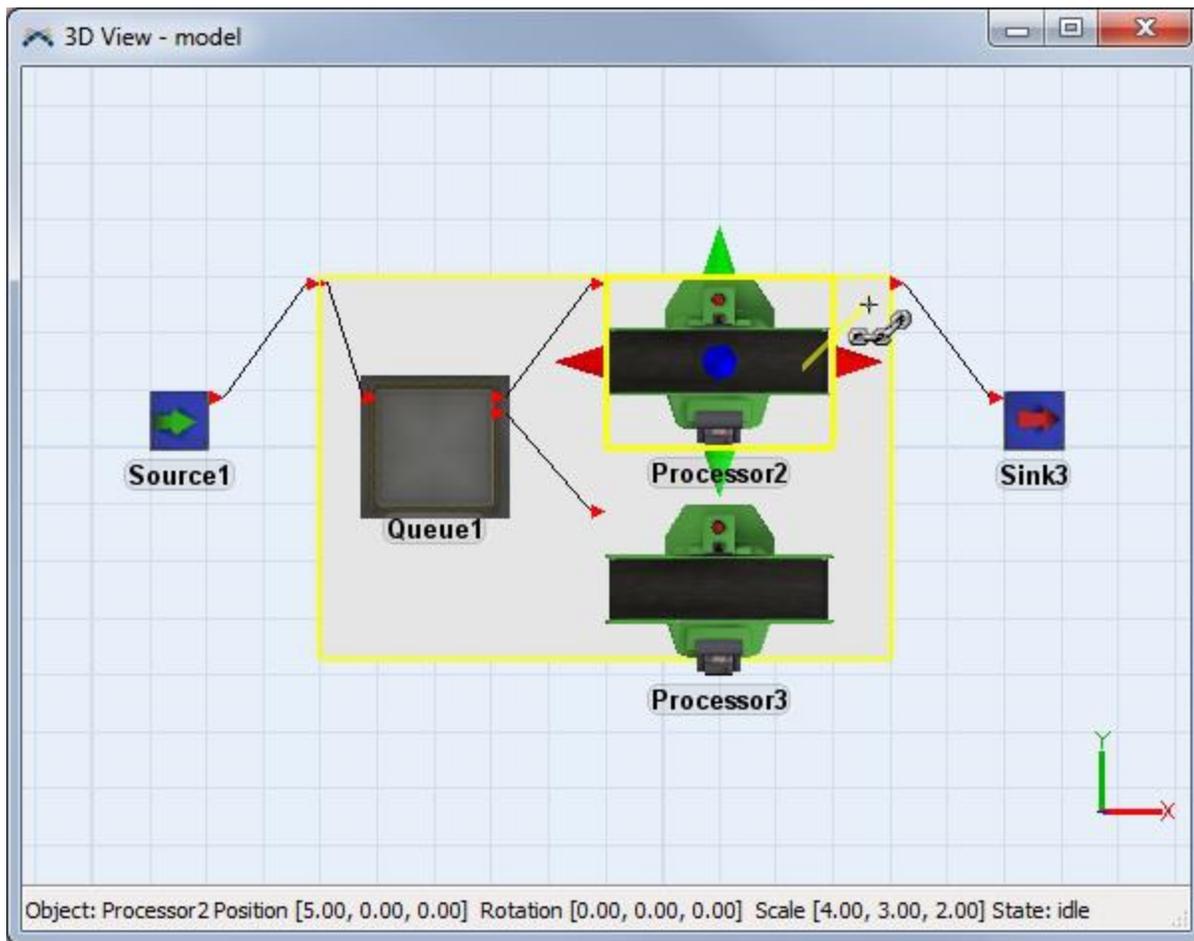
Etapa 6: Conecte o Queue aos Processors

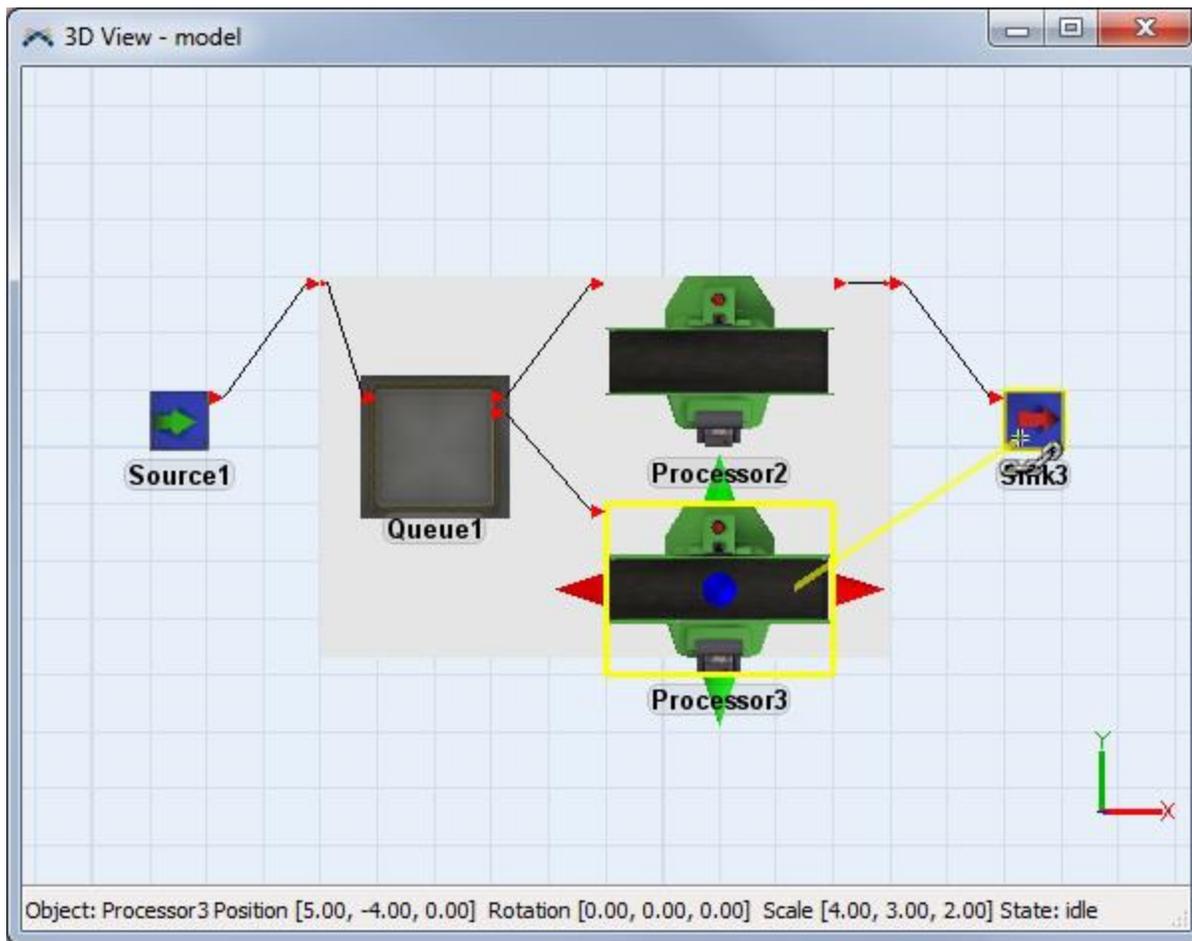
Siga o mesmo procedimento para conectar o queue aos 2 processors.

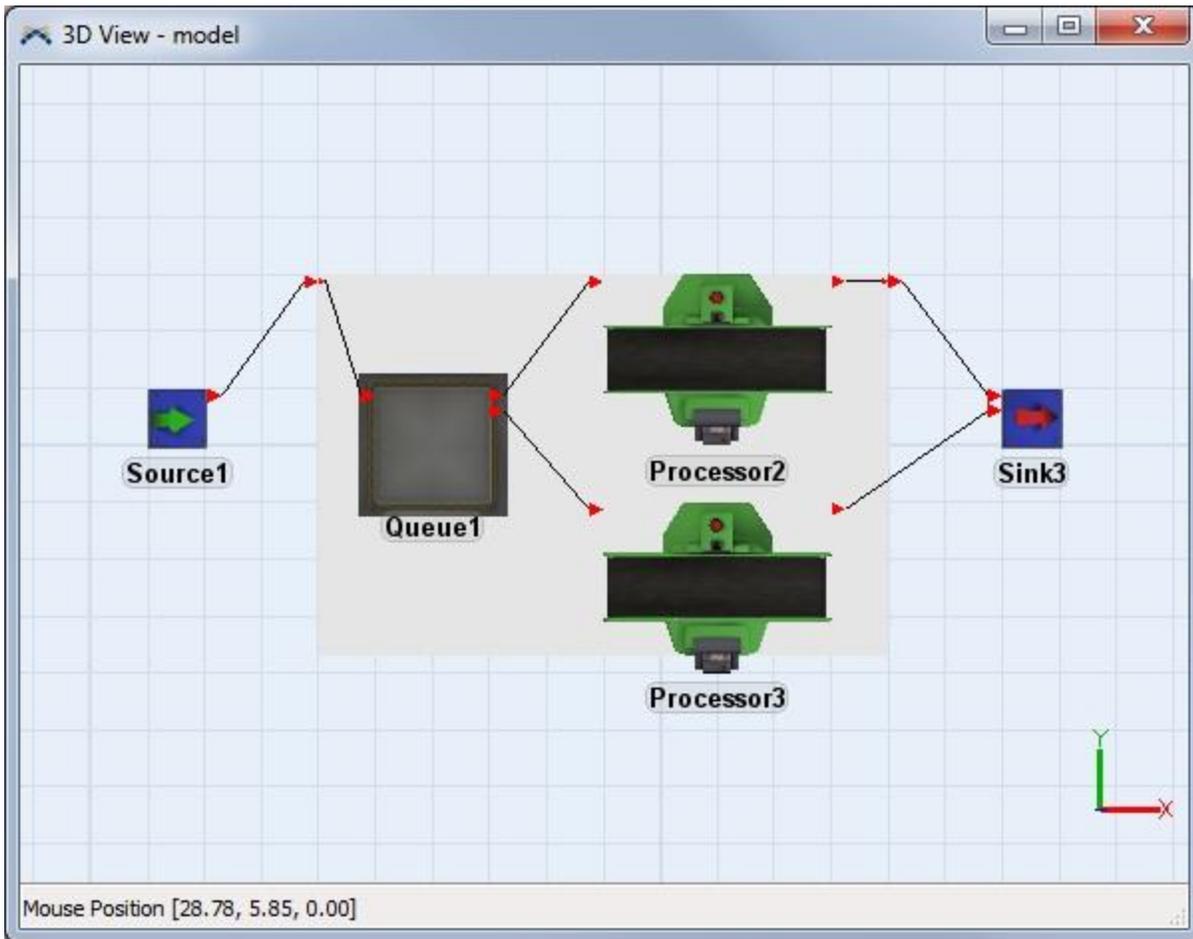


Etapa 7: Conecte os processors para a superfície contenedora ou diretamente para o sink

Existem duas formas de conectar “dentro” ou “fora” da superfície contenedora. A primeira forma conforme mostrado na etapa 4 quando uma conexão foi feita do Source para a superfície contenedora e então da superfície contenedora para o queue. Você pode no entanto, conectar diretamente do Processor para o Sink ao clicar e arrastar um link. Para este exemplo, o primeiro processor será conectado a superfície contenedora o qual então conecta-se ao Sink, e o segundo Processor se conectará diretamente para o Sink.



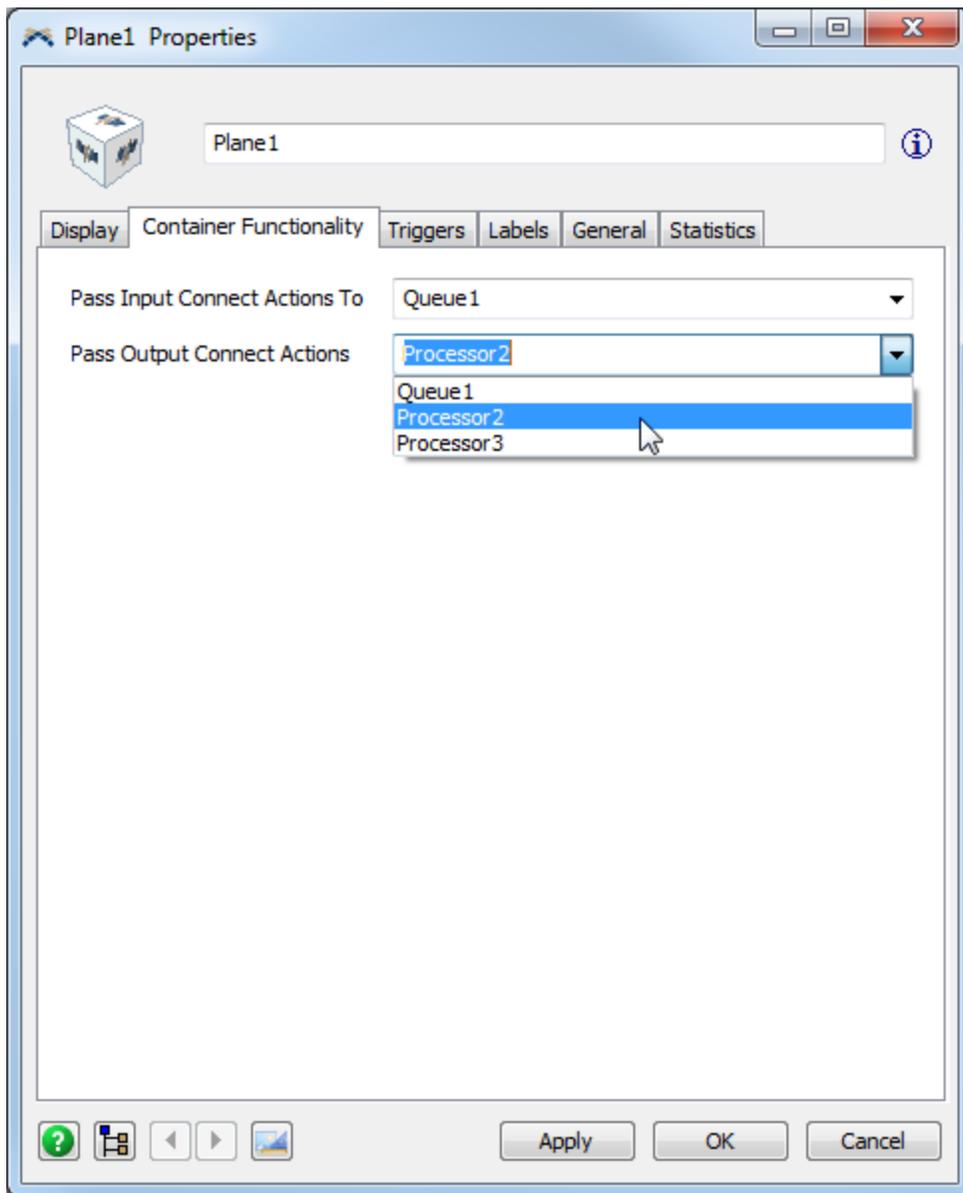




Etapa 8: Configure as funcionalidades da superfície contenedora

Neste momento, você agora tem uma superfície contenedora que possui dentro um sub-modelo. Você pode deixar a superfície contenedora como está, de forma que os itens que estão acima dele estão visíveis e editáveis, ou você pode esconder os equipamentos para que a superfície contenedora se torne uma caixa preta completamente encapsulada. Para esconder os demais equipamentos, desmarque a opção **Show Contents** na caixa guia General da página de propriedades do Visual Tool.

Configurando-o para a funcionalidade de black box você pode encontrar a guia Container Functionality da página de propriedades do Visual Tool. Configure o "Pass Input Connect Actions To" para o objeto que deve estar conectado na porta de entrada do Visual Tool. Configure "Pass Output Connect Actions" para o objeto que deve ser conectado na porta de saída do Visual Tool. A porta interna e porta de saída será conectado quando um objeto externo é conectado para a porta de entrada externa da superfície contenedora ou porta de saída.



Você pode também usar qualquer opção da guia Display para apresentar o que pertence aquela superfície contenedora como um caixa, forma em 3D, ou texto. Os itens de dentro da superfície contenedora pode ser vistos a qualquer momento ao clicar com o botão direito do mouse sobre o Visual Tool na janela Ortho e selecionando View > View Contents.

Ferramentas de Modelagem

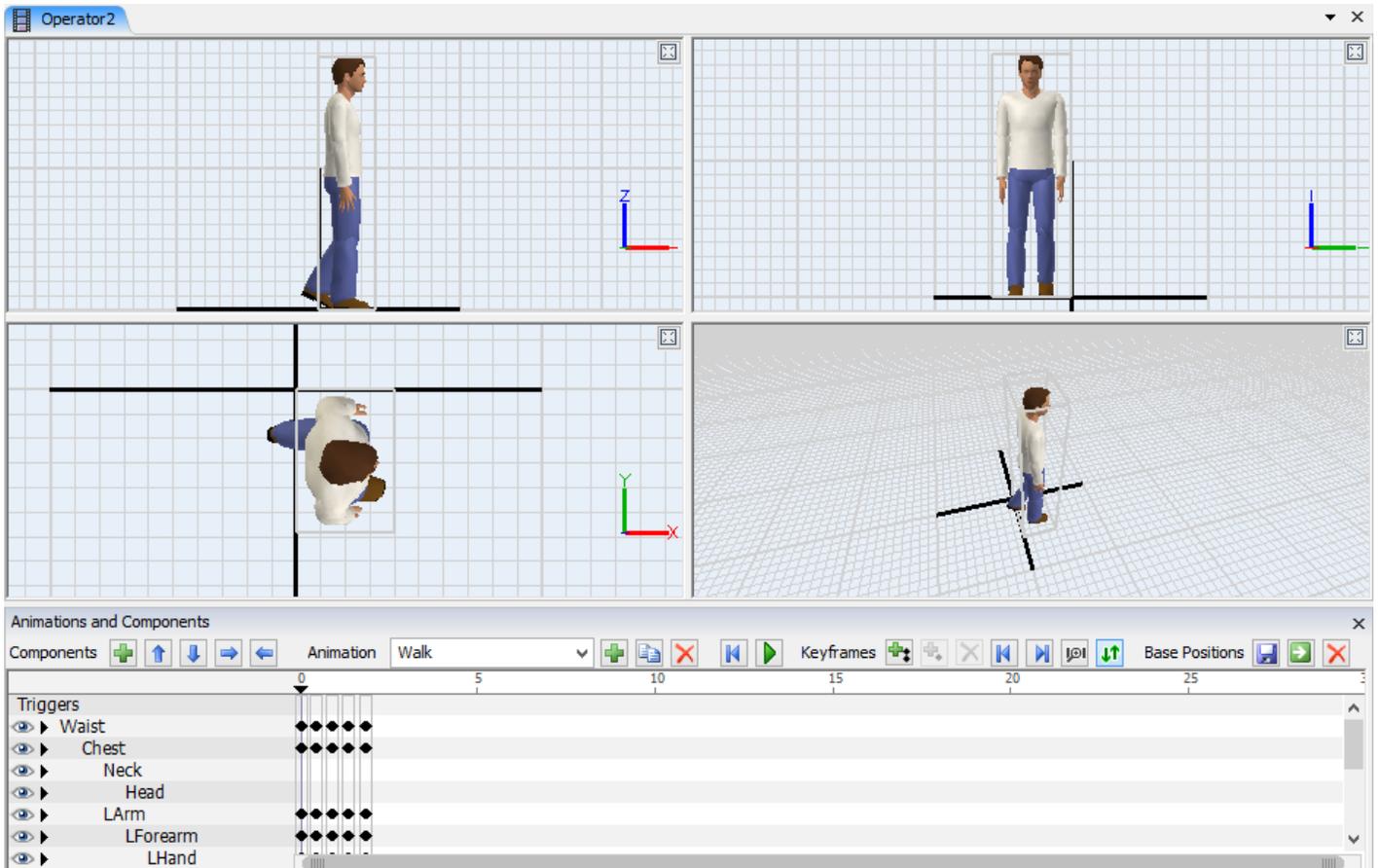
Esta seção contém informação sobre diferentes ferramentas de modelagem que você usará na construção, configuração e obtenção de resultados do seu modelo. Muitas destas ferramentas podem ser acessadas através da ferramenta menu, enquanto outras são acessadas de outras áreas do software.

- 1. Animation Creator**
- 2. AVI Maker**
- 3. Flowitem Bin**
- 4. Global Tables**
- 5. Time Table**
- 6. Graphical User Interfaces**
- 7. Import Media Files**
- 8. Model Background**
- 9. Visio Import**

Animation Creator

1. Conceitos
2. Exemplo
3. Referencia

Conceitos do Animation Creator



Tópicos

- Usando Animações
- Components
- Grade de Biblioteca de Objetos
- Repetindo Animações
- Time Basis
- Variáveis de Animação
- Keyframe Triggers
- Surrogates
- Comandos

O Animation Creator pode ser acessado através de um clique com o botão direito do mouse na superfície de simulação 3D através de **Edit > Animations** ou através da guia General da janela de propriedades dos objetos. O Animator permite você criar e salvar animações customizadas para os objetos do FlexSim em seu modelo.

No FlexSim, as animações são feitas de componentes e Keyframes. Componentes são somente formas em 3D dentro de seu objeto. Keyframes são valores que são salvos para diferentes propriedades de seus componentes. O Animation irá automaticamente gerar keyframes para dar animações suaves aos movimentos. Um objeto pode ter qualquer número de componentes e cada componente pode ser qualquer número de keyframes. Keyframes podem ser usados para especificar a posição, tamanho, cor e shape

frame do componente. As animações podem também ter suas triggers customizadas que ativam certos keyframes na animação. Estas opções permitem você adicionar animações sofisticadas para seus objetos. Para criar/editar animações, dê um duplo clique em qualquer objeto para abrir a janela de propriedades, clique na guia General, e clique no botão Edit que está na mesma linha de Visuals/Animations.

Usando Animações

Animações podem ser usadas em qualquer ponto em seu modelo. Um exemplo de animação é o operador caminhando. O operador não é uma forma em 3D, além do mais é feito de 16 diferentes formas. Estas formas, ou componentes, cada uma tem seus próprios keyframes que mudam suas posições e rotações conforme o operador caminha. A animação de caminhar é baseada na distância viajada, mais do que no tempo.

Animações podem ser usadas para fazer com que o processor se pareça mais realista, ou para mostrar um flowitem alterando em um processo. Você pode iniciar ou parar estas animações customizadas através dos comandos FlexScript.

Componentes

Componentes são formas em 3D dentro de seu objeto. Eles podem ser manipulados usando keyframes para alterar suas posições, tamanho, rotação etc..No operador, conforme visto acima, existem componentes como o pulso, pescoço, cabeça, peito, cintura etc...

Grade de Biblioteca de Objetos

Quando o Animation Creator é ativo, a grade de biblioteca de objetos muda para mostrar uma lista de formas:



Você pode filtrar a lista de formas ao digitar o nome dentro do campo . Formas que foram importadas para dentro do seu modelo irão aparecer na parte inferior de sua biblioteca. Você pode arrastar e soltar formas da biblioteca para dentro da superfície de simulação em 3D para adicionar um componente para seu objeto com aquela forma.

Repetindo Animações

Animações podem ser repetidas usando os seguintes tipos:

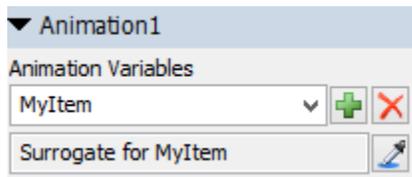
- **Do Not Repeat** - A animação irá parar logo que estiver completa.
- **Repeat Indefinitely** - Faz com que a animação retorne para o início sempre.
- **Repeat Set Number** - Faz com que a animação retorne para o início X vezes.
- **Repeat After Time** - Isto irá fazer com que a animação retorne após X vezes (ou distância) independente se a animação foi completada ou não. Por exemplo, uma animação levando 10 segundos com um tempo de repetição de 5 segundos irá somente rodar metade de sua animação.
- **Time After Animation End** - Uma vez que a animação terminou, o animation irá aguardar X vezes (ou distância) antes de repetir os movimentos novamente do início.

Time Basis

Quando uma animação é configurada como Time Based, a animação irá rodar baseado na velocidade de executar a simulação. Os números mostrados no timeline são na unidade de tempo do modelo (conforme definido na janela Model Settings).

Travel-Distance Based - Quando uma animação é configurada como Travel-Distance Based, a animação irá rodar conforme o objeto se movimenta no espaço em 3D. O número mostrado no timeline são unidades de distância do modelo (conforme definido na janela Model Settings), em vez de tempo. Uma exemplo de uma animação Travel-Distance Based , é uma animação de um operador caminhando.

Animation Variables

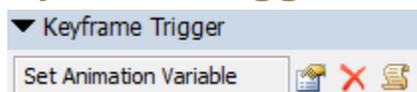


Variáveis de animação podem ser adicionados para qualquer animação. Estas variáveis pode ser usadas para dinamicamente alterar seus valores em sua animação. Por exemplo, alterando a posição ou rotação de um componente, alterando a cor de um componente, ou mesmo alterando um shape frame. Variáveis de animação são pontuados para os parâmetros sobre seus componentes. Na imagem acima, MyItem é uma variável de animação apontando para o Surrogate no componente MyItem. A variável pode ser configurada ao usar o comando `setanimationvar()`. Neste caso, a variável de animação é apontado para um objeto. Variáveis de animação podem ser pontuados como:

- Surrogates
- Keyframe Times
- Time Gaps
- Component Keyframes (Position, Size, Rotation, Color, Shape Frame)

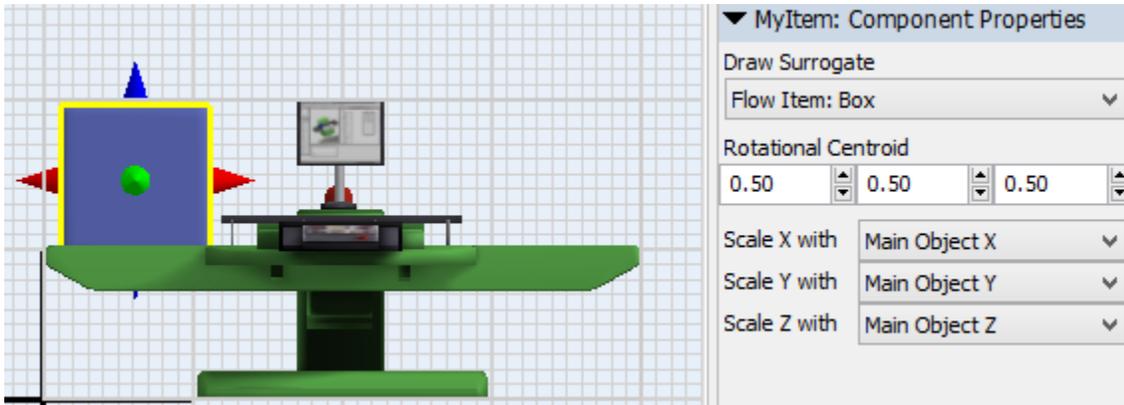
Se você quer dinamicamente alterar o comprimento de sua animação baseado nos parâmetros em seu modelo em certo tempo, você pode criar uma variável de animação que aponta para um espaço no tempo, e então configurar a variável de animação através da trigger keyframe ou da trigger em seu modelo.

Keyframe Triggers



Triggers podem ser adicionadas para animações que serão disparadas quando uma animação chegar a algo específico (ou distância). Elas são adicionadas através do timeline. Triggers permitem a você dinamicamente alterar ou atualizar variáveis de animação, parar animação, alterar alguns parâmetros em seu modelo, ou executar qualquer outro código Flexscript.

Surrogates



Surrogates são usados quando você quer substituir um componente em sua animação dinamicamente, com um objeto no modelo. Por exemplo, conforme visto na imagem acima, nós temos um componente chamado MyItem que atualmente tem um Surrogate desenhado como uma caixa. A forma que é desenhado para o surrogate não é importante e pode não ser mostrado no modelo. Isto é puramente para conveniência do modelador.

Quando você adiciona um Surrogate, será solicitado a você criar uma variável de animação. Esta variável de animação pode ser configurado para um objeto no modelo que será desenhado no lugar deste componente. Uma vez que você fez um surrogate, você pode especificar seu Rotational Centroid e determinar como será escalonado o objeto no modelo.

Surrogates podem ser configurados ao usar o comando `setanimationvar()` de algum lugar em seu modelo, ou você pode especificar uma trigger keyframe em sua animação, que irá configurar a variável de animação. Para um exemplo do uso do surrogate, veja o exemplo do Surrogate.

Comandos

Os seguintes comandos lidam com as animações:

startanimation(obj, animation) - Inicia uma animação de um objeto. Animações podem ser ambos, uma string value que é o nome da animação, ou pode ser um número o qual é um animation rank.

stopanimation(obj, animation) - Para uma animação de um objeto. Animação pode ser ambos uma string value que é o nome da animação, ou pode ser um número o qual é um animation rank.

resumeanimation(obj, animation) - Retoma uma animação de um objeto que foi anteriormente parado. Animações podem ser ambos, uma string value que é o nome da animação, ou pode ser um número o qual é um animation rank.

getanimationvar(obj, animation, "varname") - Retorna o valor de uma variável de uma animação.

setanimationvar(obj, animation, "varname", value) - Configura o valor de uma variável de animação. Os valores podem ser um número ou um objeto e é baseado no que a variável de animação está linkado.

Exemplo de criação de animação

Exemplos

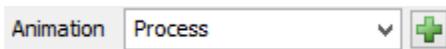
- Criando uma animação
- Keyframe Triggers
- Surrogates

Criando uma animação

Como um exemplo, nós iremos criar uma simples animação para um operador que será usado durante o processamento de um item em um equipamento (Processor).

Etapa 1

- Crie um operador
- Clique com o botão direito do mouse sobre um operador e selecione **Edit > Animations....** Isto irá abrir o Animation Creator.
- Crie uma nova animação ao clicar no botão  próximo ao Animation combobox. Renomeie a animação como "Process".



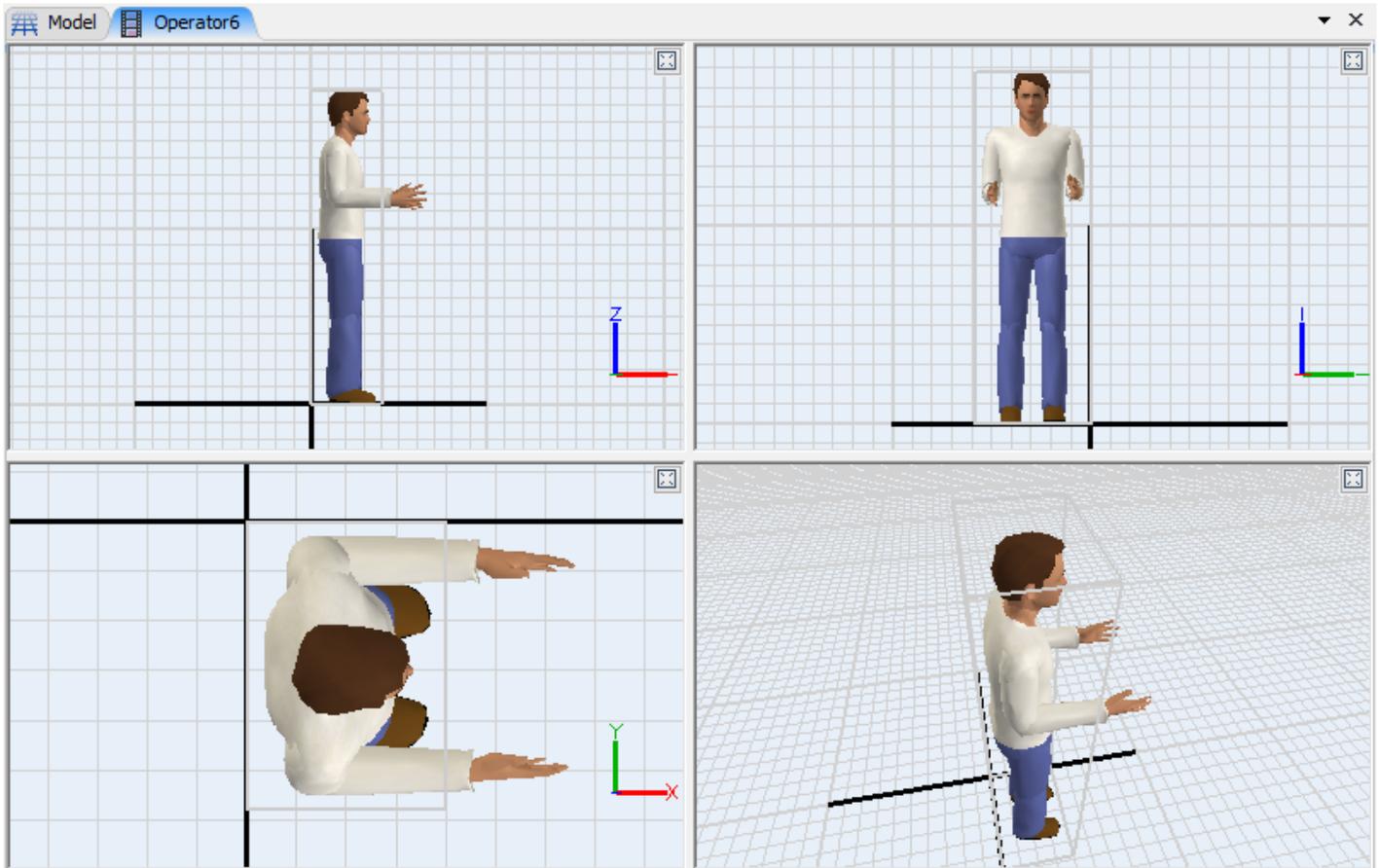
Você agora tem um "blank state" no qual você pode criar uma nova animação, customizada para suas necessidades. Para simplificar, você irá criar uma animação com somente 3 keyframes e alguns movimentos muito simples em seu braço para que você possa aprender as funcionalidades básicas do Animation Creator.

Você irá notar na janela Animations and Components que existe múltiplos componentes que compõe o operador (Pescoço, Peito, etc.). Componentes que estão dentro de outros componentes (indicado pelo deslocamento dos componentes para direita) estão sujeitos a alterações realizadas pelos seus componentes parentes (ou sub componentes). Por exemplo, clique em LArm, e então clique e arraste-o para qualquer lugar na superfície de simulação em 3D para move-lo. Ou altere os valores de X, Y, ou Z na janela Quick Properties. Conforme você movimentar o LArm, você verá o LForearm e o LHand se movimentar juntamente com o LArm. O mesmo é verdade para rotating e scaling, embora existam algumas diferentes opções para scaling que serão discutidos mais tarde.

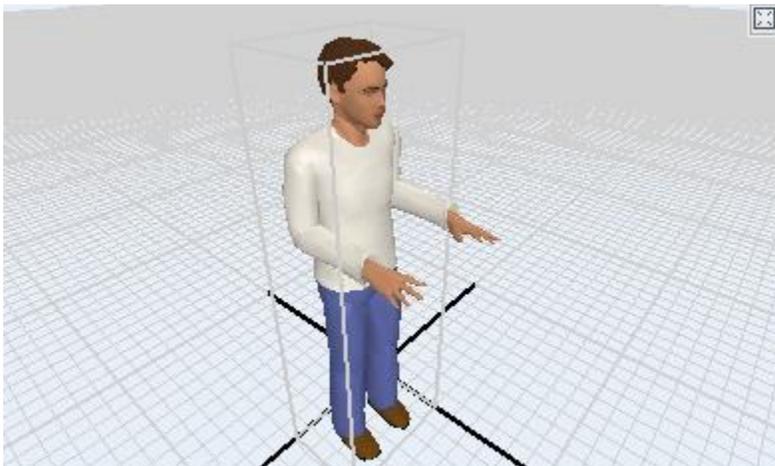
Quando realizamos mudanças para os objetos no Animation Creator, geralmente se iniciará da parte mais externa (componentes parente ou sub-componentes), e irá trabalhar ao longo de todos os componentes inferiores. Se o braço inteiro do operador irá alterar sua posição, começará com o braço, então com o antebraço, e então com as mãos. Antes de você seguir, se você perdeu como o operador estava se parecendo, clique em  Go To Base Positions para retornar a situação normal e padrão do operador.

Etapa 2

- Clique no botão  para assegurar-se que você está trabalhando no início do timeline.
- Clique em LForearm na listagem de componentes, e altere o RX () na janela Quick Properties para -90. Faça o mesmo para o RForearm.



- Clique em LHand na árvore (tree), e altere o RZ para 0. Mude o RHand RZ para -90. Ele agora terá ambos os braços estivados formando 90 graus com o ante braço, com as palmas da mão voltada para baixo. Esta será nossa posição inicial.



- Agora que nossas posições estão configuradas, clique no botão  para adicionar keyframe no tempo 0.

Etapa 3

Você irá notar que o cursor sobre o timeline pula para frente em 1 segundo quando apertamos o botão . Isto permite a você rapidamente realizar ajustes e adicionar keyframes tudo de uma vez sem sobrescrever qualquer mudança. Você podem então alterar o timing do keyframes ao clicar e arrastar eles ao longo do

timeline. Queremos garantir que a nossa animação é perfeita quando ela se repete. A maneira mais fácil de fazer isso é criar um outro keyframe, uma cópia do primeiro

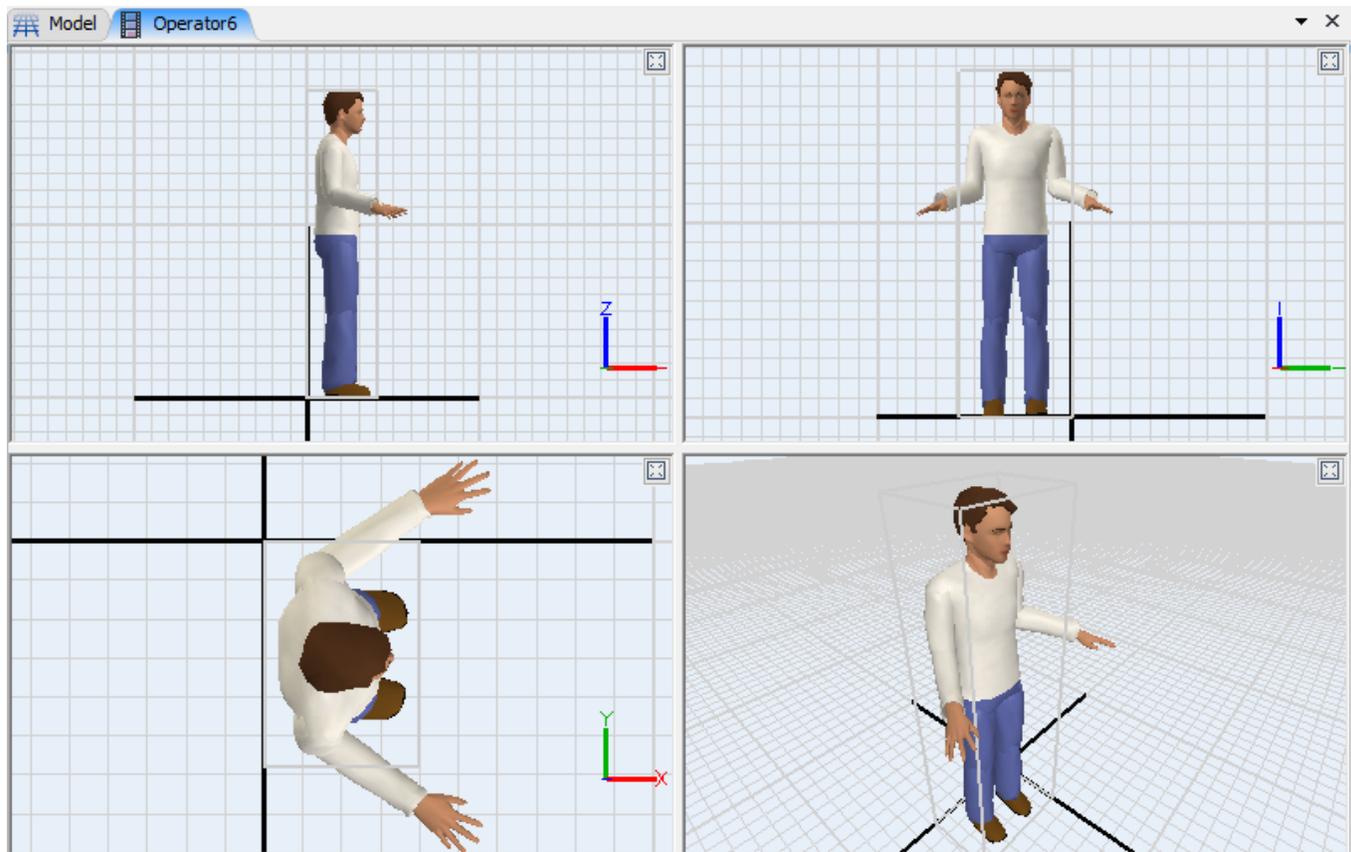
- Clique no botão  para adicionar um keyframe para o tempo 1.

	0	1
Triggers		
▶ Waist	●	●
▶ Chest	●	●
▶ Neck	●	●
▶ Head	●	●
▶ LArm	●	●
▶ LForearm	●	●
▶ LHand	●	●
▶ RArm	●	●
▶ RForearm	●	●
▶ RHand	●	●

Etapa 4

Nós iremos agora inserir um keyframe no meio de nossa animação.

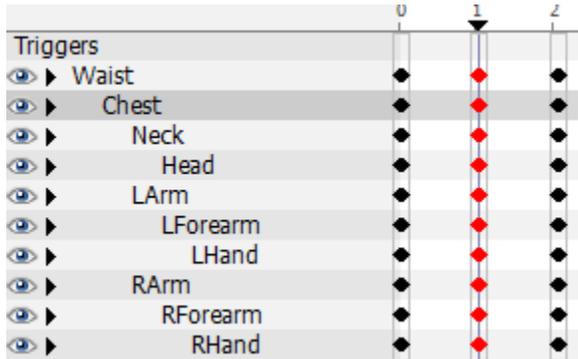
- Movimento o curso para algum lugar entre 0 e 1 ao clicar no timeline.
- Clique em LForearm, e altere o RZ para 35, e o RX para -75. Faça o mesmo para o RForearm, mas altere o RZ para -35



- Clique no botão  para adicionar um keyframe no timeline.

Agora que nós temos todas nossas 3 keyframes, nós queremos reduzir a animação de forma suave.

- Clique entre duas das keyframe verticais no tempo 1 (clikando sobre um keyframe irá somente seleccionar aquele keyframe). A coluna de keyframes irá se tornar vermelho.
- Arraste os keyframes do tempo 1 para o tempo 2.
- Clique e arraste o seu Segundo keyframe (tempo 0.5) e movimente o para o tempo 1.



Você pode agora visualizar a animação ao clicar no botão .

Etapa 5

Nós queremos assegurar-se que a animação se repita enquanto o operador estiver processando um item. Você pode alterar o Repeat type para a animação na janela Quick Properties sobre o painel Process.

- Configure o Repeat Type para **Repeat Indefinitely**.

Step 6

Agora que você criou uma animação, você pode ter o seu operador rodando esta animação quando ele processa um item em um processor. Tudo que você precisa fazer é chamar o seguinte código Flexscript:

```
startanimation(operator, "Process");
```

Para parar a animação chame:

```
stopanimation(operator, "Process");
```

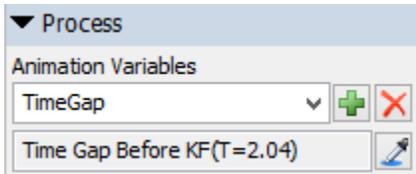
Keyframe Triggers

Neste exemplo nós iremos mostrar como criar uma animação dinâmica usando as Triggers Keyframe. Se você não fez isso ainda, complete o exemplo anterior sobre Creating An Animation antes de iniciar este exemplo. Na animação do operador, assegure-se que você está editando a animação para "Process".

Adicione uma variável de animação

Nós iremos adicionar uma variável de animação para nossa animação que aponta para o Time Gap entre o 2nd e o 3rd keyframes. Isto irá permitir a nós opcionalmente diminuir a animação baseado em um dado parâmetro no modelo.

- No pai  Process na janela Quick Properties, adicione uma variável de animação ao pressionar no botão .
- Configure o nome de uma nova variável para TimeGap.
- Clique sobre o  para entrar no modo "Sample". Aponte o ponteiro do "Sample" sobre o gap entre os keyframes 2 e 3 na timeline. O curso deve ler algo como "Time Gab Before KF(T=2.0)". Clique com o ponteiro do "Sample" para copiar este valor.



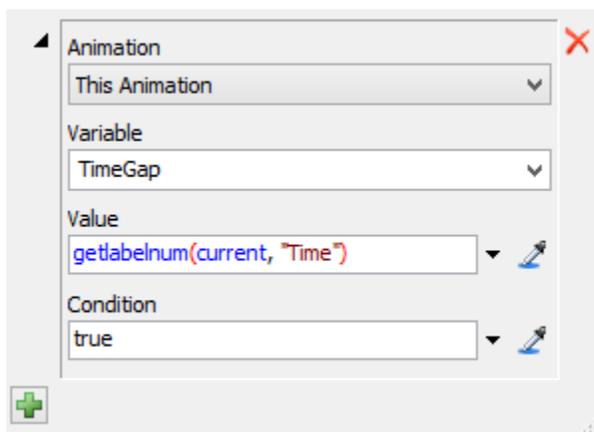
Adicione uma Trigger Keyframe

- Clique sobre a linha Triggers no timeline acima do 2nd keyframe.
- Clique sobre o botão  para adicionar uma trigger keyframe.



Um novo painel irá aparecer na janela Quick Properties chamada **Keyframe Trigger**.

- Clique sobre o botão  e selecione **Set Animation Variable**.
- Entre com os seguintes valores no popup:



Adicione uma Time Label

A última etapa é adicionar uma label para o seu operador chamada "Time". Você pode então alterar este valor durante a execução de seu modelo e a animação irá dinamicamente alterar a distância entre o keyframe 2 e 3, efetivamente reduzindo ou aumentando a velocidade daquela seção na animação.

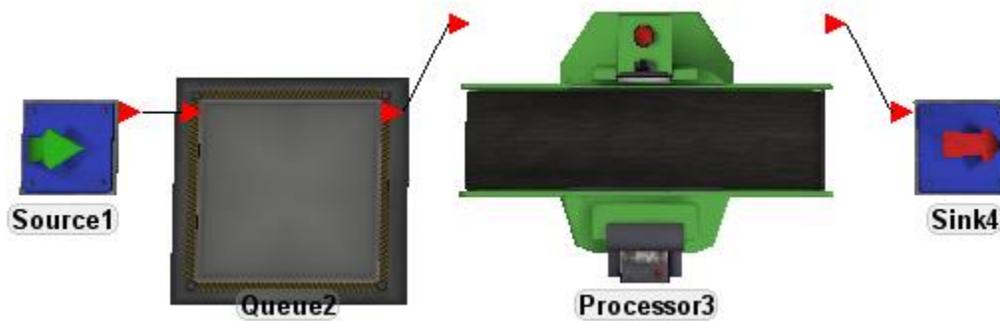
Este mesmo método pode ser usado para alterar posição, tamanho, rotação, cor e shape frames dos componentes.

Surrogates

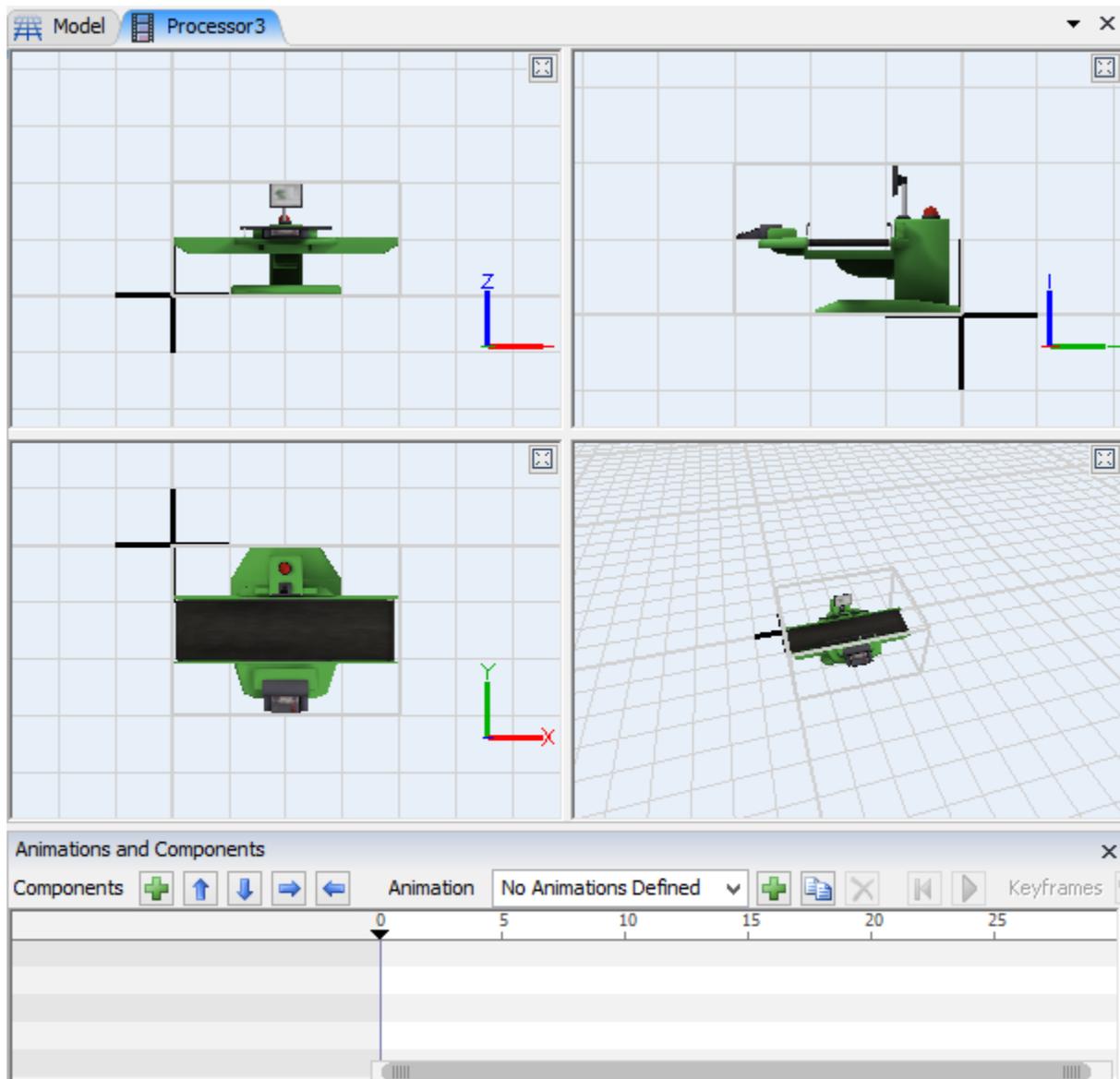
Digamos que você quer customizar o que os flowitem (entidades) fazem quando eles entram em um Processor ao invés de somente movimentar ao longo dele. A forma mais fácil de fazer isso é criar uma animação customizada sobre um Processor usando um Surrogate.

Crie o Modelo

Crie o seguinte modelo simples:



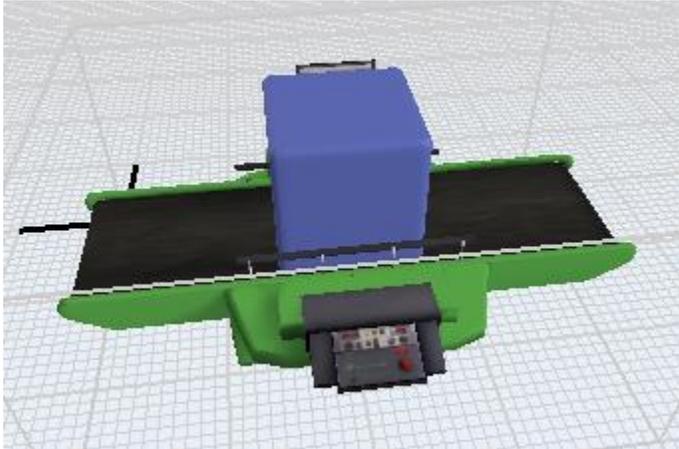
Clique com o botão direito do mouse sobre o processor e acesse **Edit > Animations...** Isto irá abrir o Animation Creator



Na janela Animations and Components, pressione o botão **+** para adicionar um novo componente. Na janela Quick Properties configure o nome do objeto para *MyItem*.



Na vista em 3D, posicione o item para que fique no centro do Processor.

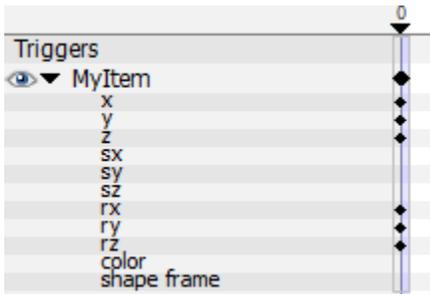


Nós agora necessitamos adicionar uma animação para o Processor. Na janela Animations e Componentes, pressione o botão **+** próximo ao Animation combobox. Renomeie a animação como “Process”.



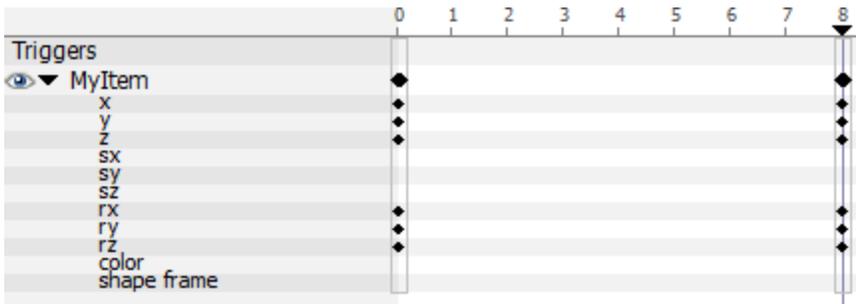
Criando Keyframes

Nós iremos criar duas keyframes no component MyItem. Visto que o componente é onde nós queremos que animação comece, nós agora podemos ir adiante e criar uma keyframe. Pressione o botão **+** para adicionar um keyframe para MyItem. Se você abrir as propriedades da lista de componentes, você verá estes adds keyframes para a posição e rotação do objeto. Em nosso caso, nós iremos apenas modificar a rotação z, no entanto, não irá causar modificações significantes tendo estes extra keyframes.



Mova o cursor do time na timeline sobre o número 8 clicando sobre o 8 ou qualquer lugar abaixo dele. Agora nós podemos reposicionar o MyItem. No nosso caso, queremos rodar os 360 graus do componente sobre o eixo z. Você pode girá-lo na superfície de simulação em 3D com o botão direito na seta azul e arrastando para cima ou para baixo, ou você pode digitar manualmente 360 no campo de rotação z da janela Quick Properties.

Agora clique no botão **+** para adicionar um novo keyframe para MyItem com o cursor sobre a posição 8. Selecione a linha z e então vá até a janela “Quick Properties” e digite o valor de 360 no campo de rotação z.



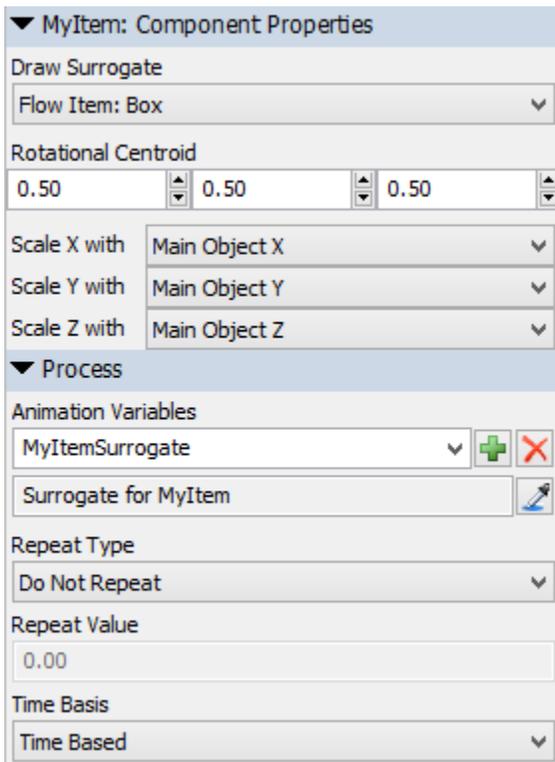
Pressione  para rodar sua animação. Você deve ver MyItem continuamente girando sobre o eixo z.

Criando um Surrogate

Tendo um cubo girando sobre nosso Processor é ótimo, mas isso não se parece o mesmo que o nosso flowitem que irá ser processado neste Processor. Para isso, nós teremos o MyItem desenhando um surrogate.

Se o MyItem não está selecionado, clique sobre ele na superfície de simulação em 3D ou na janela Animations and Components. Na janela Quick Properties, você verá um painel nomeado como *MyItem: Component Properties*. Configure o **Draw Surrogate** para o Flowitem: Box.

Clique no botão Add para linkar o surrogate para uma variável de animação (Animation Variable) chamada "MyItemSurrogate".



Rode a Animação

Volte para o seu modelo. Você notará que o box não aparece. Quando nós referenciamos ele para desenhar um surrogate, ele configura o componente para ser escondido até que nós o referenciamos como um objeto válido em 3D.

Nas páginas de propriedade do Processor, e sob a guia Trigger, pressione o botão  próximo a trigger **OnEntry** para abrir o editor de código. Entre com o seguinte código:

```
setanimationvar(current, "Process", "MyItemSurrogate", item);
```

```
startanimation(current, "Process");
```

Edite a trigger OnExit com o seguinte código:

```
stopanimation(current, "Process");
```

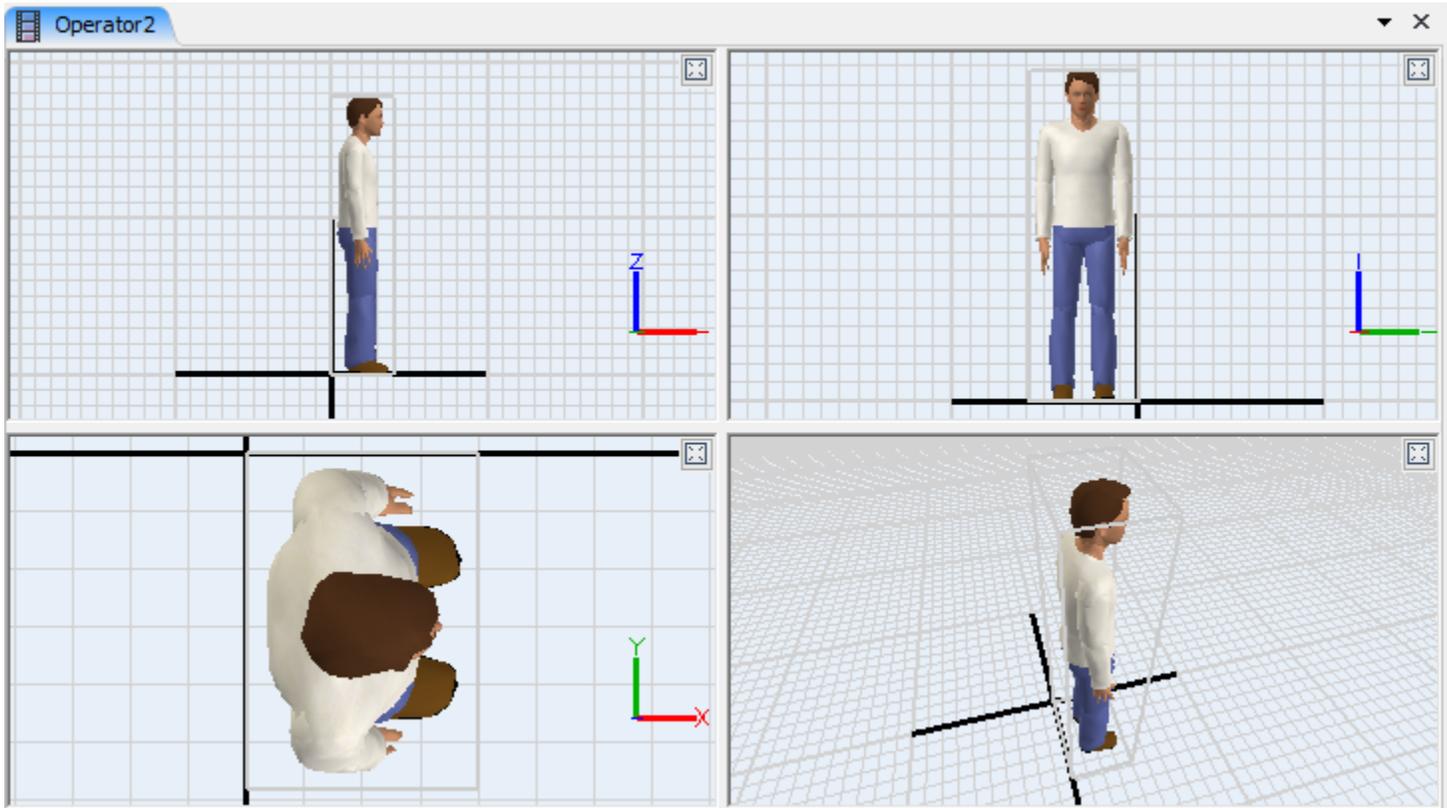
Por último, queremos ocultar o item que está no processor enquanto o nosso surrogate estará atuando como o item. Na guia **general** do Processor desmarque o botão **Show Contents**

Agora Reset e Execute o modelo para ver o que a sua animação se parece.

Atualmente estamos usando a classe Flowitem (Flowitem class) padrão do source, o Box. No entanto, você pode alterar o source para qualquer flowitems com qualquer formato 3D e a novo forma será exibida no processor, girando.

Animation Creator Reference

Animation Editor



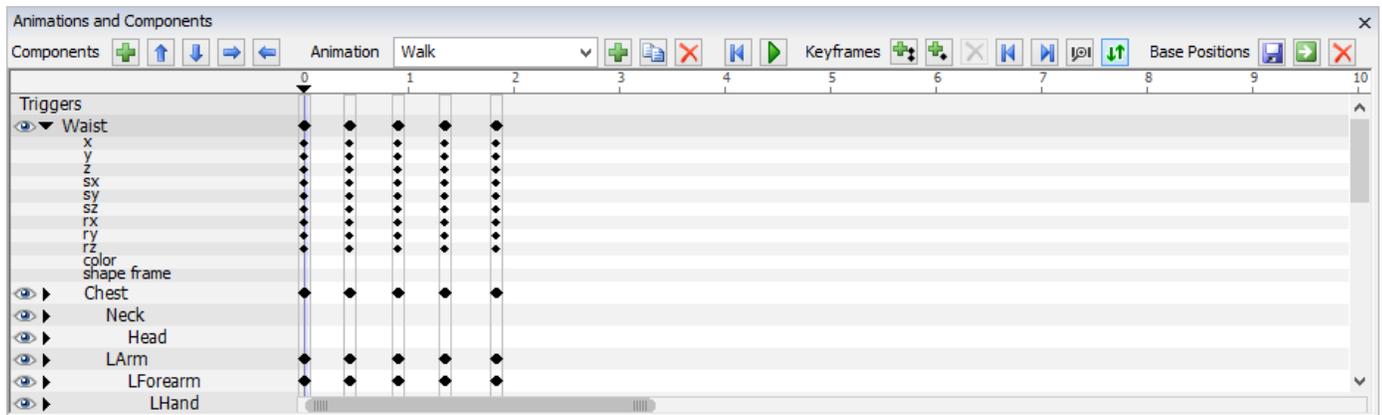
Clique sobre um componente em um das janelas de visualização em 3D para destacar aquele componente nas demais vistas em 3D e também na timeline Animations and Components.

A vista em 3D funciona como a vista em 3D padrão do software. Você pode movimentar-se na superfície de simulação em 3D (embora a rotação seja permitida somente na vista em perspectiva) e posicionar, aumentar e rotacionar componentes.

 - Esconde as outras janelas de visualização 3D e expande a janela selecionada 3D para usar totalmente a vista do Animation Editor.

Flip Axis - Disponível no lado, frente e parte superior da vista em 3D através do clique com o botão direito do mouse. Esta opção inverte o eixo do ponto de vista. Por exemplo, se você virar o eixo que está atualmente exibindo a vista superior, irá agora mostrar a vista inferior.

Janela Animations and Components



Components

- Selecione um componente da lista e clique em delete para remover o componente.
- Remove o component selecionado para cima ou para baixa na lista.
- Remove o componente selecionado para esquerda ou direita da lista. Isto cria relação de componentes parentes/filho(a)s com o componente superior. Quando a posição ou rotação de um componente parente (sub-componente), todos os demais sub-componentes irão ter suas posições/rotações movimentadas respeitando o que aconteceu com seu componente superior.

Animation

Animation combobox - Este campo mostra a animação atual sendo editada assim como um lista de todas as animações para este objeto. Selecione uma animação para mostrar a animação no timeline.

- Adicione uma nova animação para o objeto.
- Duplica a animação atual.
- Remove a animação atual do objeto.
- Move o cursor de tempo atual para o início da animação (tempo 0)
- Inicia/Para a animação. Animações irão automaticamente fazer vários loops e repetir as animações desde o início.

Keyframes

- Adiciona um novo keyframe para a animação de todos os componentes.
- Adiciona um novo keyframe para a animação para o componente selecionado somente.
- Remove o keyframe(s) selecionado.
- Move o curso do tempo para a keyframe anterior ou posterior.
- Aumenta ou diminui o zoom do timeline para fazer com que todos os keyframes encaixem na tela.
- Sincroniza a vista em 3D. Se alternado, a vista em 3D irá atualizar o objeto para mostrar os valores atuais de todos os seus componentes conforme você movimenta o cursor no timeline.

Base Positions

- Salva os valores atuais de todos os componentes do objeto como posição base. Isto é como o objeto será mostrado no modelo quando o modelo for resetado.
- Atualiza a vista em 3D para mostrar o objeto em sua posição base atual salva. Isto não afeta o timeline.

✗ - Remove as posições base salvas.

Timeline

Clique sobre um keyframe para selecioná-lo (ficará em vermelho). Você pode arrastar os keyframes ao redor do timeline para reposicioná-los. Você pode aumentar ou diminuir o zoom ao arrastar para o final a barra da parte inferior. Você pode alterar o curso do tempo atual ao clicar e/ou arrastar para qualquer lugar no timeline.

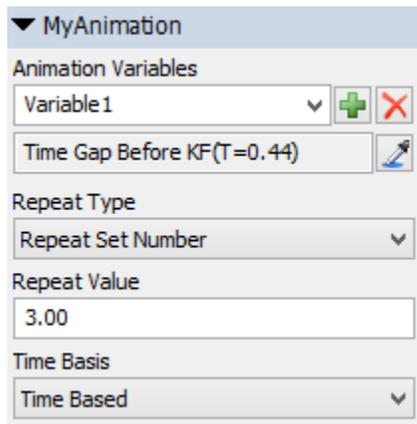
Selecione um component para destacá-lo na vista em 3D do Animation Editor.

Triggers - Mostra as triggers keyframe para a animação.

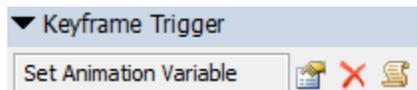
👁 - Alterna os componentes entre visível ou escondido.

▾ - Abre as propriedades individuais daquele componente, mostrando os keyframes para a posição, tamanho, rotação etc...

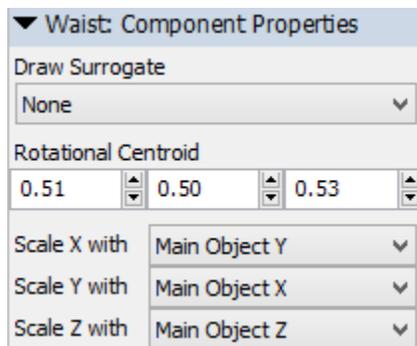
Quick Properties



Veja o Concepts page para mais informações sobre Repeat Type e Time Basis.



Para mais informações sobre a Triggers Keyframe, veja o Concepts page.

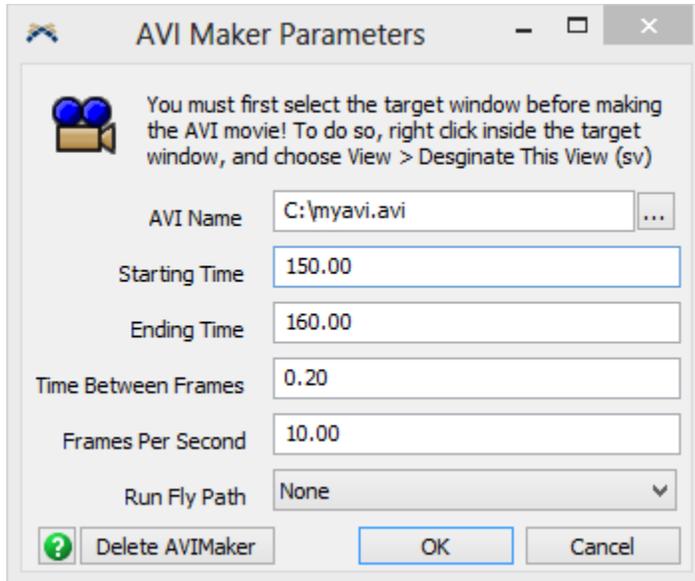


Para mais informações sobre Surrogates, veja o Concepts page.

AVI Maker

1. Conceitos
2. Exemplo

AVI Maker Conceitos

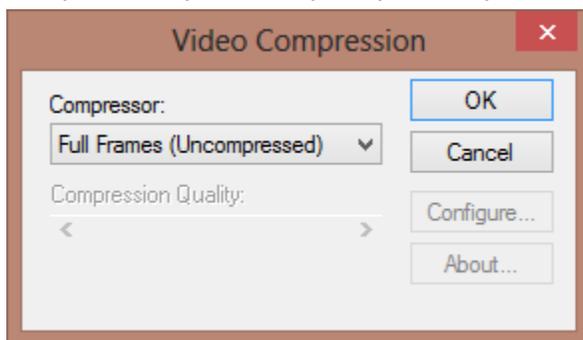


O Presentation Builder é acessado do menu Tools > **Presentation** > **AVI Maker**.

O AVIMaker é um objeto que é adicionado no campo Tools da árvore do modelo (tree) quando a opção AVI Maker é selecionada no menu Tools. É um objeto especial no modelo que chama o comando para fazer um arquivo AVI do modelo enquanto ele é executado. Irá criar um arquivo AVI enquanto o objeto AVIMaker estiver no modelo. Se o usuário não quiser fazer o arquivo AVI, eles necessitam deletar o objeto do modelo. Antes de executar o modelo, o usuário deve designar ou referenciar os pontos de visualizações que serão gravados. Isto é feito dando um duplo clique sobre a vista em 3D desejada, ao selecionar **View > Designate This View (sv)**. O modelo poderá ser executado bem devagar enquanto o arquivo AVI estiver sendo gravado. Ele não irá responder a barra de velocidade do controle de execução durante este momento.

Nota sobre o tamanho da tela: O AVI Maker irá capturar toda a janela 3D conforme salva cada frame. Quanto maior a janela, maior o tamanho do arquivo de vídeo e mais tempo irá levar para exportar o arquivo .avi.

Uma vez pressionado o botão  Run, uma janela irá aparecer permitindo a você especificar o tipo de compressão que você quer que o arquivo .avi use.



Selecione a compressão desejada e então pressione OK para fechar a janela.

AVI Name - Este é o nome e o caminho do arquivo que o AVIMaker irá adotar. Deve ter uma extensão .avi.

Starting Time - Este é o tempo de simulação que o AVIMaker deve iniciar a gravar o arquivo AVI.

Ending Time - Este é o tempo de simulação que o AVIMaker irá parar de gravar o arquivo AVI. Recomenda-se que o modelo não ser interrompido antes deste tempo, pois isso pode corromper o arquivo que está sendo gravado.

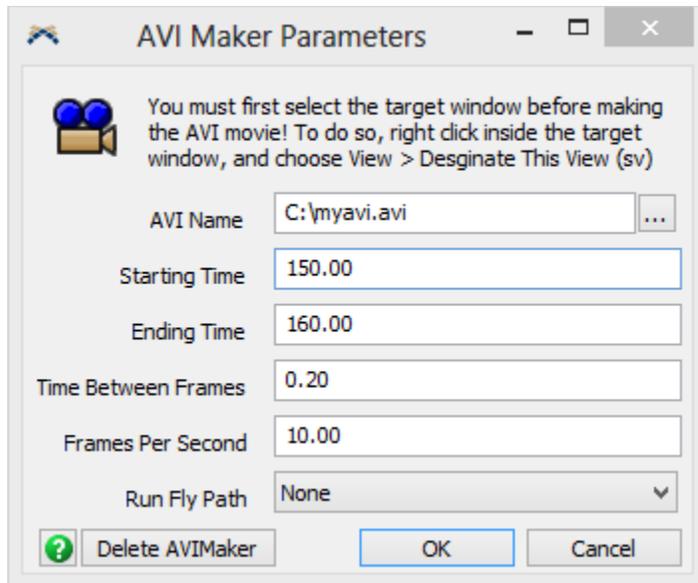
Time Between Frames - Esta é a forma como o tempo de simulação passa no modelo entre os frames gravados.

Frames Per Second - Esse número define quantos frames por segundo o arquivo AVI reproduz. Este valor é baseado em segundos reais não em tempo de simulação.

Run Fly Path - Se você tiver criado um fly path em seu modelo, você pode selecionar esse fly path nesta caixa suspensa (drop down). A visão vai seguir esse fly path conforme o modelo é executado, dando mais realidade para o arquivo avi gravado.

Delete AVIMaker - Quando este botão é pressionado, o AVIMaker é excluído do modelo. O modelo será executado em velocidade normal novamente, e o arquivo AVI não será gerado.

Exemplo do AVI Maker



Trabalhando com o AVI Maker

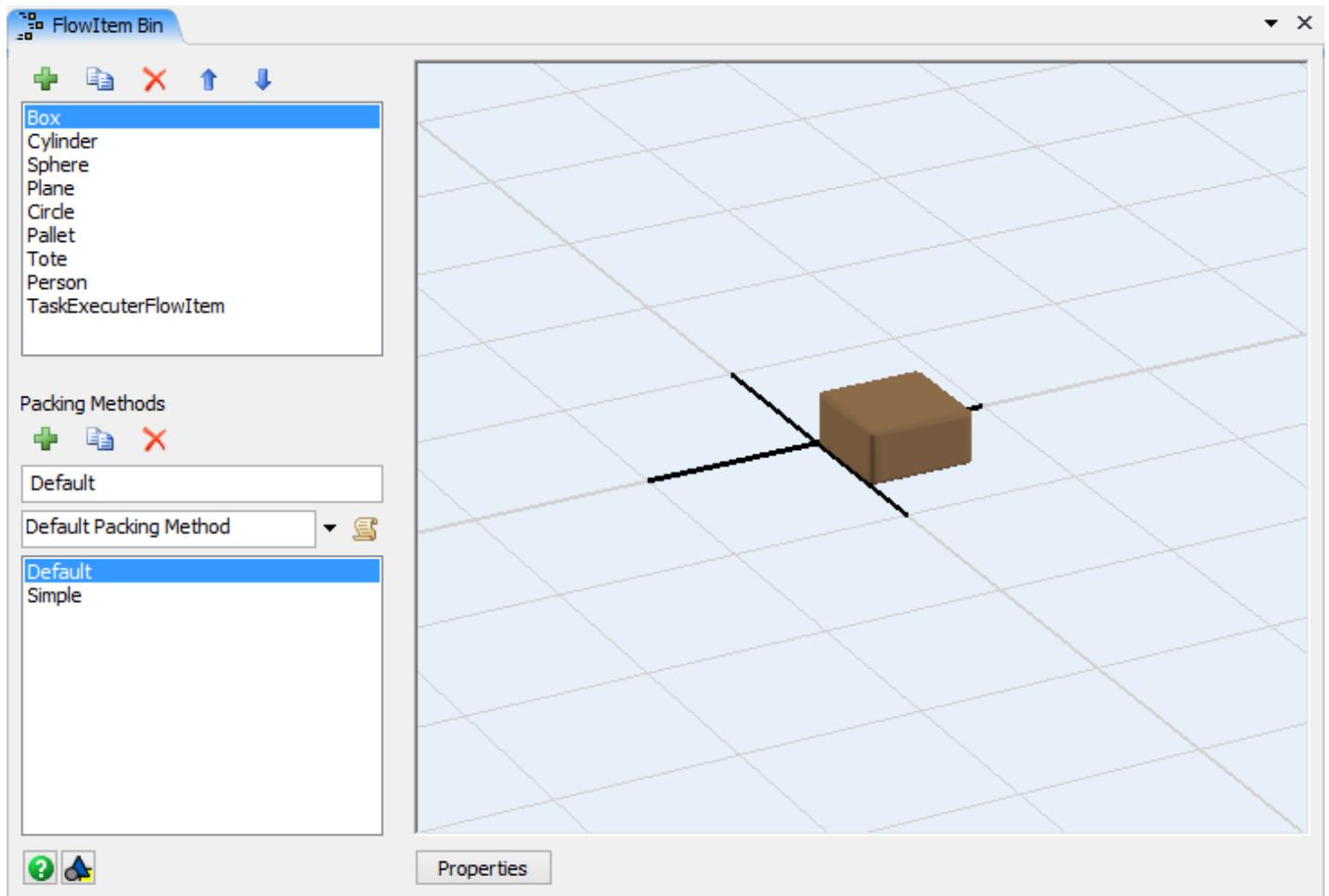
A produção de arquivos AVI muitas vezes pode ser complicado para trabalhar. Aqui estão os passos a passos a percorrer para se certificar de que o seu AVI é criado com o mínimo de problemas possível

1. Abra a janela do AVI Maker (adicionando um objeto AVIMaker para o seu modelo) indo no menu Tools menu > Presentation > AVI Maker.
2. Preencha os campos mencionados acima apropriadamente.
 - a. Assegure-se se o nome do arquivo avi não é o nome do arquivo que já existe, para não sobrescrever o arquivo.
 - b. Configure o início e tempo de parada de acordo com o tempo de simulação que você quer o arquivo avi seja gravado.
 - c. Configure o valor de frames por segundos para qualquer valor que você quer para a velocidade de seu playbck avi.
 - d. Configure o valor do tempo entre frames baseado no valor de frames por segundo que você especificou. Encontre uma velocidade ideal de execução que você quer que o avi seja gravado para o modelo (do painel de controle de execução do modelo). O tempo entre os frames deve ser calculado como a velocidade ideal de execução do modelo dividido pelo frames por segundos.
3. Clique com o botão direito do mouse na vista em 3D que você quer que seja gravado e selecione View > Designate This View (sv).
4. Redimensione a vista em 3D para a resolução que você quer gerar o filme .avi. Configurar uma janela de tamanho menor pode drasticamente aumentar a velocidade com que o avi maker pode criar os filmes.
5. Resete e execute o modelo.
6. Uma janela pop up sobre o codec a ser usado irá aparecer. Entre com o codec/compression que você quer usar.
7. **IMPORTANTE:** Aguarde até o avi maker estiver finalizado. Uma vez que o modelo atinge o tempo de início do avi, não pressione qualquer botão ou clique sobre alguma coisa até que o tempo do modelo estiver passado do tempo final que você especificou para o avi maker.
8. Uma vez que o modelo passou o tempo de término da criação do arquivo avi, pare o modelo. Não clique em resete novamente até que você tenha deletado o AVI Maker.
9. Delete o avi maker usando o botão Delete AVI Maker.

Flowitem Bin

1. Conceitos
2. Referencia

Flowitem Bin Conceitos



Você pode acessar o Flowitem bin através do menu Tools, ou através da barra de ferramentas do FlexSim. O Flowitem bin armazena todas as entidades (flowitems) usadas em seu modelo. Você pode aprender mais sobre o Flowitems no FlexSim Concepts – página dos Flowitems. Diferentes classes de flowitems são criadas nesta janela e são armazenadas no Flowitem Bin.

Flowitems que são criadas no modelo são cópias exatas de Flowitems no Flowitem Bin. O objeto source especifica sempre qual Flowitem será criado. Ele cria uma cópia do Flowitem e os coloca dentro do modelo.

Tipos de Flowitem

Existem três tipos de flowitems, Basic, Container e Task Executer. Quando um novo modelo é criado, um conjunto padrão de Flowitems é adicionado ao Flowitem bin. A caixa, cilindro, esfera, círculo e pessoa são todos Basic Flowitems. O pallet e o tote são container flowitems e o TaskExecuterFlowItem é um flowitem executor de tarefas.

Basic Flowitems

Básico Flowitems tem as seguintes propriedades:

Name - Cada Flowitem tem seu próprio nome. No entanto, ao contrário dos objetos FixedResource e TaskExecuter, os nomes dos Flowitem não precisam ser único dentro do modelo. Se o Flowitem é nomeado como “Box” no Flowitem Bin, então todas as cópias daquele Flowitem serão nomeados “Box” a menos que o nome esteja explicitamente alterado.

Itemtype – Veja FlexSim Concepts - Itemtype page para mais informações sobre Itemtypes.

Labels – Veja FlexSim Concepts - Labels page para mais informações sobre Labels.

Location, Size, Rotation - Flowitems ocupam espaço físico no seu modelo. Isso permite que você defina suas Flowitems para ser do mesmo tamanho que as peças que você está modelando (ou seja, garrafas em uma linha de engarrafamento).

Color - Tal como acontece com outros objetos no Flexsim, Flowitems tem uma cor designada. Essa cor pode ser alterada no Flowitem Bin, mas também pode ser alterado no modelo conforme o Flowitem faz o seu caminho através de processos. Muitas vezes, as cores são usadas para representar tipos de produtos específicos e permite que o modelador siga mais facilmente o que está acontecendo em um modelo.

3D Shape - Veja a seção de media 3D para mais informações sobre as forma em 3D.

Shape Frames - Tal como acontece com outros objetos, Flowitems podem ter Shape Frames. Isso é particularmente útil com Flowitems conforme ele pode ajudar a mostrar como uma peça muda conforme ele se move através do modelo. Por exemplo, em uma linha de engarrafamento, o Flowitem pode começar como uma peça de plástico ou de vidro, em seguida tornar-se um frasco, em seguida, ele é cheio, e uma tampa é colocada em cima. Isso pode ser demonstrado através da criação de cinco formas 3D diferentes e mudando o Shape frame.

Animations – Novo no FlexSim 7, Flowitems pode ter sua própria animação customizada usando o Animation Creator.

Container Flowitems

Container Flowitems tem todas as mesmas funcionalidades que o Flowitems Basic. Apesar de todos os Flowitems poderem agir como recipientes, apenas o Container Flowitem executa o código quando outro item é colocado dentro dele (embalado). Você pode criar métodos de embalagem personalizados na seção de métodos de embalagem do Flowitem Bin.

Container Flowitems tem uma página extra em sua janela de propriedades que permite que você especifique que método de embalagem usar.

TaskExecuter Flowitems

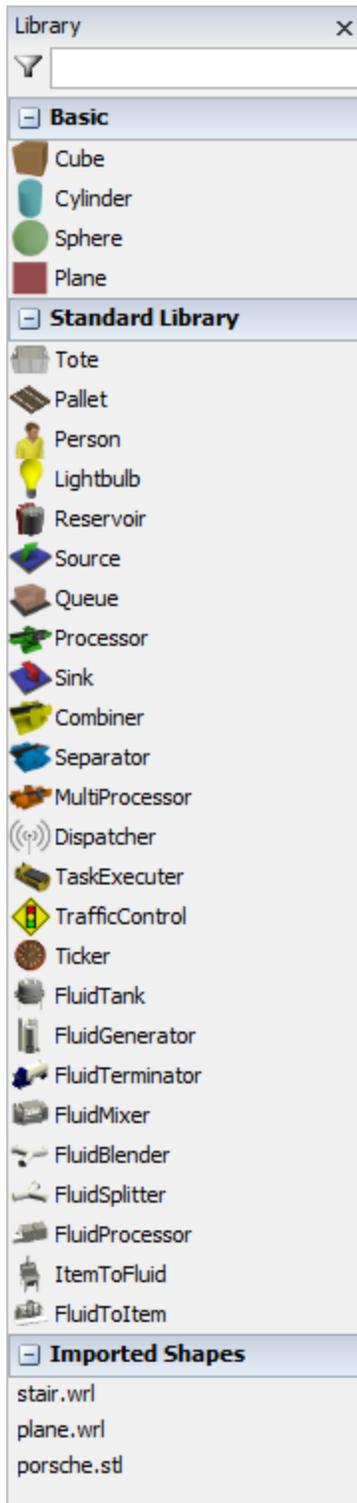
TaskExecuter Flowitems tem todas as mesmas funcionalidades que os Basic Flowitems. Elas também têm a capacidade de agir como um objeto TaskExecuter padrão. Ao contrário de outros Flowitems, eles podem ter suas próprias conexões que podem ser adicionados dinamicamente usando o comando contextdragconnection.

TaskExecuter Flowitems têm as seguintes páginas adicionado à sua janela de propriedades:

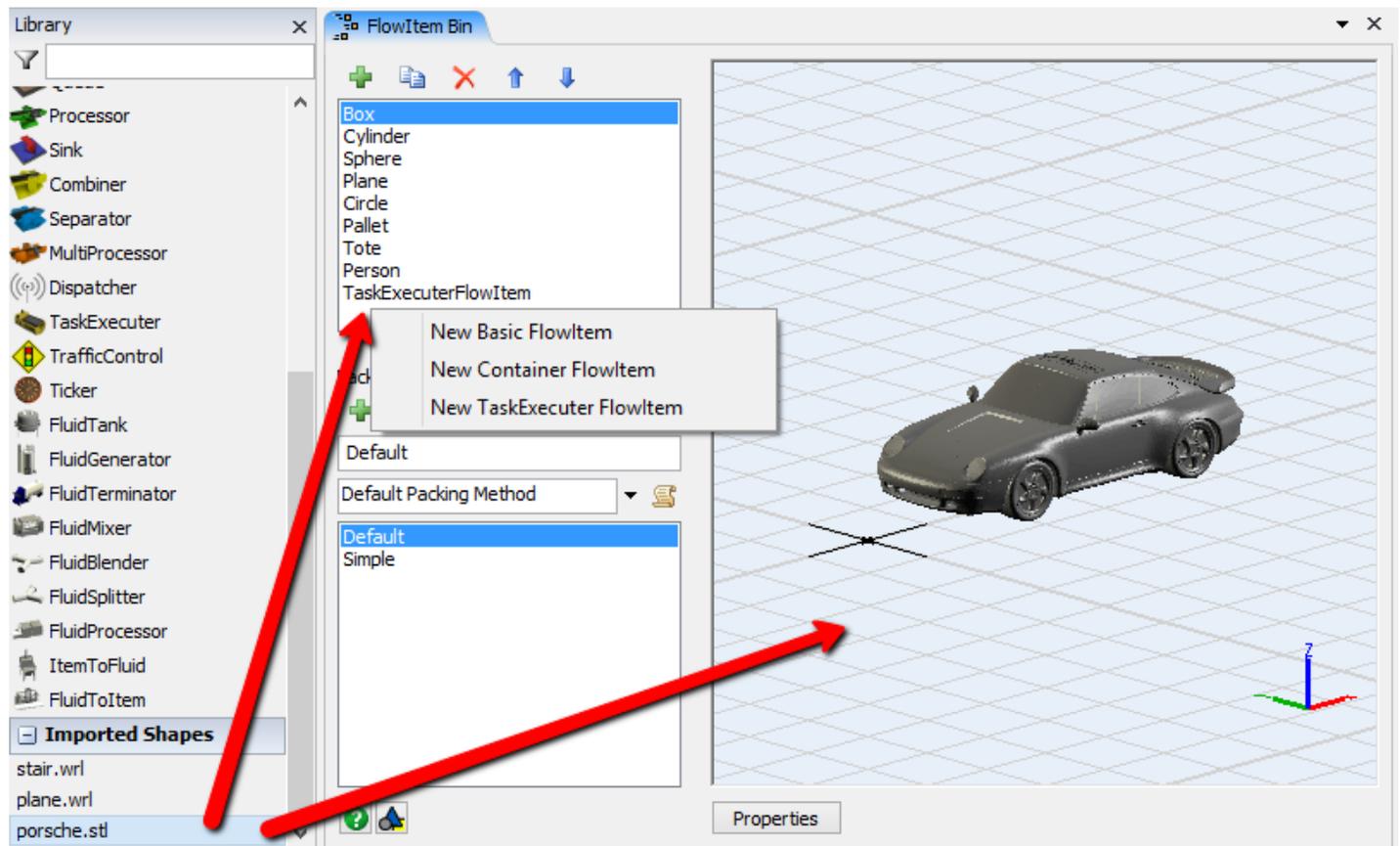
- TaskExecuter
- Breaks
- Collision
- Triggers
- Statistics

Grade de Biblioteca de Objetos

Quando o Flowitem Bin é ativo, a grade de biblioteca de objetos se altera para mostrar uma nova lista de formas:



Shapes que foram importados para o seu modelo será exibido na parte inferior da Biblioteca. Você pode arrastar e soltar as formas a partir da Biblioteca para a visualização em 3D para mudar rapidamente a forma 3D do Flowitem, ou você pode arrastar e soltar na lista Flowitem Class para criar um novo Flowitem:

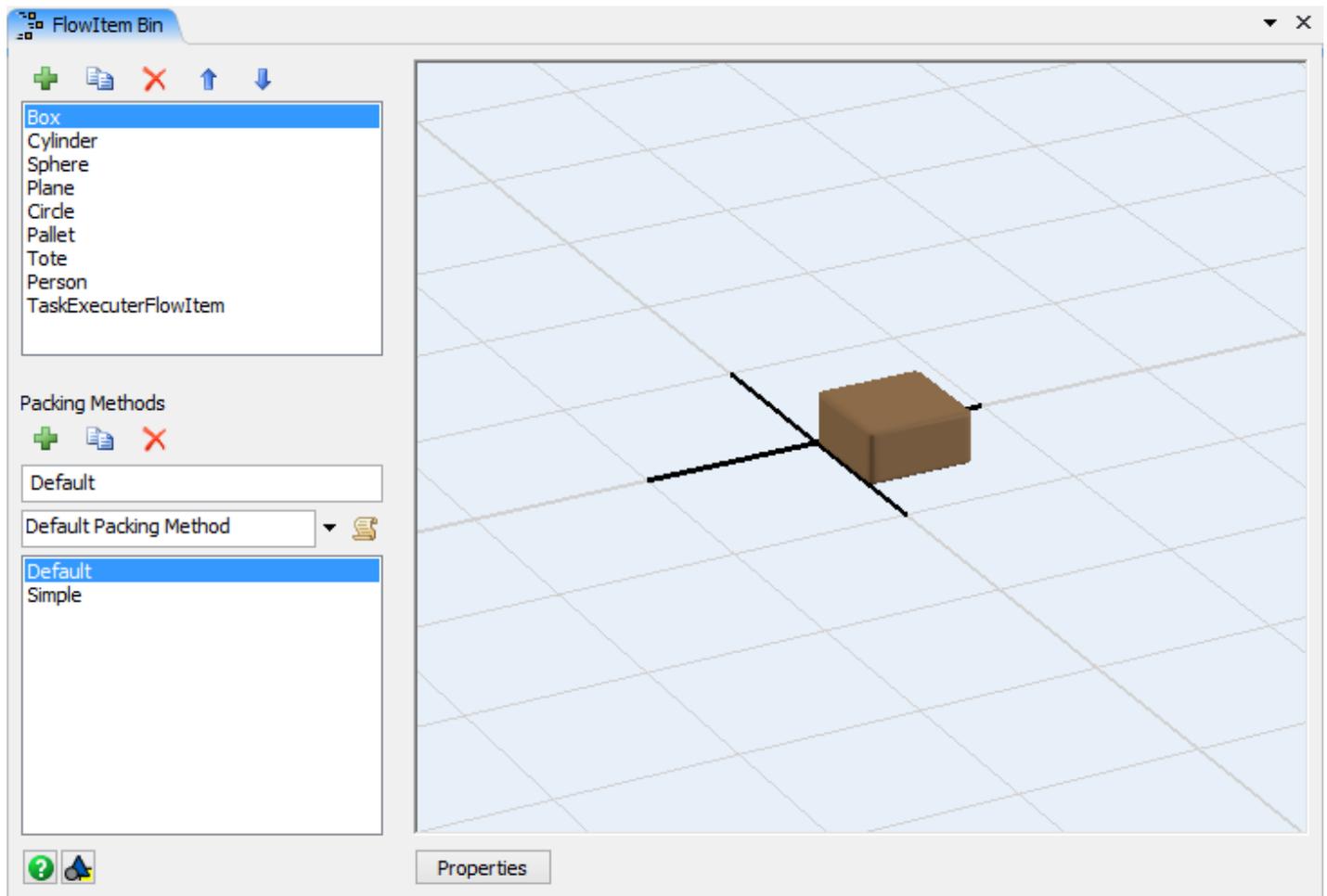


Você pode filtrar a listagem de formas ao entrar com o nome no campo 

Flowitem Bin Referencia

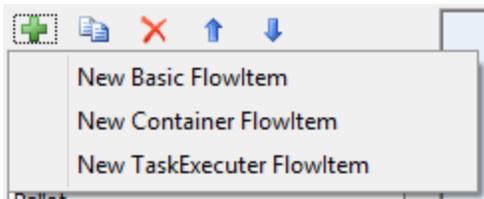
Tópicos

- Flowitem Classes
- Packing Methods
- Right-Click Menu
- Quick Properties



Flowitem Classes

+ - Adiciona um novo Flowitem para ambos Basic, Container ou TaskExecuter tipo de Flowitem Bin.



D - Duplica o Flowitem atualmente selecionado.

X - Remove o Flowitem atualmente selecionado do Flowitem Bin.

↑ ↓ - Reordena o Flowitem atualmente selecionado para cima ou para baixo na lista.

Flowitem Class List - Esta lista contém todas as Flowitem Classes disponíveis. Quando um é selecionado, é mostrado na vista em 3D a direita. A janela Quick Properties irá também atualizar para mostrar as propriedades para aquele Flowitem class.

Packing Methods

 - Adiciona um novo método de embalagem (Packing Method)

 - Duplica o método de Packing atual.

 - Remove o método de Packing atual.

Packing Method Name - Altera o nome do método de embalagem (Packing Method) ao entrar com o novo nome neste campo.

Packing Method Picklist - Permite a você pegar da listagem de métodos de embalagem pré-configurados ou para editar o código manualmente na janela de edição do código.

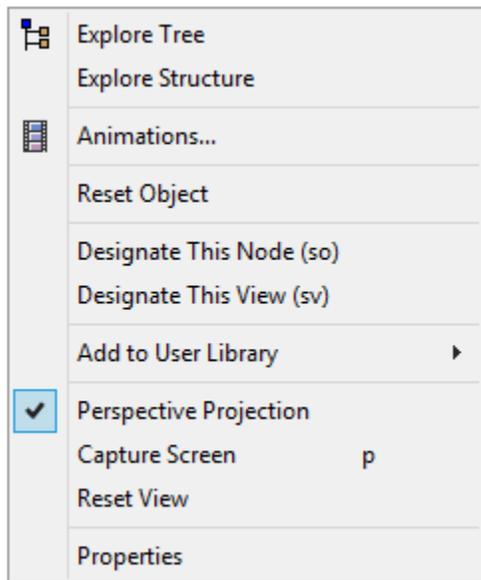
Packing Methods List - Mostra uma listagem de todos os métodos de embalagem.

 - Mostra a página de ajuda.

 - Adiciona o Flowitem selecionado para a biblioteca de usuário (User Library) como um ícone que pode ser arrastado ou um componente auto instalado.

Properties - Este botão abre a janela atualmente selecionada de propriedades do Flowitem. Você também pode acessar esta janela dando um duplo clique sobre o Flowitem na vista em 3D.

Right-Click Menu



Explore Tree - Abre a janela Tree e mostra o objeto Flowitem na árvore (Tree).

Explore Structure - Esta opção abre a janela do Tree para explorar a estrutura do modelo em 3D diretamente no Tree.

Animations... - Abre o Animation Creator para editar a animação dos objetos.

Reset Object - Isto reseta as rotações x/y/z e a localização z do objeto para 0.

Designate This Node (so) - Isto designa o objeto como "Selected object" o qual pode ser referenciado em código como so(). Você usará esta opção para escrever código no console script. Lá pode ter apenas um so() por vez.

Designate This View (sv) - Isto designa o objeto como “Selected object” o qual pode ser referenciado em código como sv(). Você usará esta opção para escrever código no console script. Lá pode ter apenas um sv() por vez.

Add To User Library - Adicione o Flowitem para uma biblioteca de usuário (User Library) para ser um ícone que pode ser arrastado ou um componente auto instalável.

Perspective Projection - Alterna a vista em 3D de Ortográfica para Perspectiva. Para mais informações veja a página de vista em perspectiva.

Capture Screen - Salva um arquivo PNG da vista atual em 3D.

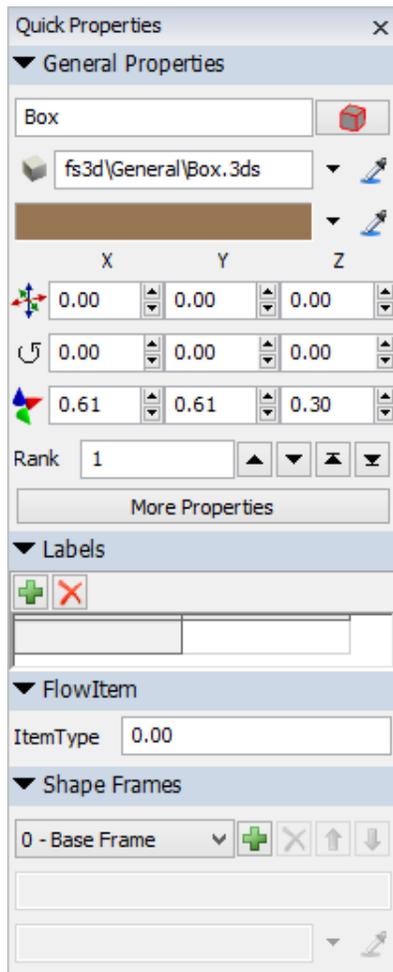
Reset View - Reseta a rotação e posição da vista para 0.

Properties - Abre a janela de propriedades dos Flowitems.

3D View - Mostra o Flowitem em 3D.

Quick Properties

A janela Quick Properties irá se alterar para mostrar o seguinte painel:



As mesmas propriedades pode também ser alterada ao abrir a janela de propriedades dos Flowitems.

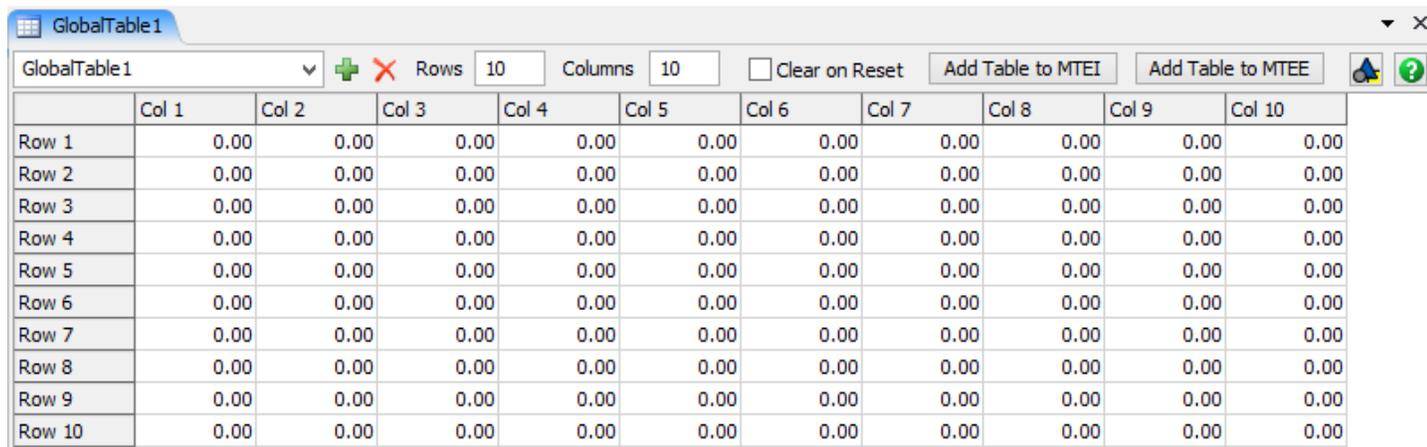
General Properties - Veja a página General para mais informações.

Labels - Adiciona, remove e edita as labels para este Flowitem.

Flowitem - Especifica o tipo de item (itemtype) do Flowitem. Para mais informações veja o FlexSim Concepts.

Shape Frames – Veja a página Shape Frames para mais informações.

Global Tables



	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10
Row 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Global Tables são acessadas do menu Tools > **Global Tables**. Estes objetos não são arrastáveis para dentro do modelo. Eles são criados através do menu Tools ou através da janela Global Table.

Global Tables podem armazenar dados numéricos ou strings. Estes dados podem ser acessados por qualquer objeto no modelo usando vários comandos de table. Um modelo pode ter qualquer número de Global Table.

Editando a Tabela

Para editar uma célula em uma tabela, clique na célula desejada e digite ou sob escreva todos os dados na célula, ou dê um duplo clique sobre a célula para selecionar o conteúdo da célula. Use o teclado de setas para navegar entre as células como no Excel. As células mantêm dados numéricos por padrão, mas podem ser configuradas para manter dados string, ao clicar com o botão direito do mouse sobre a célula e selecionar Assign String Data. O clique com o botão direito do mouse também possibilita inserir/deletar linhas e colunas, limpando os dados da célula, e escolher por coluna.

Name Combobox - Esta é o nome da tabela e tem uma lista de todas as Global Table do modelo. O nome deve ser memorizado e descreve a função da tabela. Os comandos para ler a tabela são acessados pelo nome dela. Você pode visualizar outras Global Tables nesta janela ao clicar na seta drop down próximo ao nome.

 - Adiciona uma Global Table para o seu modelo.

 - Remove a Global Table atual do seu modelo.

Rows - Este é o número de linhas na tabela.

Columns - Este é o número de colunas na tabela.

Clear on Reset - Se esta caixa estiver marcada, todas as células de número na tabela serão configurados como 0 e as células com dados string serão zeradas quando o modelo é resetado.

Add Table to MTEI - Este botão adiciona uma tabela como uma linha na Multiple Table Excel Import

Add Table to MTEE - Este botão adiciona uma tabela como um linha na Multiple Table Excel Export

 Este botão permite a você adicionar esta tabela para biblioteca de usuário como um ícone arrastável ou como um componente para instalação automática.

Note: Uma janela similar a esta é usada quando editamos uma tabela de label de uma label de um objeto ou quando editamos um node como uma tabela(acessível através do clique com o botão direito do mouse na janela Tree ou página de Labels). No entanto, o , , Limpa no on Reset e o botão  não estará disponível.

Comandos

Os seguintes comandos podem ser usados com as Global Tables. Quando especificado qual Global Table, você pode querer passar o nome da Global Table como uma string ou pode querer usar o comando `reftable()` para passar duplamente (este método é mais rápido do que acessar a tabela pelo nome para comandos de múltiplas tabela). Estes comandos podem também ser usados com outras tabelas no FlexSim ao passar no node da tabela. Veja a documentação de comandos para mais informações.

`reftable("GlobalTable")` Retorna a referência para a Global Table que pode ser passada para dentro da tabela subsequente de comandos.

`gettablenum(table, row, col)` Retorna os valores numéricos armazenados na tabela especificados por linha e coluna.

`gettablestr(table, row, col)` Retorna o valor string armazenado na tabela em uma célula especificada por linha e coluna.

`settablenum(table, row, col)` Define o valor numérico na célula da tabela especificada por linha e coluna.

`settablestr(table, row, col)` Define o valor da string na célula da tabela especificado por linha e coluna.

`gettablecell(table, row, col)` Retorna a referência para a célula (node) na tabela especificada por linha e coluna.

`settablesize(table, row, col)` Configura o tamanho da tabela para linhas e colunas. Veja a documentação de comandos.

`gettablecols(table)` Retorna o número de colunas na tabela.

`gettablerows(table)` Retorna o número de linhas na tabela.

`clearglobaltable(table)` Escreve zeros para todos as células numéricas e limpa o texto em todas as células string da tabela.

`addtablecol(table)` Adiciona uma coluna específica para a tabela.

`addtablerow(table)` Adiciona uma linha específica para a tabela

`deletetablecol(table, col)` Deleta uma coluna específica para a tabela.

`deletetablerow(table, row)` Deleta uma linha específica para a tabela.

`movetablecol(table, col, newcol)` Movimenta a coluna `col` para `newcol`.

`movetablerow(table, row, newrow)` Movimenta a linha `row` da tabela para `newrow`.

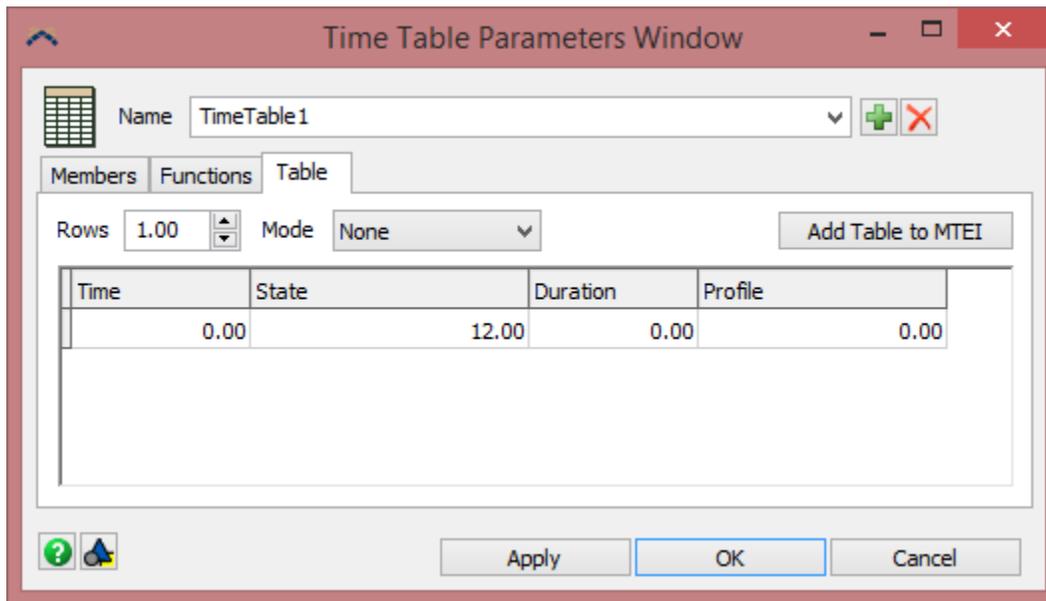
`gettableheader(table, row/col, rowcolnr)` Veja a documentação de comandos.

`executetablecell(table, row, col)` Executa a célula na tabela como Flexscript. Se a célula contém dados numéricos, retorna o número.

`importtable(table, "filename", importcolheaders, importrowheaders)` Importa dados de tabela do arquivo especificado pelo nome do arquivo de formato .CSV.

exporttable(table, "filename") **Exporta a tabela no formato .CSV para o arquivo especificado pelo nome.**

Time Tables Concepts



Topics

- **Functions**
- **Down State and State Profiles**
- **Modes**
 - None
 - Daily Repeat
 - Weekly Repeat
 - Custom Repeat
 - Date Based

Time Tables are accessed from the Toolbox. (View menu > Toolbox > Add > Time Table).

Time tables are used to schedule state changes, such as scheduled down-time, for specific objects in the model. Each Time Table may control many objects, and each object may be controlled by many Time Tables. A model may contain any number of Time Tables.

Functions

When the time table hits a down time, two sets of functions are called. First, the Down Function is fired. This happens once for each member of the Time Table. This allows you to stop the associated object, send a TaskExecuter to some specified location, etc. After all of the Down Functions have fired, the On Down function will fire once. The On Down function passes in the list of members and the table row associated with the down time.

Down State and State Profiles

State Profile

Down State

For each down time you can specify the state and state profile that the members will go to during that down period. An object can be tied to multiple Time Tables and be stopped multiple times. Each object stores its set of states each time it is stopped so the object can then be resumed and move back through the subsequent set of states.

State Profiles: When sending objects to a down state using state profiles, all members of the Time Table should have the associated state profile.

Modes

The TimeTable can be set up in different modes allowing for non-repeating/repeating schedules or schedules based upon date and time. The Daily Repeat, Weekly Repeat, and Date Based modes all utilize the Model Start Time and Date as defined in the Model Settings.

None

Rows Mode

Time	State	Duration	Profile
120.00		12.00	20.00
360.00		12.00	30.00
1000.00		12.00	20.00
1200.00		12.00	30.00

When the mode is set to none, the times listed in the table are absolute times based upon the model time units. The schedule will not repeat.

Daily Repeat

	:00	:05	:10	:15	:20	:25	:30	:35	:40	:45	:50	:55
12 AM												
1 AM												
2 AM												
3 AM												
4 AM												
5 AM												
6 AM												
7 AM												
8 AM	█	█	█	█	█	█	█	█	█	█	█	█
9 AM	█	█	█	█	█	█	█	█	█	█	█	█
10 AM	█	█	█	█	█	█	█	█	█	█	█	█
11 AM	█	█	█	█	█	█	█	█	█	█	█	█
12 PM	█	█	█	█	█	█	█	█	█	█	█	█
1 PM	█	█	█	█	█	█	█	█	█	█	█	█
2 PM	█	█	█	█	█	█	█	█	█	█	█	█
3 PM	█	█	█	█	█	█	█	█	█	█	█	█
4 PM	█	█	█	█	█	█	█	█	█	█	█	█
5 PM												
6 PM												
7 PM												
8 PM												
9 PM												
10 PM												
11 PM												

This graphical view allows you to specify the *Operation Time* and *Down Time* If you set your Time Table to repeat daily, the Model Start Time (as defined in the Model Settings). The Model Start Date does not come into effect. For example, if your Model Start Time is set to 08:00:00 AM and your Graphical Table looks like the above table, the members of the Time Table will begin the Model in an Operational state, and no functions will be fired. If however, you change the Model Start Time to 07:00:00 AM, when you reset and run your model, the Down functions will fire and the members will begin in a Down state. One hour later

(based on the model time units, so 3600 seconds if the model time is set to seconds), the Resume functions will be fired and the members will begin their Operational Time.

Weekly Repeat

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8 AM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
9 AM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
10 AM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
11 AM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
12 PM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
1 PM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
2 PM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
3 PM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
4 PM	██████████	██████████	██████████	██████████	██████████		
	██████████	██████████	██████████	██████████	██████████		
5 PM	██████████	██████████	██████████	██████████	██████████		

Setting the Time Table to repeat weekly will behave similarly to the Repeat Daily, except that the Time Table will also take into account the Model start *day of the week*. If the Model Start Time begins on a Tuesday at 08:00:00 AM and our Time Table is set to the above values, then the Time Table will skip all of Monday and jump to Tuesday at 08:00:00 AM with the Time Table's members being Operational. When the Model Time hits Friday at 05:00:00 PM, the members will go Down and remain down until Monday at 08:00:00 AM where the Time Table will start over.

Note: If you want to use the Graphical Time Table to build your Time Table, but you don't want to tie into the Model Start Time, you can set the Time Table's repeat time to *Daily* or *Weekly*, make your necessary changes, hit **Apply**, then set the Repeat time to *Custom*. This will auto fill the numerical table with the correct values associated with the Daily or Weekly table.

Custom Repeat

Rows Mode

Time	State	Duration	Profile
0.00		12.00	28800.00
61200.00		12.00	54000.00
147600.00		12.00	54000.00
234000.00		12.00	54000.00
320400.00		12.00	54000.00
406800.00		12.00	198000.00

The custom repeat mode is similar to the None mode, except you now have the option of specifying the repeat time for the table.

Date Based

Rows 42.00 Mode Date Based Add Table to MTEI

Start Date 8:00:00 AM 6/30/2014 Snap To Graphical Table

Date (select weeks)	Sun 00:00 12:00	Mon 00:00 12:00	Tue 00:00 12:00	Wed 00:00 12:00	Thu 00:00 12:00	Fri 00:00 12:00	Sat 00:00 12:00
Jun 29							
Jul 6							
Jul 13							
Jul 20							
Jul 27							
Aug 3							
Aug 10							

Start 5:00:00 PM 6/30/2014 End 8:00:00 AM 7/ 1/2014 Duration 00:15:00:00

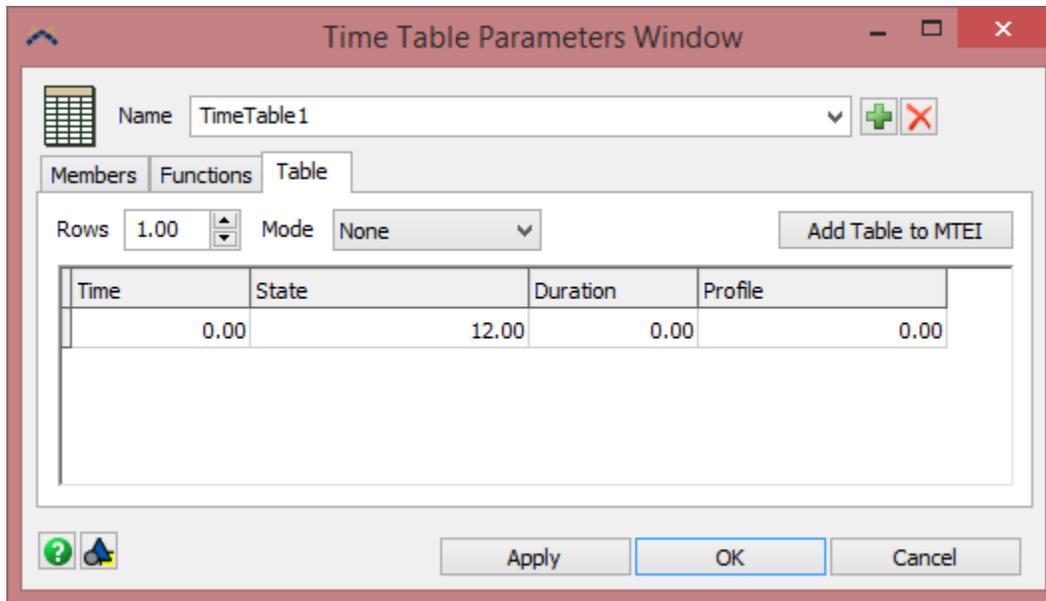
State Profile Default State Profile Down State 12 - scheduled down

The date based mode allows you to set up down times where each down time is associated with a specific date. These dates are based upon the Model Start Date and Time (as defined in the Model Settings) as well as a Time Table start date and time. This mode does not repeat.

In the above image, this Time Table has been set to start at 8:00 AM on 6/30/14. The first down time is set for 5:00 PM. If the Model Start Date and Time is set to the same start date and time as the Time Table, the associated members will start operational and go down at 5:00 PM. Changing the Time Table's start time to 09:00 AM will cause the members to go down at 6:00 PM in the model's time. Changing the Model Start Date and Time will allow you to jump into the Time Table's scheduled at the specified time.

See the Reference page for more information on how to use the date based interface.

Time Tables Reference



Pages

- **Members**
- **Functions**
- **Table**
 - **Table Editor**
 - **Graphical Table**
 - **Date Based**
 - **Date Based Editor View**

Name - The name of the TimeTable. The combobox has a list of all TimeTables in the model, allowing you to quickly jump to different Time Tables.

 - Create a new Time Table object.

 - Delete the current Time Table.

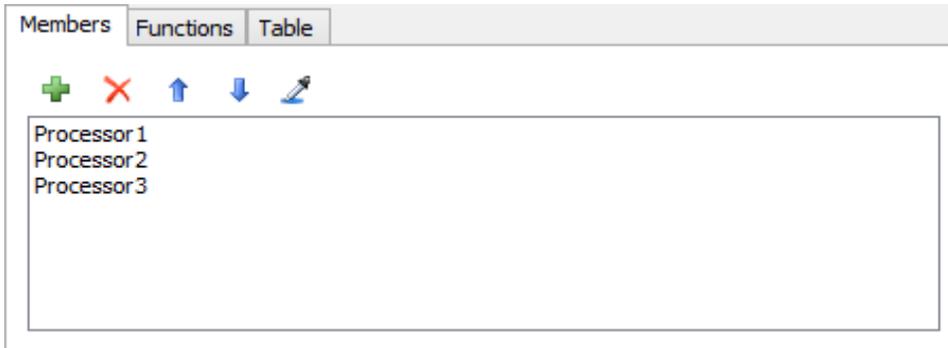
 - Adds the Time Table to a User Library as either a Draggable Icon or an Auto-Install Component.

Apply - Saves all changes to the Time Table.

OK - Saves all changes to the Time Table and closes the window.

Cancel - Cancels any unsaved changes made to the Time Table and closes the window.

Members



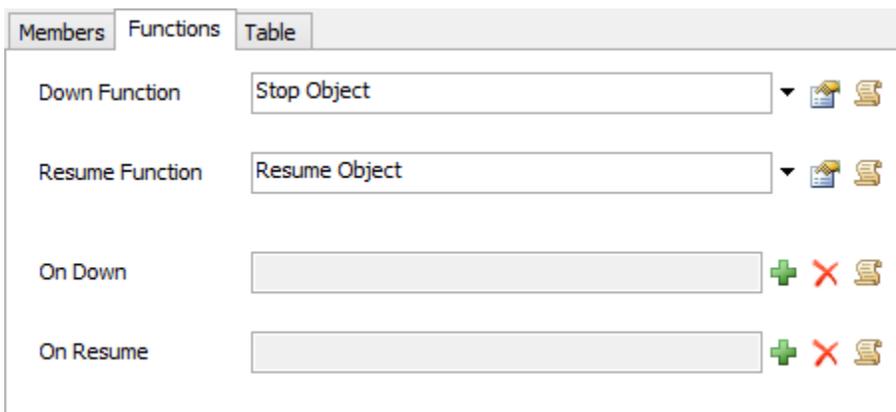
 - This will open an object selection GUI where you can select multiple objects in the model.

 - Removes the selected member(s) from the list.

  - Reorder's members Up or Down in the list.

 - Click to enter "Sample" mode, then click on any object in the model to add it as a member.

Functions



Down Function - This picklist is executed when the objects in the member list go down. It is executed once for each object in the member list. This is where you specify what to do to stop the object.

Resume Function - The picklist is executed when the objects in the member list resume their operation. It is executed once for each object in the member list. This is where you specify what to do to resume the object.

On Down - This picklist is fired immediately after the Down Function has been fired for all objects, but it is only executed once, instead of once for each object in the member list. See Down/Resume Trigger.

On Resume - This picklist is fired immediately after the Resume Function has been fired for all objects, but it is only executed once, instead of once for each object in the member list. See Down/Resume Trigger.

Table

Table Editor

Members Functions Table

Rows 6.00 Mode None Add Table to MTEI

Time	State	Duration	Profile
0.00		12.00	28800.00
61200.00		12.00	54000.00
147600.00		12.00	54000.00
234000.00		12.00	54000.00
320400.00		12.00	54000.00
406800.00		12.00	198000.00

Rows - This is the number of rows in the table.

Mode - This specifies the mode of the Time Table, None, Daily Repeat, Weekly Repeat, Custom Repeat or Date Based.

Add Table to MTEI - This button will add the table to the model's multiple table import accessed through the Excel Interface. When used with modes None, Custom Repeat and Date Based, the imported table takes the form of the numeric table (Time, State, Duration, Profile).

Table - Each row records the following:

- **Time** - This is the time since the table began that the state change should occur. If the mode is set to None, the table does not repeat, so the times listed in the Time column are absolute.
- **State** - This is the state that the objects controlled by this table will change into when the time table tells it to go down. If you click on this column, a drop-down box will appear at the top, giving you a list of possible states. Refer to the library objects for more information about what each state means to each object. Refer to the state list for a quick reference of each state's number and macro definition.
- **Duration** - This is how long the objects will stay in the new state before changing back to their original state.
- **Profile** - This is the state profile that the Down State is associated with. If you click on this column, a drop-down box will appear at the top, giving you a list of possible state profiles. Changing the State Profile will update the State column drop-down.

Rows 6.00 Mode Custom Repeat 604800 Add Table to MTEI

Time	State	Duration	Profile
0.00		12.00	28800.00
61200.00		12.00	54000.00
147600.00		12.00	54000.00
234000.00		12.00	54000.00
320400.00		12.00	54000.00
406800.00		12.00	198000.00

Repeat Time - Specify the time, in model time units, to repeat the Time Table.

Graphical Table

Rows 6.00 Mode Daily Repeat Add Table to MTEI

	:00	:05	:10	:15	:20	:25	:30	:35	:40	:45	:50	:55
12 AM												
1 AM												
2 AM												
3 AM												
4 AM												
5 AM												
6 AM												
7 AM												
8 AM	████	████	████	████	████	████	████	████	████	████	████	████
9 AM	████	████	████	████	████	████	████	████	████	████	████	████
10 AM	████	████	████	████	████	████	████	████	████	████	████	████
11 AM	████	████	████	████	████	████	████	████	████	████	████	████
12 PM	████	████	████	████	████	████	████	████	████	████	████	████
1 PM	████	████	████	████	████	████	████	████	████	████	████	████
2 PM	████	████	████	████	████	████	████	████	████	████	████	████
3 PM	████	████	████	████	████	████	████	████	████	████	████	████
4 PM	████	████	████	████	████	████	████	████	████	████	████	████
5 PM												
6 PM												
7 PM												
8 PM												
9 PM												
10 PM												
11 PM												

State Profile Default State Profile Make Operational Time
Down State 12 - scheduled down Make Down Time

Rows 6.00 Mode Weekly Repeat Add Table to MTEI

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8 AM							
9 AM							
10 AM							
11 AM							
12 PM							
1 PM							
2 PM							
3 PM							
4 PM							
5 PM							

State Profile Default State Profile Make Operational Time

Down State 12 - scheduled down Make Down Time

For more information on editing the Graphical Table, see the Concepts page.

Add Table to MTEI - This button will add the table to the model's multiple table import accessed through the Excel Interface. This will add the Graphical Table to the MTEI (not the numeric table).

State Profile - Specifies the state profile for the down state.

Down State - Specifies what state the members should go to when they enter their *Down Time*.

Make Operational Time - Select a set of cells from the table and press this button to make those times Operational (|||||).

Make Down Time - Select a set of cells from the table and press this button to make those times Down (empty cells).

Setting operational time: When the Time Table builds the numeric down time table based upon the graphical table, it looks at the text length of each cell. If the text length equals 0, the cell is referencing a down time. This allows you to put any string value into a cell to make that time an operational time.

Date Based

Rows Mode

Start Date Graphical Table

Date (select weeks)	Sun 00:00 12:00	Mon 00:00 12:00	Tue 00:00 12:00	Wed 00:00 12:00	Thu 00:00 12:00	Fri 00:00 12:00	Sat 00:00 12:00
Jun 29							
Jul 6							
Jul 13							
Jul 20							
Jul 27							
Aug 3							
Aug 10							

Start End Duration

State Profile Down State

Add Table to MTEI - This button will add the table to the model's multiple table import accessed through the Excel Interface. The imported table takes the form of the numeric table (Time, State, Duration, Profile).

Start Date - Specifies the start date and time of the Time Table. This value can be thought of like an offset to the Model Start Date and Time, allowing you to quickly modify every down time for the Time Table with regards to the model.

Snap To - Specifies the time to snap entries to in the view. By default there is no snap to. You can specify the preset 10 min, 15 min, 30 min and 1 hour, or set your own custom snap to time. The custom time is in model time units.

<input type="text" value="15 min"/> <input type="text" value="0"/>	<input type="text" value="Custom"/> <input type="text" value="7200"/>
--------------------------------------------------------------------	-----------------------------------------------------------------------

Graphical / Table - Switch between the graphical view and the numeric table view.

Start - The start date and time of the selected entry.

End - The end date and time of the selected entry.

Duration - The duration of the selected entry. Of form DD:HH:MM:SS.

State Profile - Specifies the state profile for the down state of the selected entry.

Down State - Specifies what state the members should go to for the selected entry.

Date Based Editor View

Creating entries - Click and drag anywhere in the white space to create a new down time. Snaps to the Snap To time.

Modifying entries - Click and drag in the middle of an entry to change the start time of the entry. Drag left or right to change the time along the week, or drag up or down to change weeks. Click and drag at the left or right edge of an entry to modify the duration. Snaps to the Snap To time. Start and end times as well as duration can also be modified through the GUI below the editor view.

Selecting weeks - Click and drag along the left side of the view under the Date column to select a week of entries. Any entries with their start time in the selected week will be selected.

Jun 29		■	■	■	■	■	■	■
Jul 6	■	■	■	■	■	■	■	■
Jul 13	■	■	■	■	■	■	■	■
Jul 20	■	■	■	■	■	■	■	■
Jul 27	■	■	■	■	■	■	■	■

Keyboard shortcuts

- *Ctrl + C* - Copy selected entries.
- *Ctrl + X* - Cut selected entries.
- *Ctrl + V* - Paste selected entries. If a single entry was copied, the pasted entry will be pasted below the cursor position. If an entire week was selected, the entries are pasted based upon the week that the cursor is positioned over, keeping start times relative to the start of the week.
- *Ctrl* - Hold the control key and click and drag an entry to create a copy.
- *Arrow Keys* - Select an entry or a week of entries and use the up, down, left and right arrow keys to adjust the entry start times based upon the Snap To time.

Controlling the View

- Hold the Ctrl key while using the scroll wheel to zoom in on the mouse location.
- Hold the Shift key while using the scroll wheel to scroll left and right.
- Use the scroll wheel to scroll up and down in the view.
- Click and drag along the bottom scroll bar to zoom in/out and scroll left and right.

Graphical User Interfaces

1. Conceitos
2. Exemplo
3. Referencia
 - GUI Events e View Attributes
 - View Attributes Reference

Graphical User Interfaces - Conceitos

Tópicos

- Por que criar GUIs?
- GUI Views
- Desenvolvendo um GUI
- Dicas na construção dos GUIs
- Trabalhando com o editor GUI
- Percorrendo o Tree
- Lista de Atributos
- Tree View – Visualizando Atributos vs. Visualizando a Estrutura
- Linkando ao Modelo
- Copiando um GUI Existente

Graphical User Interfaces (GUIs) são acessado do menu Tools > **Graphical User Interfaces**.

GUIs permite a você criar sua própria janela de interface para o seu modelo e seus objetos. Um GUI pode comunicar-se com qualquer objeto no modelo, de qualquer forma que você queira.

GUIs são tão comuns que existe um botão na barra de ferramentas para abrir tal janela. 

No FlexSim, GUIs são armazenados como um node com sub-nodes em uma árvore (tree). Cada node representa uma parte de seu GUI. Os atributos nos dados do objeto de cada um destes nodes representam as variáveis que afetam como estas partes do seu GUI trabalham. Quando um GUI é aberto, ele cria um novo node em VIEW:/active que é uma cópia.

Por que criar GUIs?

- Nem todos sabem como navegar no FlexSim. Você pode dar aos outros a habilidade de manipular certas partes de seu modelo sem ter conhecimento no FlexSim.
- Economize tempo em testes extensivos. Os GUIs podem ajudar você a alterar os parâmetros rapidamente em seu modelo durante os testes de processos.
- O GUI se parece bem mais profissional.

Visualizações de GUIs

Os GUIs do FlexSim são feitos de blocos de construção chamados views. São janelas que performam regras específicas e podem ser combinados hierarquicamente. Estas vistas permitirão a você visualizar e manipular dados na estrutura Tree do FlexSim. Uma vez que os dados tomam diversas formas, existem muitos tipos de vistas. Abaixo um lista de tipos de vistas (views) disponíveis no FlexSim. O número próximo a cada tipo de vista (view) é a viewwindowtype. Este número é uma referência direta ao tipo de view e não irá alterar nos futuros lançamentos da FlexSim.

Para mais informações sobre os GUI views, veja a página View Attributes Reference.

View Types			
Windows Common Controls		FlexSim Registered Controls	
Static (or label) Button	103	Dialog Panel	4 102
Radiobutton	100	Groupbox	107

Checkbox	106	Table	5
Edit	105	Graph	6
Trackbar	101	3D View	2
Combobox	122	Tree	0
Listbox	109	IconGrid	7
TabControl	114	Script	8
Scrollbar	115	HTML	124
Statusbar	104		
Spinner	110		
DateTimePicker	123		
TreeView	125		
	119		

Designing a GUI

Existem alguns livros que descrevem sobre a filosofia de construção de um GUI. O maior tópico, fácil de usar. Evite a tentação de fazer uma janela GUI que faça muito mais do necessário. GUIs devem ser bem simples para navegar e focados em suas funções.

É recomendado você arquitetar e estruturar o que você quer que sua janela GUI se pareça antes de você abrir o editor GUI.

Dicas na Construção dos GUIs

É importante ter uma boa convenção de controle de nomes. Um bom e curto nome irá fazer referência ao objeto em código bem mais fácil.

Use Panel controls (invisível e não invisível) para agrupar, movimentar e copiar seções de seu GUI.

A construção do GUI essencialmente consiste de adicionar views para sua janela GUI e dar estas visualizações para os atributos apropriados. Enquanto a opção "View Specific" da lista de atributos dá a você algumas pistas de qual atributos são apropriado para adicionar, existem ainda outras formas de você tornar-se mais confortável e experiente construindo os GUIs.

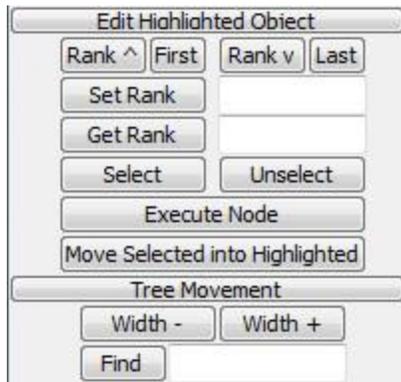
Outra forma que você tornar-se mais familiar com a construção dos GUIs é através de simples experimentações. Adicione um atributo que você nunca adicionou ou viu antes e veja como isso afeta o GUI. Se não houver nenhuma alteração, o atributo pode não estar fazendo nada para este tipo de visão, ou que de alguma forma não foi implementado de forma correta. Mexer com diferentes configurações e valores para o atributo para ver como ele altera o GUI. Se ainda não muda nada, então é só seguir em frente e tentar outra coisa. Se isso acontecer, ótimo! Essa é mais uma ferramenta que você tem em sua base de conhecimento.

Prática, prática, prática. Conforme você continuar a construir mais e mais GUIs, a velocidade e a eficiência com que você irá fazê-lo irá aumentar significativamente, e você será capaz de obter uma sensação de que os GUIs são cada vez mais amigável.

Trabalhando com o Editor GUI

Aqui estão algumas dicas de como trabalhar com o editor GUI:

- Pense não só na forma de organizar os controles na tela GUI (editor de janela), mas também na árvore (tree). Ter familiaridade com a árvore(tree) é um requisito para a boa construção de um GUI.
- Movendo controles na árvore pode ser feito usando o recurso Edit Highlighted Object e Tree Movement fields na janela de construção do GUI.



- O atributo nodes dos controles do GUI define como o controle irá se parecer/comportar. Cada controle tem um conjunto de atributos nodes padrão mas outros podem ser adicionados do palette Controls da janela GUI Builder.

Percorrendo o Tree

@ : Este símbolo indica o percurso para ir para a vista principal do node atual. Para o node coldlink, é a janela principal do GUI.

> : Este símbolo indica o percurso para ir para dentro do atributo dos nodes na tree.

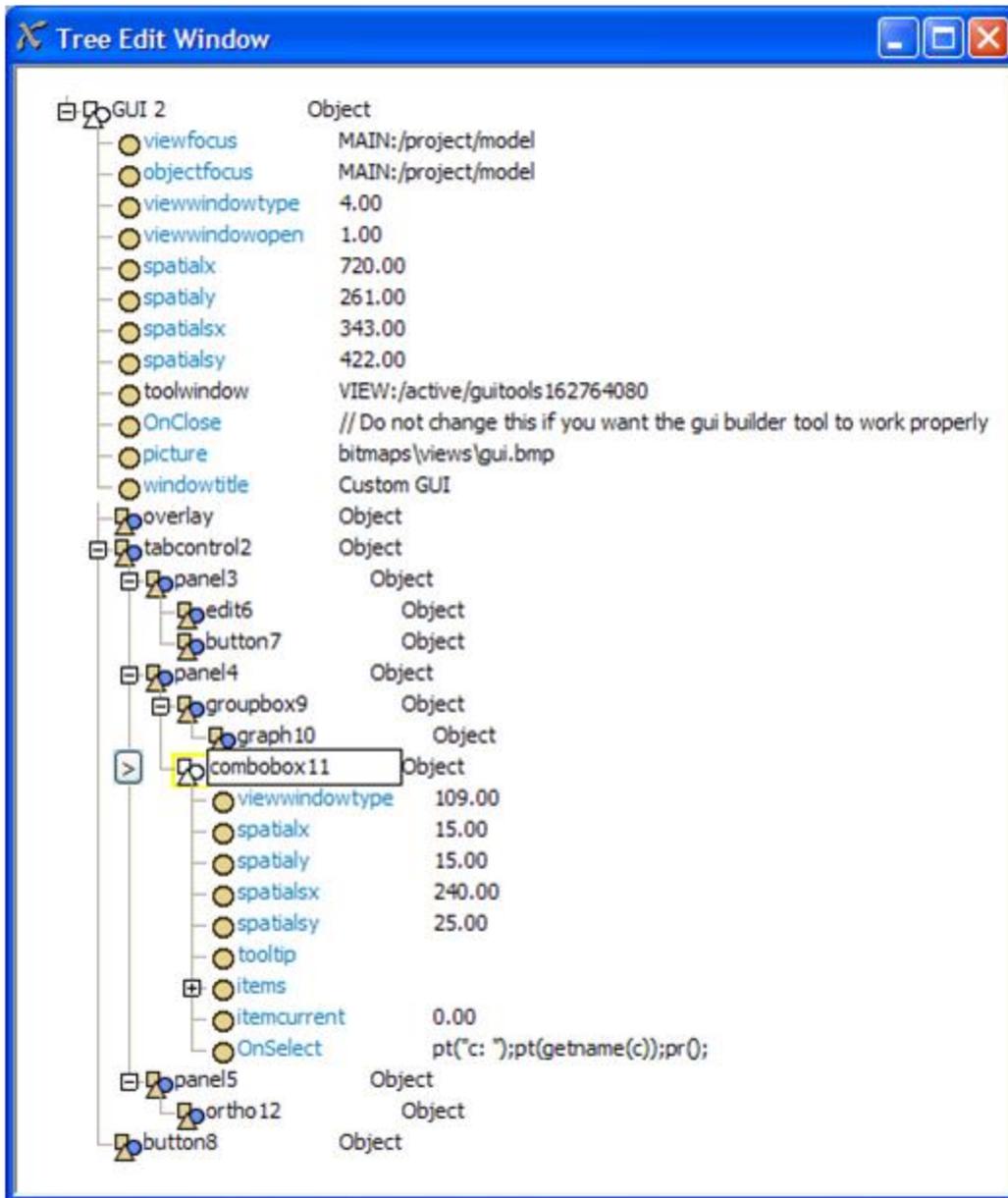
+ : Este símbolo indica o percurso a ser lido para o texto do node atual como um caminho para um objeto, e vai para aquele objeto.

/ : Este símbolo indica o percurso a ser caminhado para dentro do node atual do sub-tree.

.. : Este símbolo indica o percurso a ser caminhado para o node atual do parente tree, ou um nível acima.

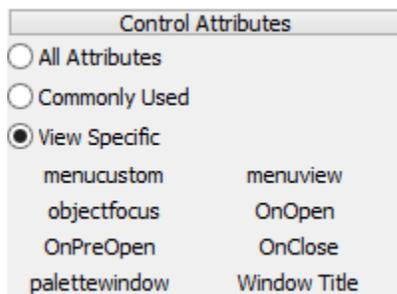
? : Este símbolo causa uma busca na árvore (tree) para um nome subsequente. Por exemplo, node("/?FlowItemBin", model()) irá buscar por um node com o nome FlowItemBin na árvore do modelo. Isto retorna o mesmo node referência como um caminho explícito que seria: node("/Tools/FlowItemBin", model()). Abaixo é um exemplo de tabela para referência da Tree (Tree mostra a tabela abaixo).

Code:	Returns:
c	combobox11
itemcurrent(c)	itemcurrent
node(">itemcurrent",c)	itemcurrent
ownerobject(c)	panel4
node("../",c)	panel4
node("../panel5/ortho12",c)	ortho12
ownerview(c)	GUI 2
node("@",c)	GUI 2
node("@>toolwindow",c)	toolwindow
node("@/tabcontrol2/panel3/1",c)	edit6
node("../3/1",c)	ortho12
node("@/button8",c)	button8
node("@>objectfocus+",c)	model



Lista de Atributos

A lista de possíveis atributos do lado esquerdo do tree GUI's builder tem três opções para a listagem de atributos de visualização. Eles são: all attributes (todos os atributos), commonly used attributes (atributos comumente usados) e view specific attributes (atributos específicos).



All Attributes - Se esta opção é selecionado, todos os possíveis atributos são mostrados. Usualmente você não precisará usar esta opção.

Commonly Used - Se esta opção é selecionado, uma lista de atributos comumente usados será mostrado. Estes incluem coisas como atributos de alinhamento, o qual permite a você ancorar as posições de vistas ou tamanho ao lado direito ou na parte inferior da janela de visualização.

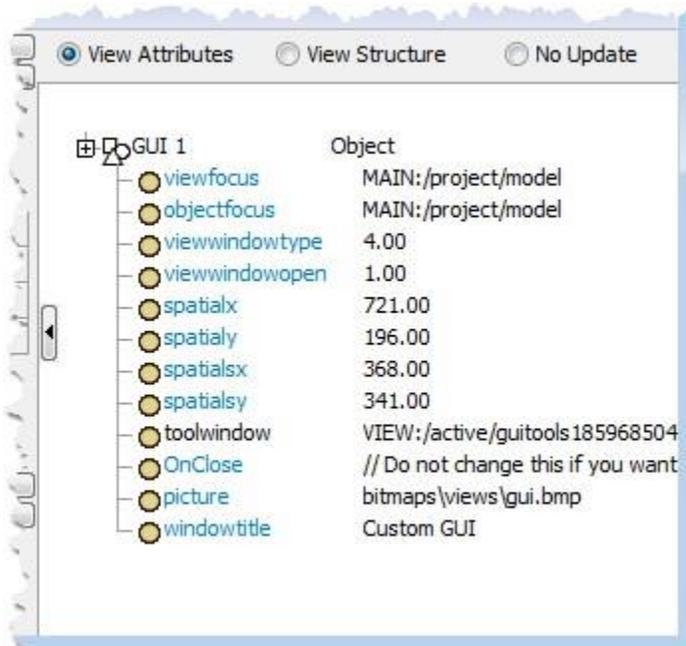
View Specific - Esta opção é a padrão. Se selecionado, então a lista de atributos que é mostrado será específica ao tipo de vista, ou vista, que você está atualmente editando. Se você clicar no botão view, uma lista de atributos irá aparecer que é específica para o botão. Isto inclui os atributos como OnPress, o qual é executado quando um botão é pressionado. A label view terá uma diferente conjunto de atributos que são usados para ele, etc.

Para ver uma listagem detalhada de atributos e seus usos por tipo de controle, vá em "View Attributes Reference".

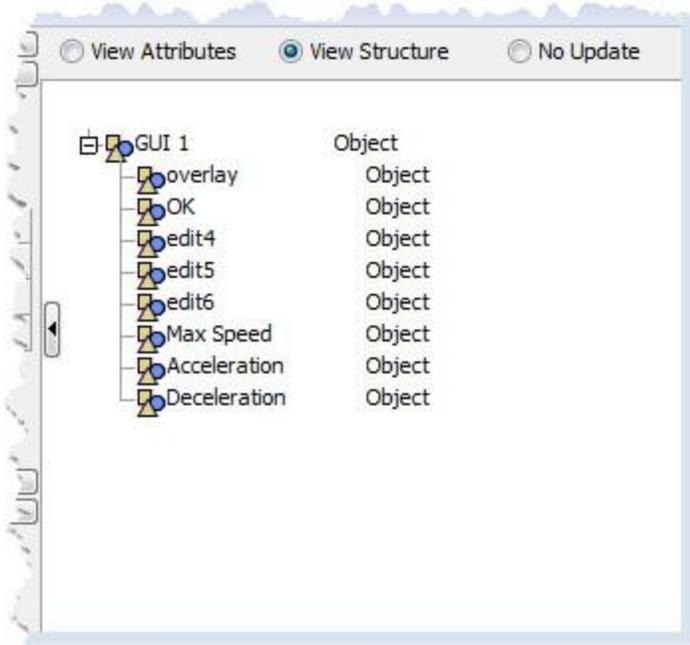
Tree View – Visualizando Atributos & Visualizando Estrutura

A view árvore (tree) GUI's builder tem duas opções para visualizar a árvore. Eles são: view attributes e view structure.

View Attributes - Esta opção irá visualizar os atributos da vista atualmente selecionada.



View Structure - Esta opção irá visualizar a estrutura da árvore (tree) da vista atualmente selecionada. Isto é útil para rearranjar e editar a estrutura do GUI's tree.



No Update - Esta opção irá fazer com que o GUI builder não atualize o tree focus quando você clicar sobre um controle no gui canvas. Alguns usuários preferem isto tendo em vista que não altera o view toda vez que você clicar no canva.

Linking ao Modelo

Flexsim usa dois tipos de links quando se faz necessário referenciar um objeto GUI para dentro do modelo. Coldlinks e hotlinks.

coldlinks

Um atributo coldlink fala para o campo de visualização de textos para ser linkado para um determinado node na árvore atributo do objeto. O link é "cold" por que ele pega o valor somente quando a janela abre, e configure o valor somente quando o botão Apply é pressionado.

Exemplo: Na guia Processor do Processo, o Max Content é um exemplo de coldlink.

hotlinks

Um "hot" link, de outra forma, será continuamente atualizado em seu campo de texto conforme seu valor vai se alterando.

Exemplo: A página Statistics do Processo, o content statistics são exemplos de um hotlink.

hotlinkx/coldlinkx

Similar aos comentários acima, mas somente quando usado um string especial de caracteres, o código FlexScript é permitido estabelecer o link. O "c" passado é um argumento 2 do exemplo abaixo e representa o próprio objeto do atributo.

Exemplo: `return(node("@>objectfocus+",c));`

Copiando um GUI Existente

Se você ver uma janela view ou GUI que você quer copiar para outro GUI, existem duas formas de fazê-lo:

Copy from Tree

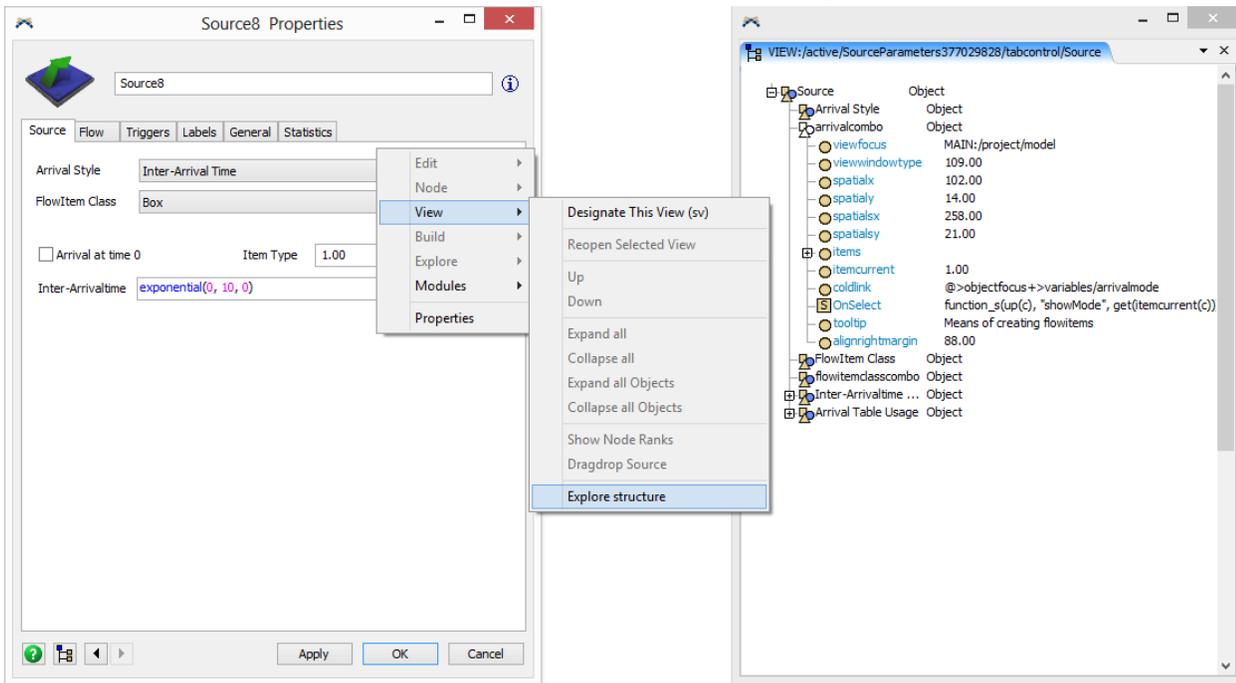
Encontre esta vista no tree, clique com o botão direito sobre ele e selecione "Edit > Copy".

Copy from Window

Abra a janela que você deseja copiar. Clique com o botão direito do mouse sobre ele e selecione "View> Explore Structure " no menu pop-up. Isto vai abrir uma janela da tree que mostra a estrutura de janela. A partir daqui você pode ver os atributos daquela janela para ver quais atributos estão presentes, de modo

que você pode adicionar esses atributos para o seu próprio GUI. Botão direito do mouse no view node e seleccione "Edity> Copy"

Agora vá para a sua própria estrutura de árvore do GUI. Na estrutura de árvore do GUI, crie um novo node em branco com o botão direito sobre o container view que você deseja colocá-lo em, e seleccione "Node> Insert Into". Em seguida, clique com o botão direito no novo node em branco, e seleccione "Edit> Paste". Isto copiará a vista em sua GUI. Pressione F5 para atualizar a exibição com seu ponto de vista adicional.



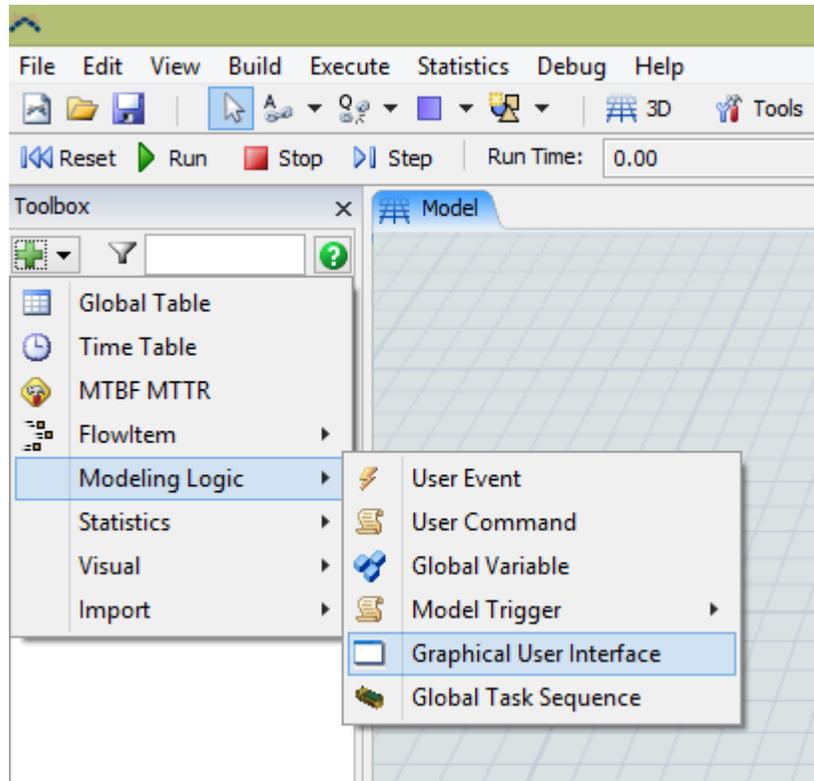
Graphical User Interfaces Example

Tópicos

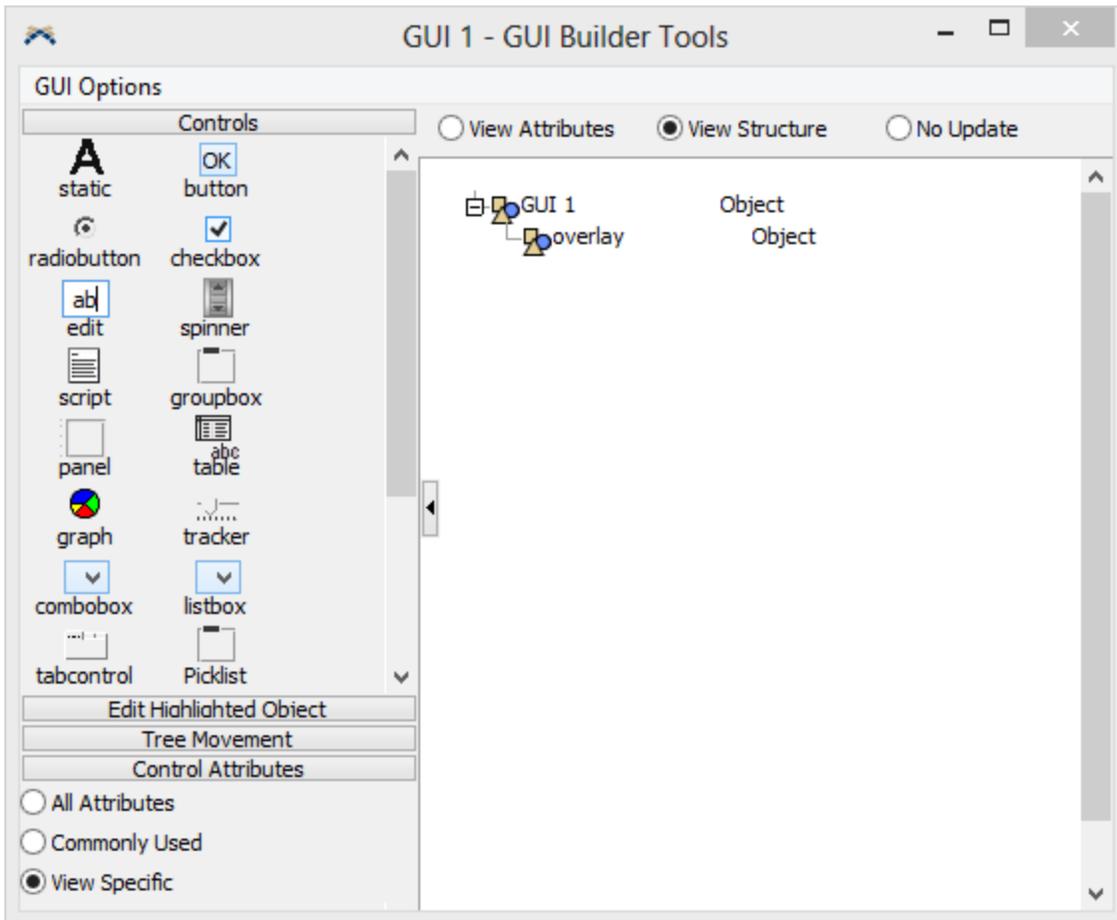
- Building a GUI
- Change View Names
- Link the Edit Views
- Direct Model Objects to This GUI

Construindo um GUI

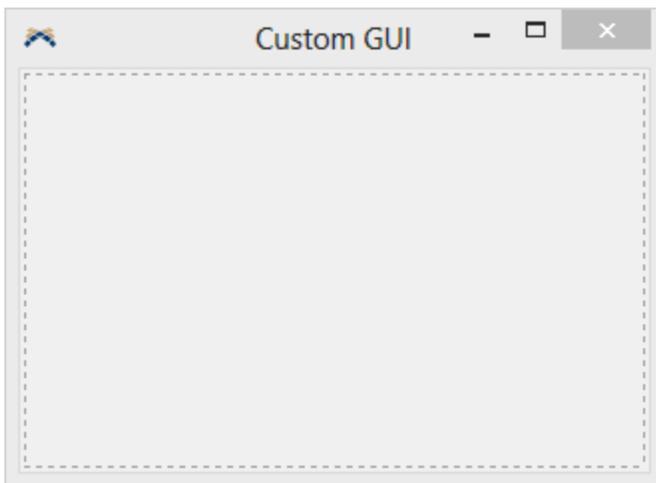
Adicione um GUI indo em **Tools > Toolbox > Modeling Logic > Graphical User Interfaces**.



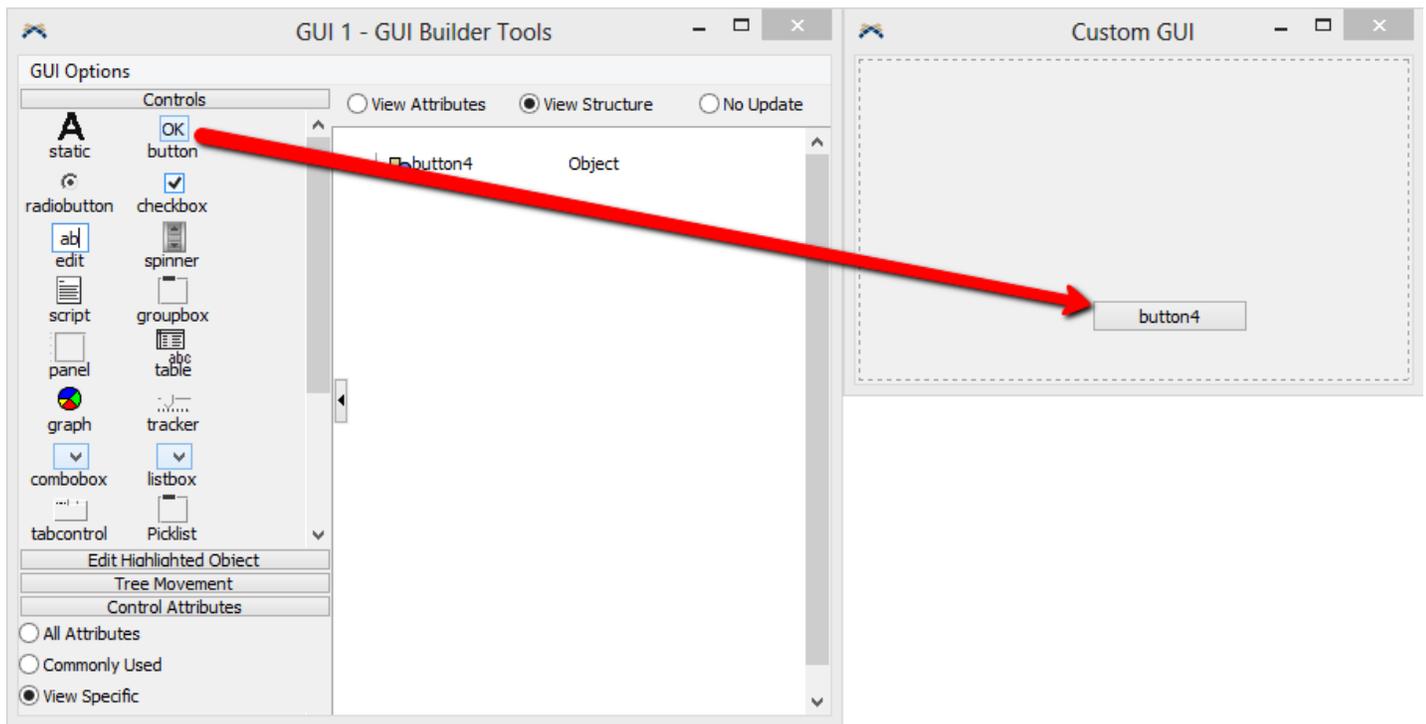
Depois de ter adicionado um GUI duas janelas se abrirão. A janela do lado esquerdo é chamado o construtor de GUI (GUI builder). Esta janela fornece várias ferramentas para a construção de sua interface gráfica (GUI).



A janela na direita é chamada de GUI canvas. Esta janela mostra o que o seu GUI se parece. Inicialmente está em branco. Nós iremos adicionar alguns GUI view nele simplesmente arrastando os objetos da janela GUI Builder.



Como um exemplo, nós iremos criar um GUI simples que permite a você editar a velocidade máxima (max Speed), aceleração (acceleration) e desaceleração (deceleration) de um objeto Operator. Primeiro, vamos arrastar algumas vistas simples para dentro de nosso GUI canvas. Do painel superior do GUI builder, arraste um botão para dentro do GUI canvas.

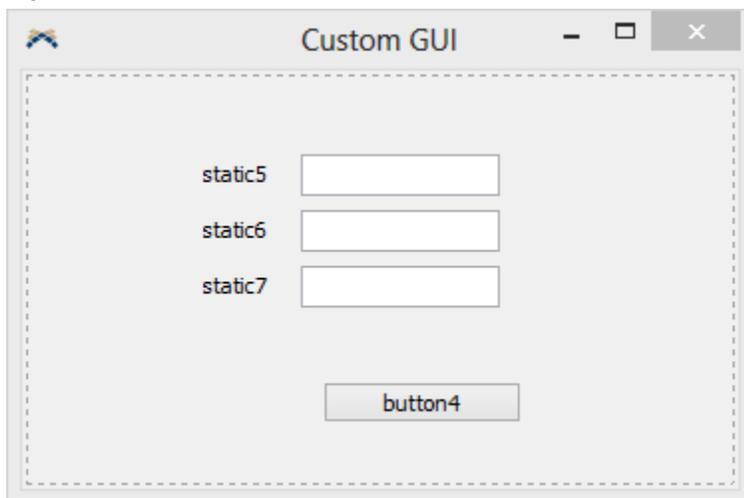


Você pode agora selecionar o botão ao clicar sobre ele. Quando o botão é selecionado, você verá uma linha pontilhada ao redor dele, assim como um quadrado preto na parte inferior do lado direito dele.



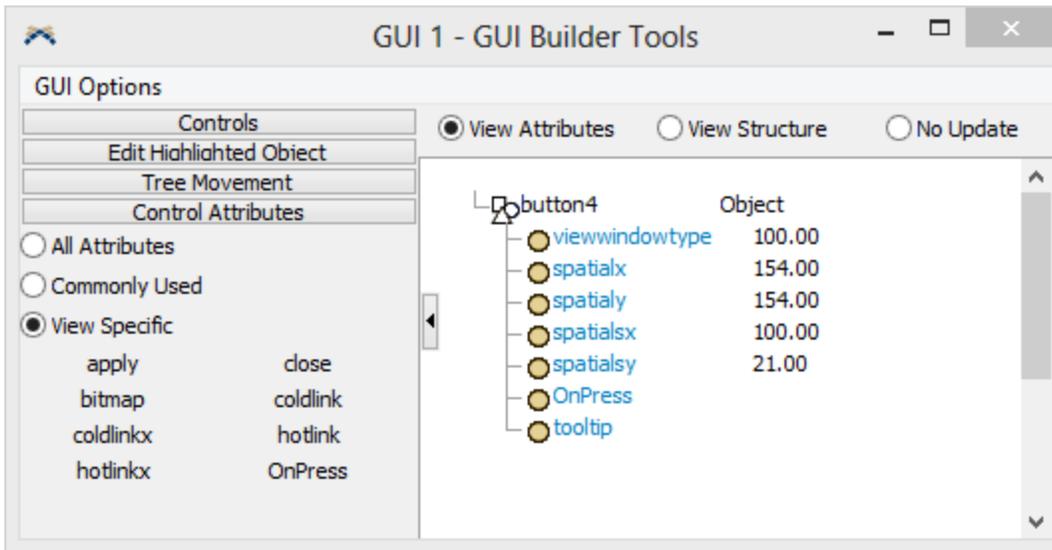
Para movimentar o botão, somente clique e arraste o botão para a localização que você deseja. Para alterar o tamanho do botão, clique e arraste o quadrado preto.

Agora, adicione três static views e três edit views para dentro do GUI canvas, conforme mostrado abaixo.

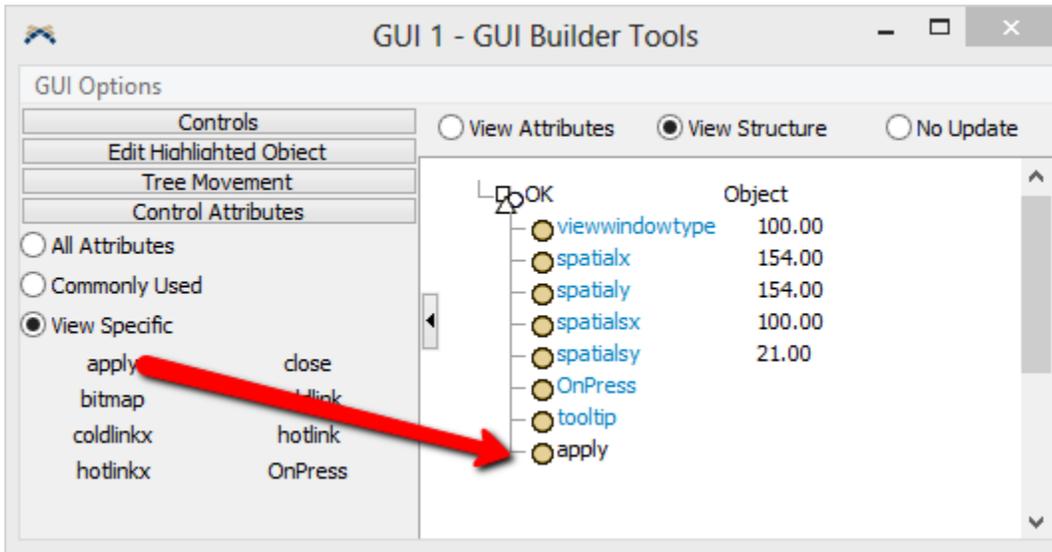


Altere os nomes de visualização

Agora queremos mudar os nomes e atributos de nossos pontos GUI views. Reduza a barra de ferramentas de controles pressionando o botão "Controls" no GUI Builder. Primeiro, vamos fazer do botão um botão OK que irá aplicar as edições que o usuário fizer, em seguida, feche a janela. Clique no botão. Em seguida, clique no botão de opção "View Attributes" na janela do construtor de GUI. Observe que o botão e seus atributos agora são mostrados na tree view do GUI Builder.



Clique sobre o nome do botão ("button4" em nosso caso) e renomeie ele como "OK". Na parte esquerda existem alguns atributos que podem ser adicionados para o botão. Para adicionar um atributo, somente arraste-o da grade de ícones da parte esquerda para área em branco na tree view.



Adicione um atributo Apply e close para o botão. O atributo Apply faz com que todos os coldlinks e hotlinks encontrados em view, serem aplicados para os seus respectivos nodes de destinos quando o botão é pressionado. Coldlinks e hotlinks serão explicados mais tarde. O atributo close faz com que a janela se feche quando este botão é pressionado.

Método Alternativo - Alternativamente, ao invés de adicionar os atributos apply e close, você pode adicionar o seguinte código para o atributo OnPress do botão:

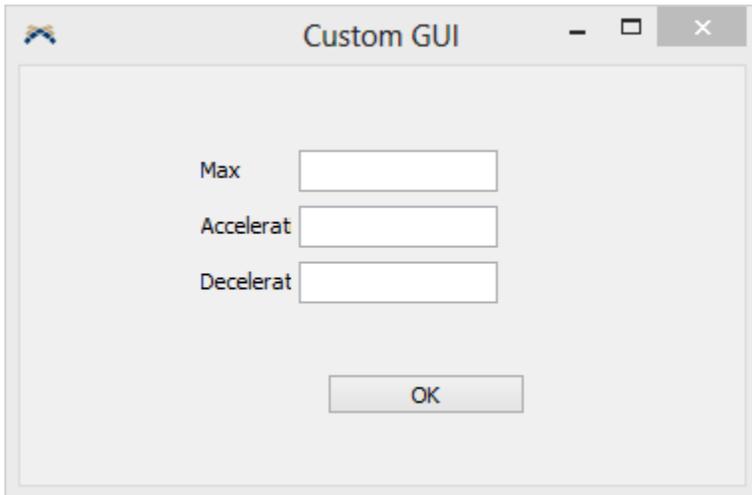
```
applylinks(ownerview(c));
postclosewindowmessage(ownerview(c));
```

Este código terá o mesmo efeito que os atributos apply e close.

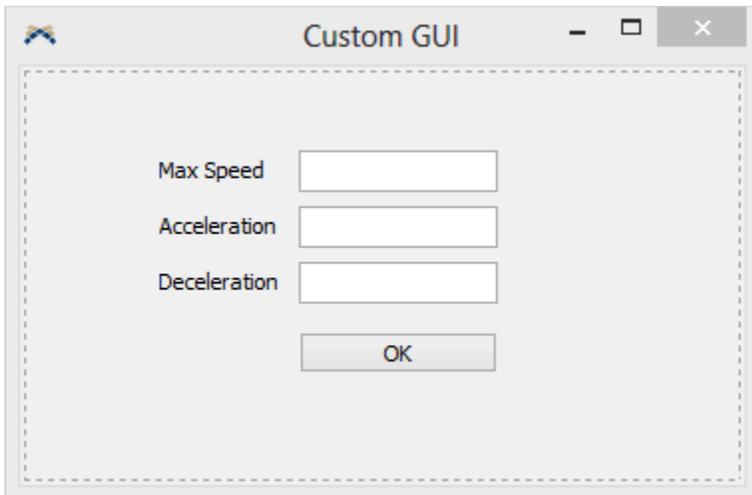
Agora clique no primeiro ponto de vista estático. Mais uma vez, a visão estática e seus atributos devem aparecer na tree view do GUI Builder. Para o nome da primeira visão estática, digite "Max Speed". Você não vai ver a mudança de nome na static view até que a exibição seja atualizada. Agora clique no segundo ponto de vista estático e defina seu nome para "Acceleration". Em seguida, clique na última visão estática e alterar o seu nome para "Deceleration".

Agora que você fez algumas mudanças para os pontos de vista do GUI que iremos atualizar para a tela do GUI canvas para ver essas mudanças. Clique na janela do GUI canvas e, em seguida, pressione o botão

F5. Isso vai mudar a tela GUI a partir do modo de edição ao modo de visualização normal. Agora sua janela deve ser semelhante a uma janela normal, sem as linhas pontilhadas em torno dele.

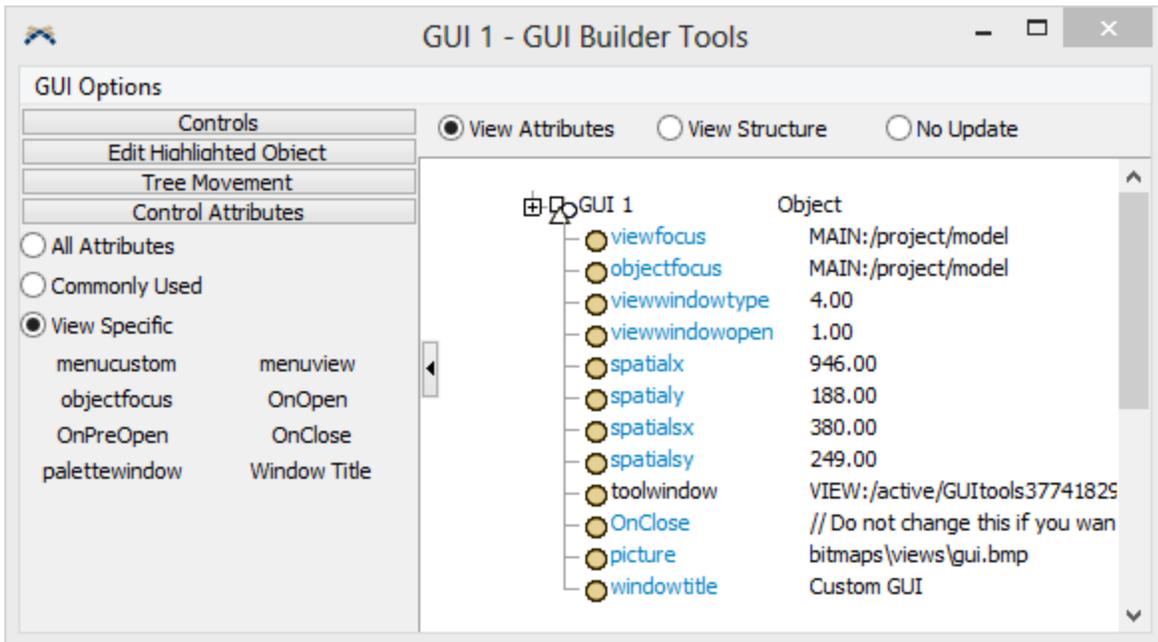


Observe que os textos das variáveis não são grandes o suficiente para o texto se encaixar onde eles estão mostrando. Para corrigir isso, volte para o modo de edição, selecionando a tela GUI e pressione F5 novamente. Agora reorganize os tamanhos e localizações de seus pontos de vista, de modo que tenha espaço suficiente para mostrar todo o texto contendo os nomes das variáveis.



Link os Edit Views

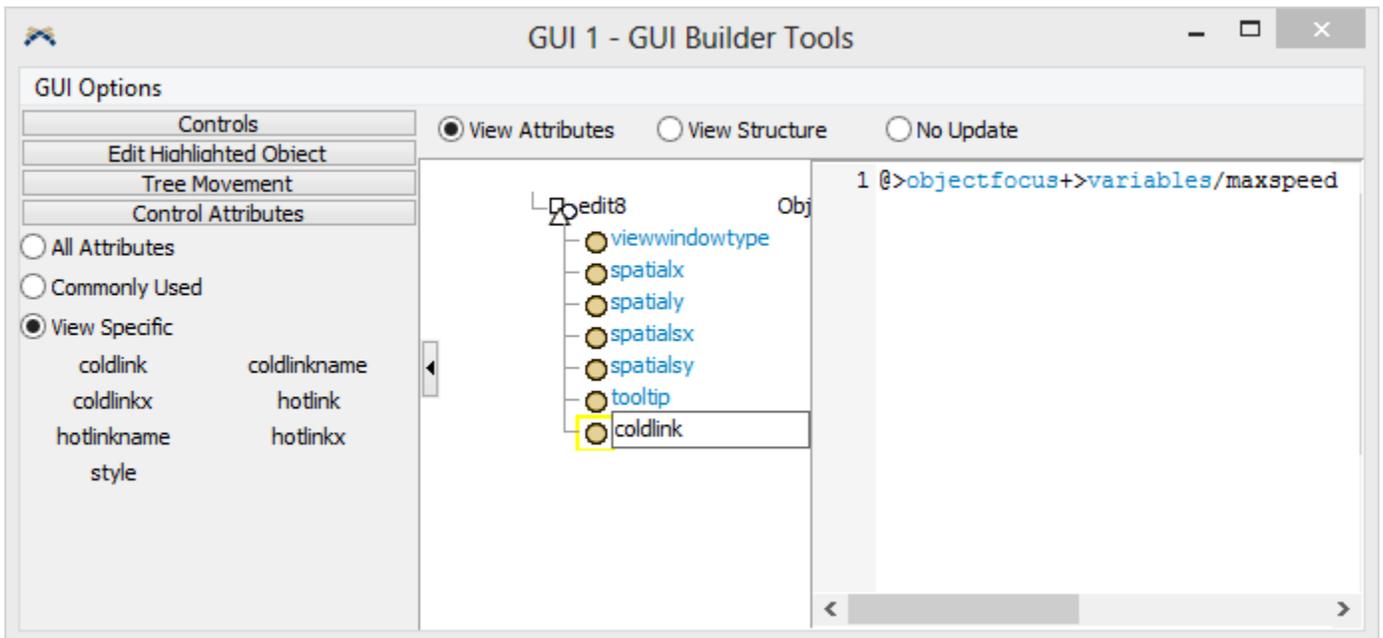
Agora vamos ligar os edit views até seus próprios nodes no modelo. Primeiro vamos explicar alguns conceitos. Clique em uma área vazia na sua tela GUI. Isso fará com que a tree view do GUI Builder mostre os atributos da janela principal do GUI.



Note que existe um atributo chamado objectfocus. Agora mesmo, este atributo mostra o caminho "MAIN:/project/model". Mais tarde, porém, uma vez que associarmos este operador com este GUI, e então darmos um duplo clique sobre o operador para abrir esta janela, o atributo objectfocus será alterado. Ele especificará um caminho para o Operator que você está editando. Por exemplo, você associou um operador chamado de Bob com este GUI. Quando você der um duplo clique sobre o Bob, uma instância deste GUI é criada, e este atributo objectfocus é configurado para o caminho da string: "MAIN:/project/model/Bob". Isto é importante saber quando criamos campos editáveis que são linkados para nossos objetos. Agora, vamos retornar para o primeiro campo editável.

Clique sobre o primeiro view edit para visualizar seus atributos na tree. Adicione um atributo coldlink da lista a esquerda.

Agora entre com o seguinte como texto do atributo coldlink: @>objectfocus+>variables/maxspeed



Links devem ser feitos para um node que contém alguns valores desejados. Às vezes linkando diretamente para o node apropriado do objeto é indesejável, como por exemplo, code nodes. Neste caso, link para uma label e use o valor da label como parte do código proprietário (use executestring()).

O texto coldlink/hotlink especifica um caminho para node que o campo editável está associado com. Este caminho inicia no node coldlink, e deve especificar um caminho para o node da variável maxspeed do operador. Os símbolos diferentes no caminho são formas de especificar como direcionar o tree para o node de destino.

O coldlink que você especificou pode fazer o seguinte:

1. Iniciando no node coldlink, vá para a janela principal GUI (@).
2. De lá, vá para dentro do seu attribute tree e encontre o atributo chamado objectfocus (>objectfocus).
3. Interprete o texto do node objectfocu como um caminho para um node, e vá para aquele node (+). Lembre-se que quando nós abrirmos este janela para o operador Bob, o atributo objectfocus será alterado para "MAIN:/project/model/Bob". Agora, irá para o nosso operador Bob no modelo.
4. De lá, vá para dentro da árvore de atributo do objeto Bob, e encontre os nomes das variáveis node (>variables)..
5. Daqui, vá para o node's sub-tree e encontre o node chamado maxspeed (/maxspeed).

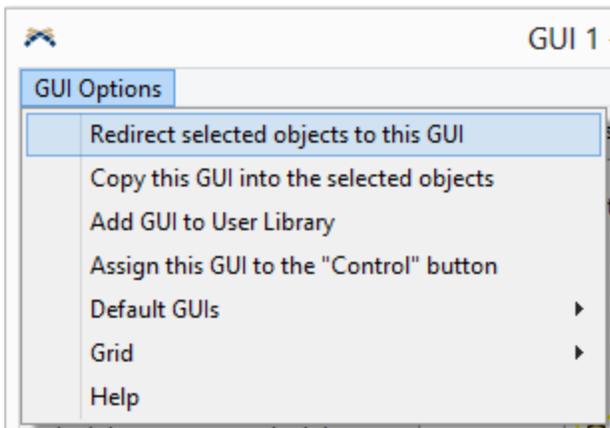
Agora conecte os outros dois edit views. Clique sobre o segundo edit view, e então na tree view adicione um atributo coldlink e especifique seu texto como: @>objectfocus+>variables/acceleration. Faça o mesmo para o último edit view e configure seu atributo coldlink para: @>objectfocus+>variables/Deceleration.

Direcione os objetos do modelo para este GUI

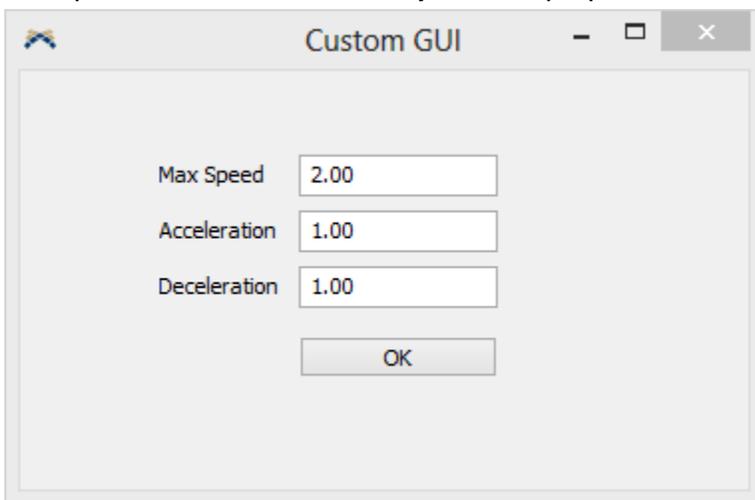
Agora que nós terminamos de criar o nosso GUI, vamos direcionar um operador em nosso modelo para este GUI. Arraste um operador para dentro do seu modelo. Selecione o operador, mantendo pressionado a tecla Control ou Shift e ao mesmo tempo clicando sobre o operador.



Agora retorne para a janela GUI Builder, e de seu menu, selecione "GUI Options > Redirect Selected Objects to this GUI".



Isto irá redirecionar todos os objetos selecionados no modelo para usar este GUI ao invés de sua janela de propriedades usual. Agora, feche a janela do GUI Builder. Isto irá fechar automaticamente a janela do GUI canvas também. Agora dê um duplo clique sobre o operador que você selecionou. Isto irá abrir a janela do GUI que você criou ao invés da janela de propriedades normal.



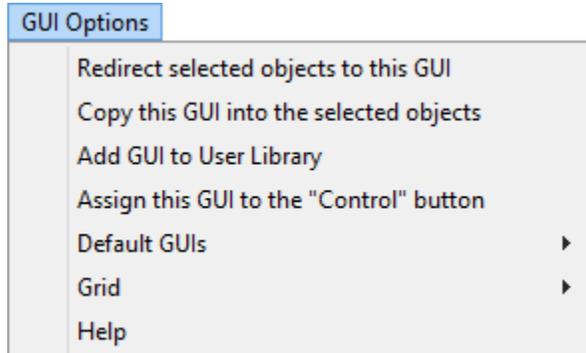
Faça algumas mudanças nos valores de max speed, acceleration e deceleration, e então pressione o botão OK para aplicar estas mudanças. Abra a janela novamente para verificar que se suas alterações foram aplicadas corretamente.

Graphical User Interfaces Referência

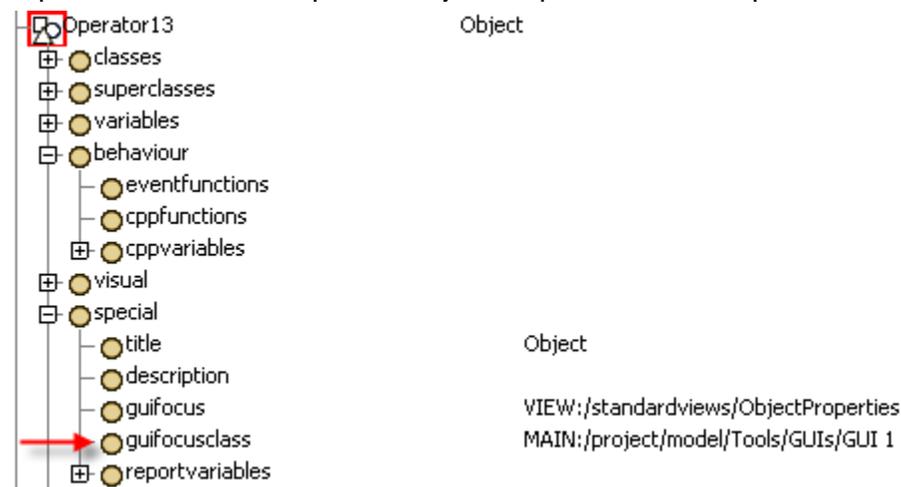
Para informações de como usar o Graphical User Interface, veja a página de exemplo.

GUI Builder Menu

O menu de edição do GUI builder permite a você fazer diversas operações com o GUI uma vez que você o criou.

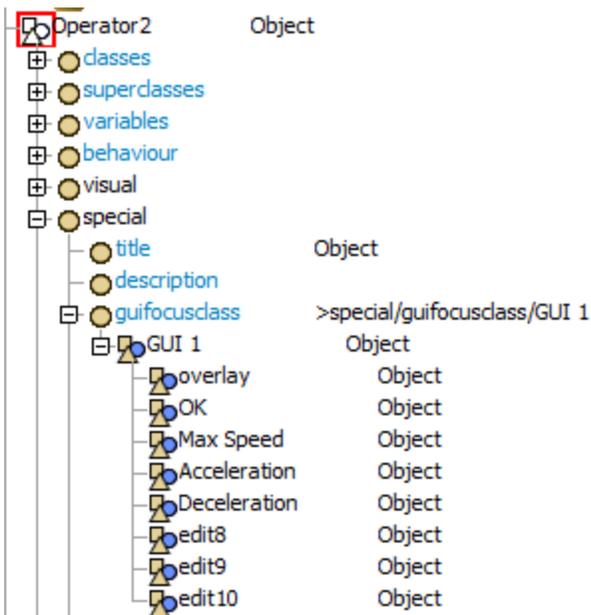


Redirect Selected Objects to this GUI - Esta opção irá redirecionar o atributo guifocusclass de todos os objetos selecionado no modelo para apontar para este GUI. Para ilustrar isso, explore o model tree, e encontre o operador que você redirecionou para o GUI criado. Expanda sua árvore de atributos e expanda o node chamado “especial”. Dentro do especial deve ter um node chamado guifocusclass. Este texto node especifica um caminho para uma janela que será aberta quando o objeto recebe um duplo clique.



Note que o caminho para este atributo é "MAIN:/project/model/Tools/GUIs/GUI 1". Isto é onde nosso GUI é armazenado. Quando nós selecionamos a opção redirect, ele alterou seu caminho da página de propriedades normal do objeto Operator para nossa página.

Copy this GUI into Selected Objects - Esta opção irá criar uma cópia completa do GUI e armazenar dentro de cada objeto selecionado. Para mostrar exatamente como isto funciona, vamos fazer isso para o GUI que nós criamos em nosso exemplo. Retorne para dentro do caixa de ferramentas (toolbox) e abra o GUI Builder e GUI canvas para o GUI que nós desenvolvemos. Então, com o nosso operador original ainda selecionado, selecione o menu option: Edit > Copy this GUI into Selected Objects. Retorne para dentro do model tree view e verifique o que foi feito com o atributo do operador guifocusclass.



Agora o atributo guifocusclass foi alterado para ">especial/guifocusclass/GUI 1". Também, uma cópia do GUI completo que nós criamos foi copiado para dentro do atributo guifocusclass

Embora você não necessitará usar a opção "Copy this GUI into Selected Objects" se você está somente usando este GUI para este modelo, ele é útil para propósitos de portabilidade. Uma vez que você criou este GUI e copiou para dentro do objeto, você pode adicionar o objeto para uma biblioteca de usuário, e então arrastá-lo para dentro de outros modelos, e o GUI será criado com o objeto.

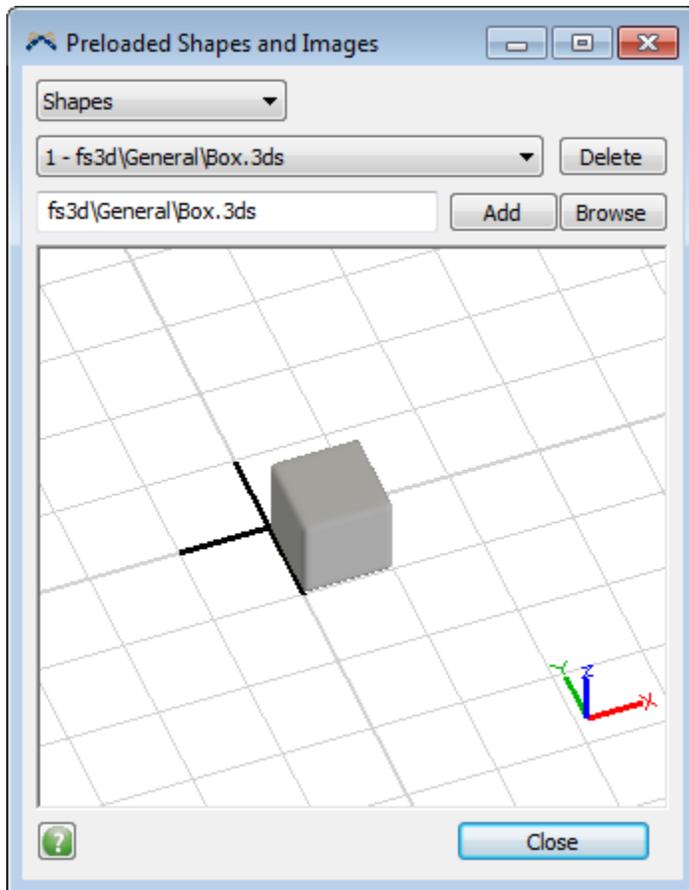
Add GUI to User Library - Esta opção irá adicionar o GUI para a biblioteca de usuário na grade de biblioteca de objetos (library icon grid). Para mais informações sobre biblioteca de usuários (user libraries), veja a documentação pertinente.

Assign this GUI to the "Control" button - Esta opção faz com o que o botão "Control" na barra de ferramentas do FlexSim abra o GUI quando pressionado. Este botão é chamado Mode Control GUI. Seu objetivo principal é permitir aos modeladores definirem GUIs personalizados para controlar os modelos e suas propriedades sem ter que alterar o view tree do FlexSim. Se este botão não estiver mais disponível, você pode adicioná-lo para sua barra de ferramentas através da janela Global Preferences.

Default GUIs - Este sub-menu permite a você também editar como outros botões na barra de ferramentas do FlexSim operam. O botão 3D View na barra de ferramentas pode ser customizado para abrir os GUIs customizados que você criou. Ao selecionar "Make this GUI the Default 3D View GUI" você pode fazer com que os botão 3D Views da barra de ferramentas abram seu GUI. Selecione "Reset Default 3D View GUI to Original" para fazer com que o botão 3D reverta para o seu GUI original. Note que fazendo um GUI customizado para o botão 3D View reseta para sua visualização padrão original.

Grid - Este sub-menu permite a você configurar o que o GUI canvas encaixa para o grid (superfície de simulação) e que tamanho é aquele grid.

Media Files



Media Files podem ser acessadas do menu Tools > **Media Files**

Este editor permite a você adicionar formas em 3D e imagens para serem pré-carregadas para o modelo, assim como dar o caminho string e index para formas/imagens que já foram carregadas. Alternativamente, você pode importar formas em 3D ou texturas (images) através da janela de propriedades de um objeto ou na janela Quick Properties. Formas referencias através destes métodos serão automaticamente importadas para dentro do modelo como uma forma pré-carregada.

O número próximo a cada opção na listagem drop-down pode também ser usado se você está escrevendo código que referencia uma textura ou shape index usando comandos como:

```
setobjectshapeindex(obj, 1);
```

```
setobjecttextureindex(obj, 1);
```

Shape indexes podem alterar quando as formas são adicionadas/removidas do modelo. Para conseguir o atual shape index para um 3D shape use:

```
getshapeindex("fs3d\\General\\Box.3ds");
```

Para mais informações sobre a media 3D no FlexSim, veja a seção 3D Media deste manual.

Shapes – Alterna a janela entre mostrar uma forma em 3D e imagens.

Shape/Image Combobox - Mostra todas as formas carregadas atualmente ou imagens.

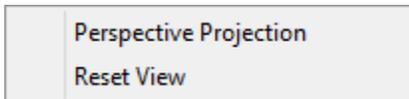
Delete - Deleta a forma atual selecionada ou imagem. Se a forma ou imagens está em uso, não será possível deletá-la.

Path - Mostra o caminho para forma 3D. Isto é também o nome da forma quando usado comandos como `getshapeindex()`.

Add - Adiciona a forma atual ou imagen como pré-carregada.

Browse - Permite a você buscar no Windows e selecionar uma imagem ou forma para importar. Isto irá somente mostrar o caminho para a forma/imagem no campo Path e mostra-la na view. Você deve pressionar o botão Add para adicionar a forma ou imagem como um pré-carregamento.

2D/3D View - Mostra a forma em 3D atualmente selecionado. Dê um duplo clique sobre o view para mostrar o seguinte menu:

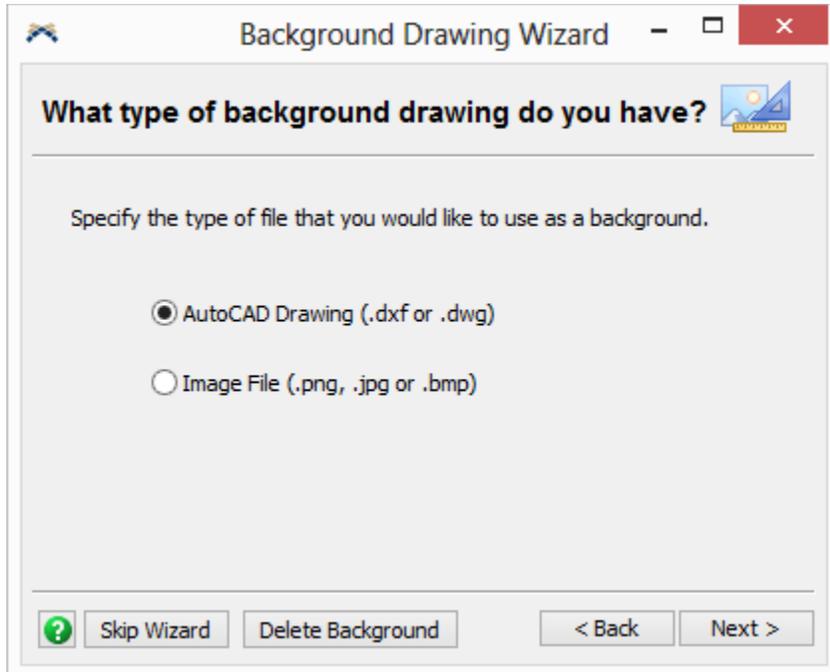


Perspective Projection - Altera a janela para ser orthographic ou perspectiva (veja Perspective View para mais informação).

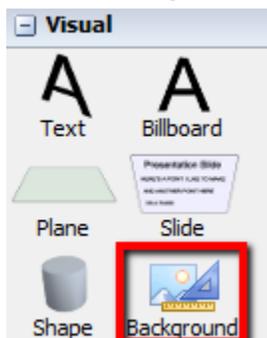
Reset View - Reseta a posição view e de rotação para 0,0,0.

Close – Fecha a janela.

Model Background (Importação da planta baixa)



O Model Background pode ser acessado do menu Tools > **Model Background** ou arrastando e soltando o objeto **Background** da seção Visual da grade de biblioteca de objetos (Library Icon Grid).

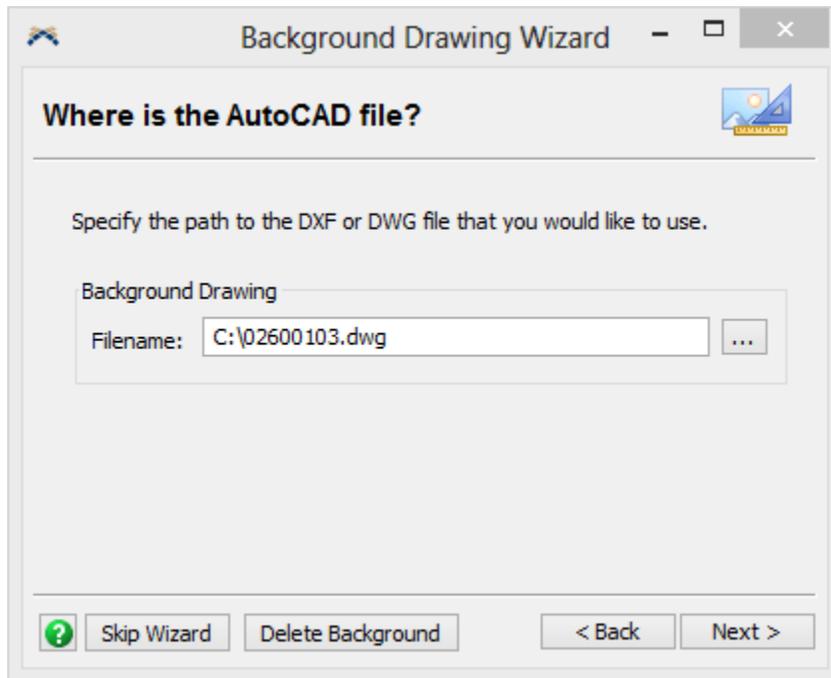


Nota: Arrastando e soltando um objeto background para dentro de um modelo já contendo uma planta baixa irá abrir a janela de propriedades do background criado anteriormente.

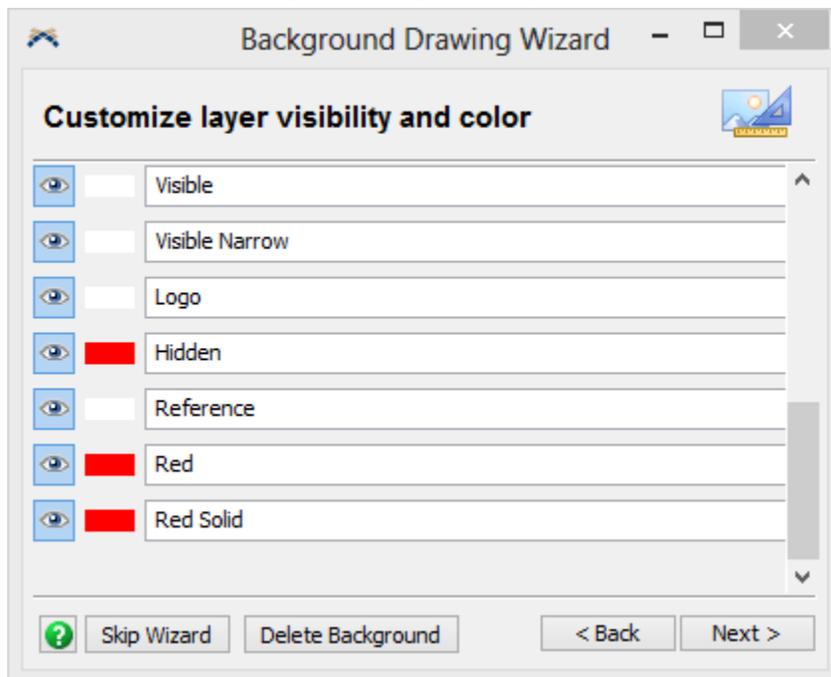
O Model Background é somente um objeto como uma Visual Tool e mostra uma imagen ou um arquivo dxf/dwg como um “piso” ou planta baixa do modelo.

Selecione o seu formato desejado (AutoCAD Drawing ou Image) e escolha o seu arquivo desejado.

Arquivos DWG



Arquivos DWG contém vários layers o quais podem ser customizados dentro do FlexSim. Uma vez selecionado o seu arquivo desejado, clique no botão Next para visualizar os layers disponíveis.

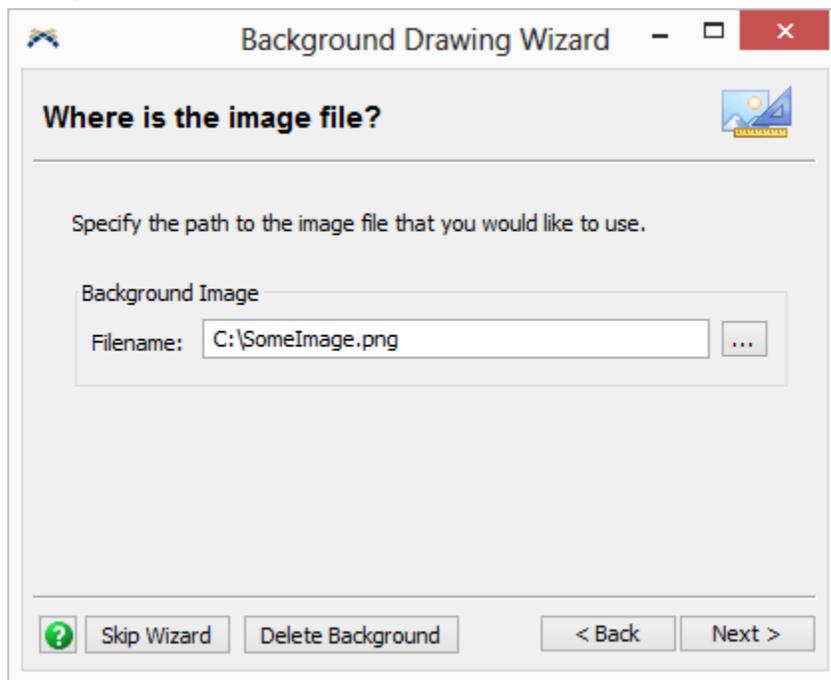


Você ativar ou desativar os layers ao clicar no botão . Próximo ao olho fica uma caixa de cores onde você pode definir a cor dos layers.

Arquivos DXF

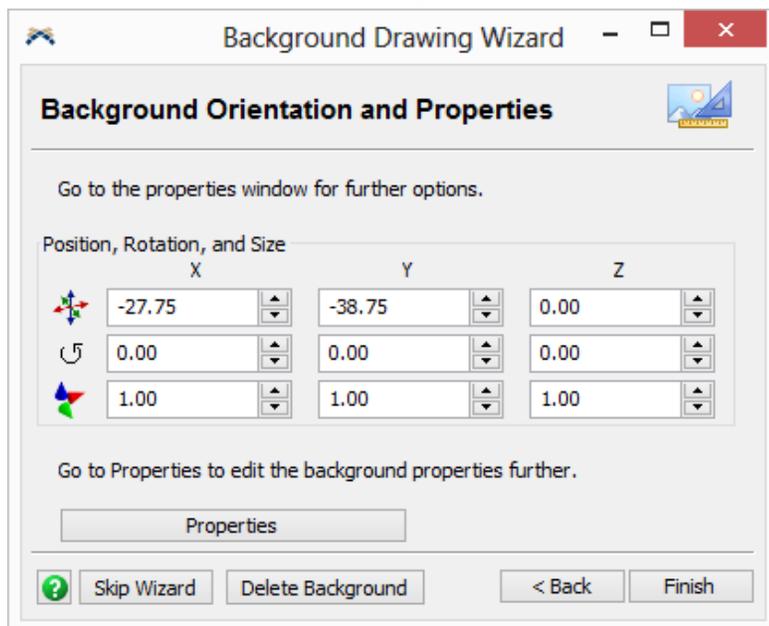
Selecione um arquivo DXF exatamente como você faria para um DWG. No entanto, não existe visibilidade do layer e cores para editar.

Image Files



Os arquivos de imagem podem ser png, jpg ou bmp.

Configurando a Posição, Tamanho e Rotação



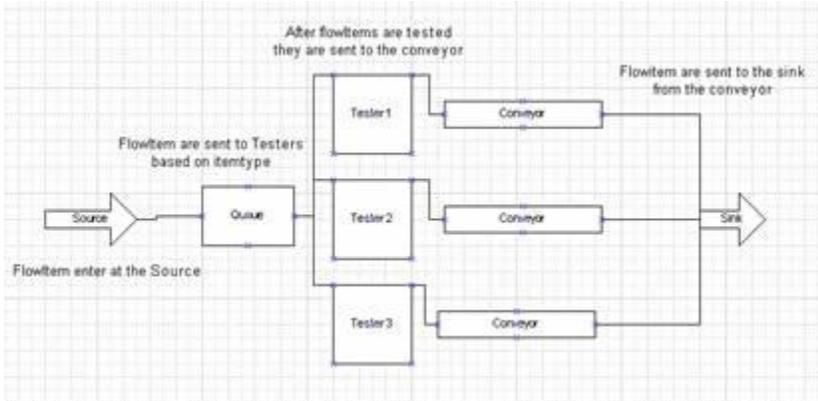
Alguns arquivos podem não ser importados com a correta posição, tamanho ou rotação. Isto pode ser facilmente corrigido através desta página final.

Arquivos DWG e DXF tem ambos unidades de medida associado com o arquivo. Quando você importa estes arquivos para dentro do FlexSim, você vai querer que estas unidades coincida com as unidades do seu modelo. Digamos que você está importando um arquivo DWG ou DXF que é o layout do chão de sua fábrica. Se a unidade do seu modelo do FlexSim estiver configurada em metros e o arquivo DWG ou DXF que te passaram estiver configurado em feet (pés), a imagem será importada em tamanho muito largo. Isto porque o FlexSim importa o modelo como uma unidade ratio 1-1. Então 1 foot (pé) é igual a 1 metro. Para corrigir seu arquivo DWG ou DXF para o tamanho correto, para este exemplo, você terá que configurar a escala do x, y, e z para 0.3048.

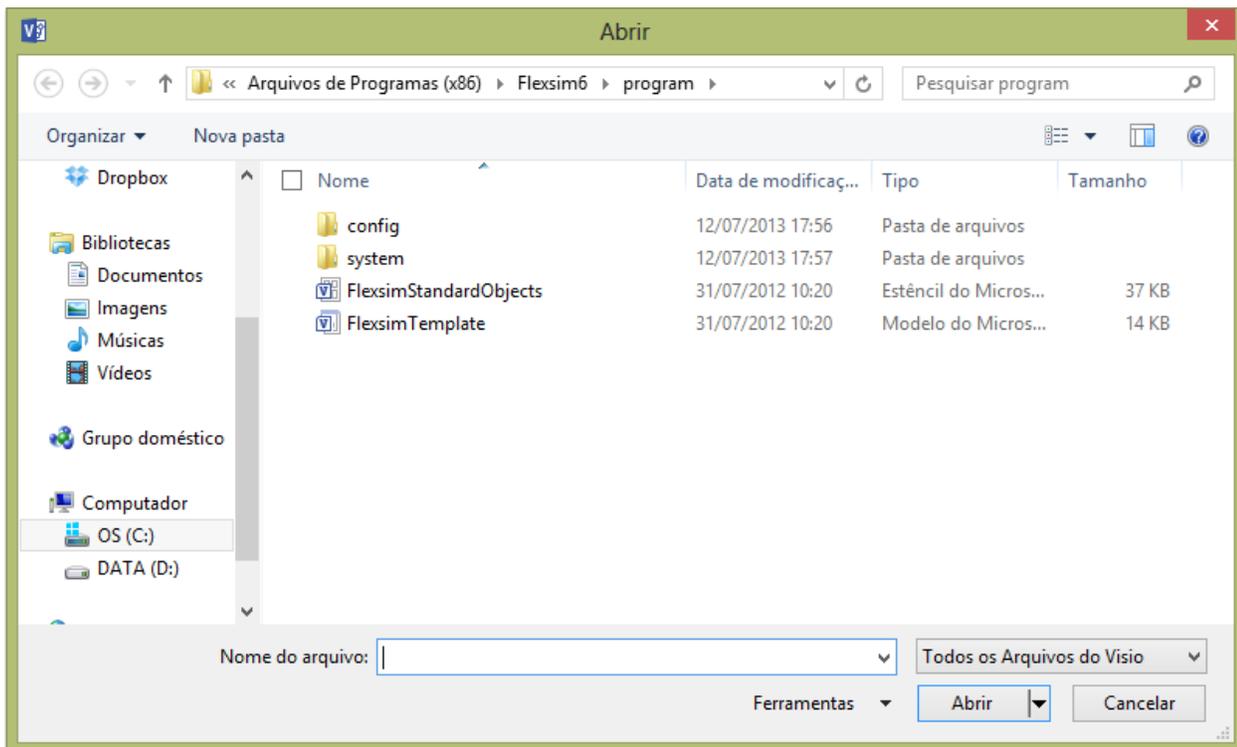
Importação do Microsoft Visio

A ferramenta de importação do visio permite você construir modelos no Microsoft Visio™ e importá-los para dentro do FlexSim. Abaixo, descrevemos como fazer isso.

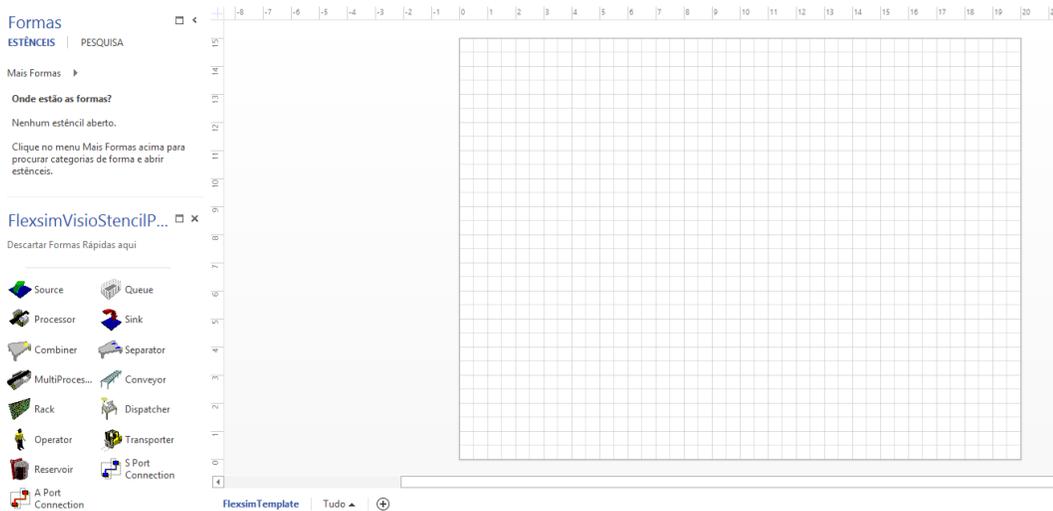
Inicie o Visio



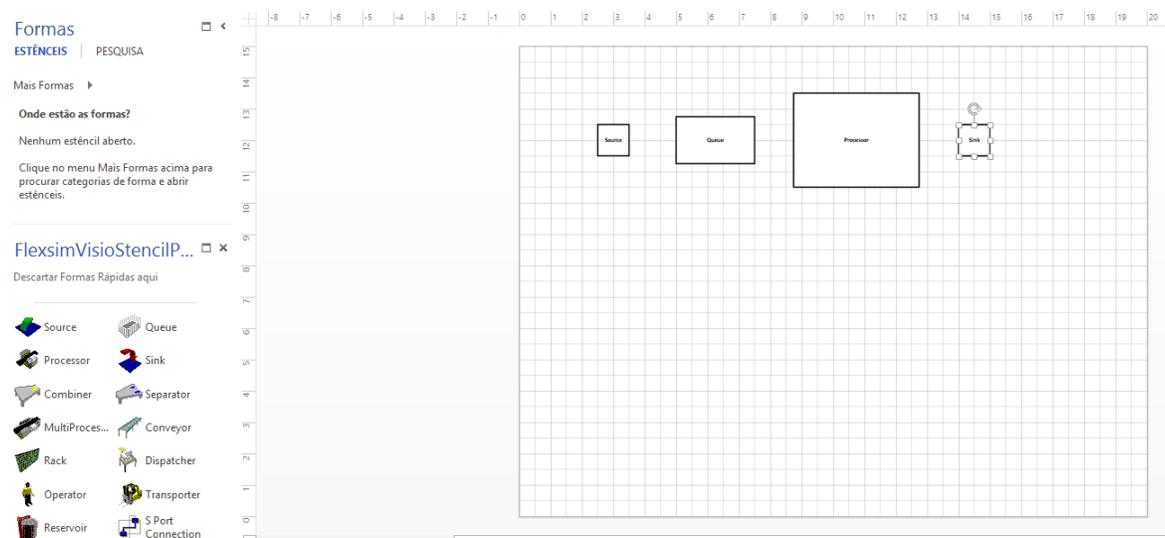
Abra o arquivo FlexSimTemplate.vst que deve estar na seguinte pasta abaixo, que está dentro da pasta do FlexSim6.



Visio deverá parecer-se com alguma coisa semelhante a tela abaixo.

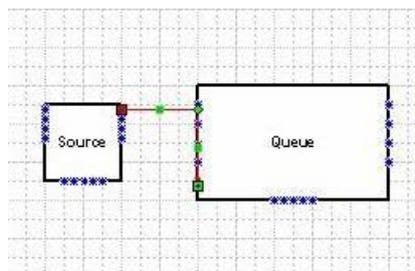


Arraste os objetos do Stencil para dentro da superfície do modelo para arrumar o layout.

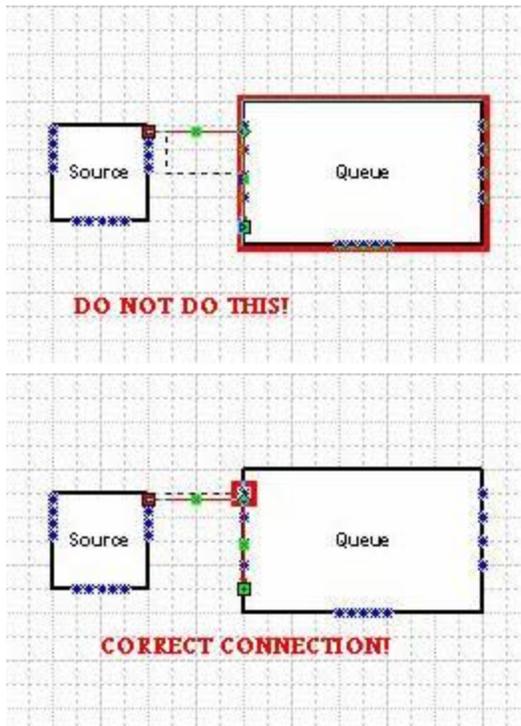


Use o teclado A para criar uma conexão entre os objetos arrastando no sentido do fluxo (porta de saída de um com porta de entrada de outro objeto)

A conexão é feita do ponto superior do lado esquerdo de um objeto para o ponto inferior do objeto que encontra-se do lado direito.. Esta é uma ligação que está ligado apenas ao objeto a montante

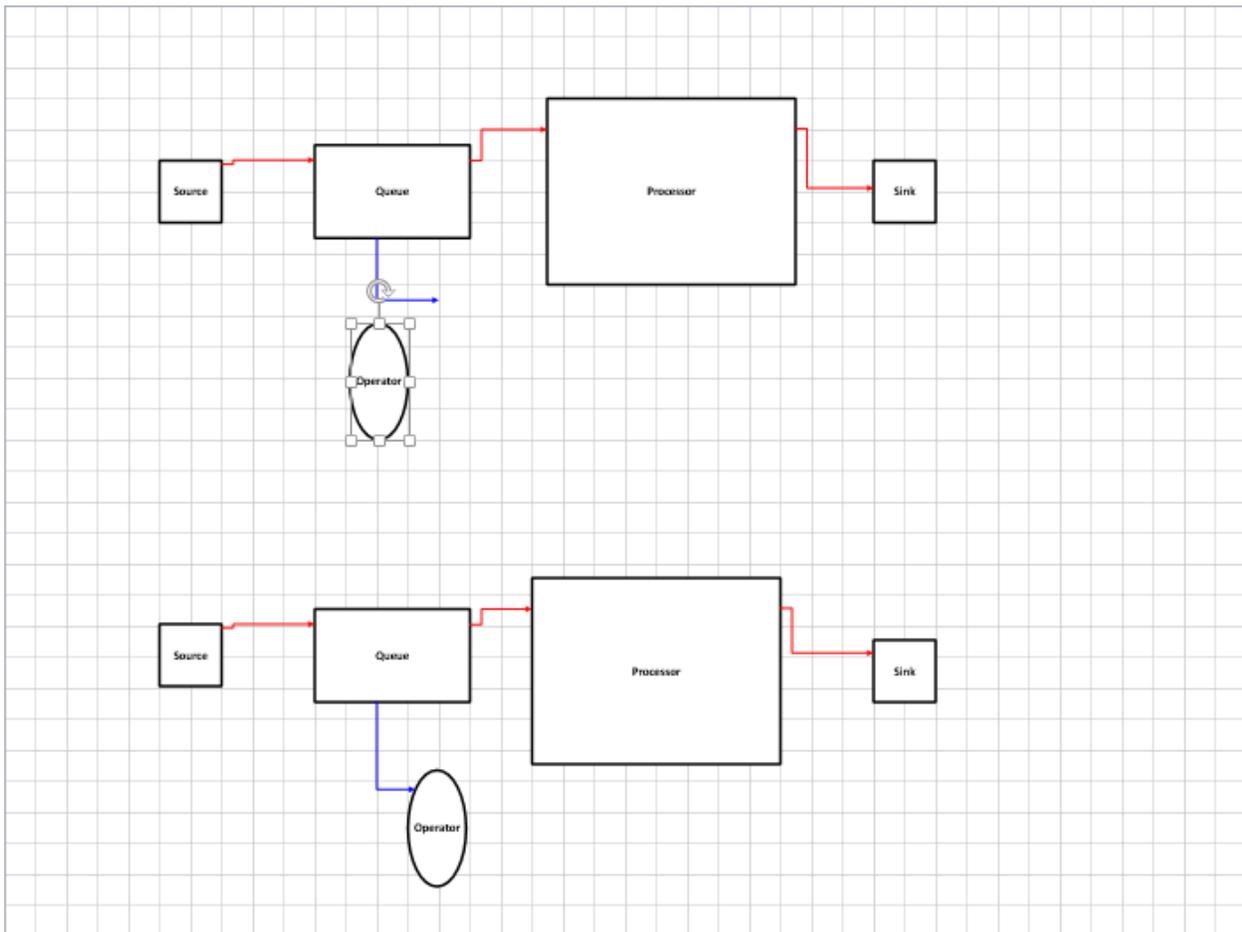


Nota: Não faça a conexão diretamente no objeto. Ele deve ser conectado a um dos pontos azuis.



Ao arrastar uma conexão, é aceitável ter ambas as extremidades tocando pontos azuis da conexão. Assim a conexão será criada corretamente. Além disso, o ponto de conexão que você conecta no Visio não importa. Conexões com Flexsim serão ordenadas de acordo com a ordem em que eles foram arrastados no Visio. O exemplo abaixo irá conectar o Queue à primeira porta de entrada do Processor, apesar de estar ligado ao segundo ponto de conexão do Processor no Visio.

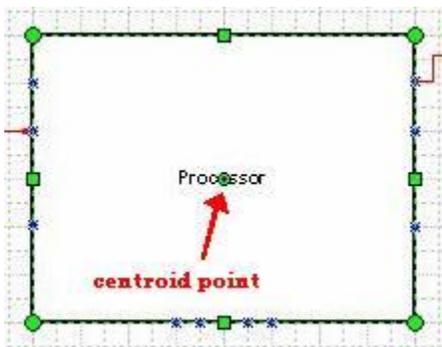
Conexões com o teclado S (portas centrais) são criados da mesma forma como as conexões de porta de entrada e saída. O ponto de conexão no Visio que eles estão ligados não se faz necessário ter uma ordem de conexão. Realizando as conexão de portas centrais com o teclado S nos pontos azuis dos lados esquerdo ou direito são aceitáveis.



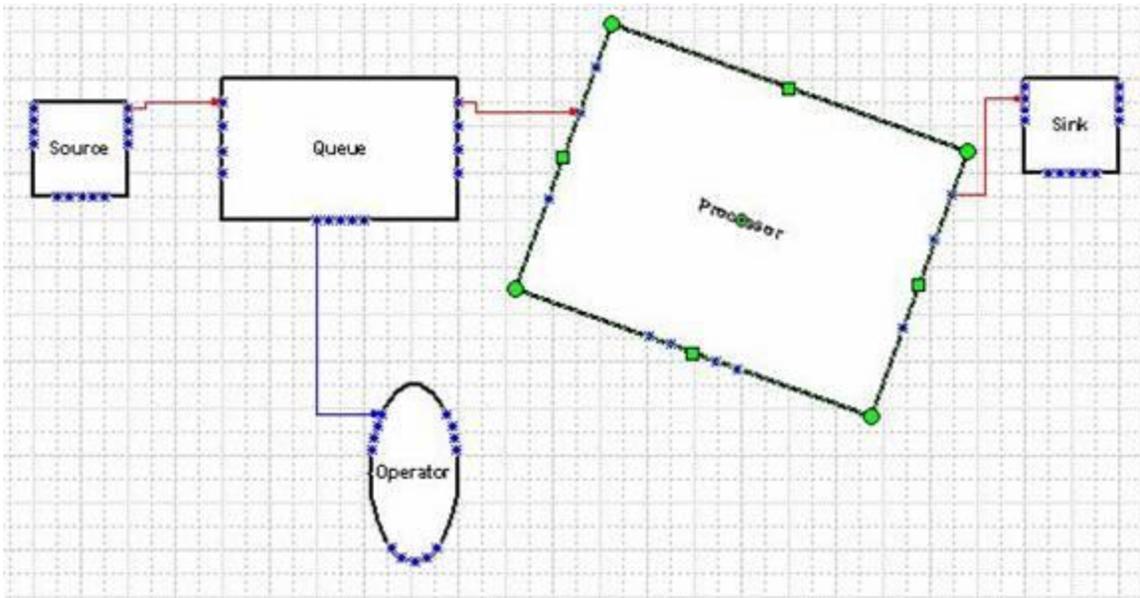
Os objetos podem ser rotacionados e redimensionados no Visio como uma forma para redimensionar os objetos no FlexSim.

A ferramenta de rotação do Visio é indicada pelo ícone ao lado. 

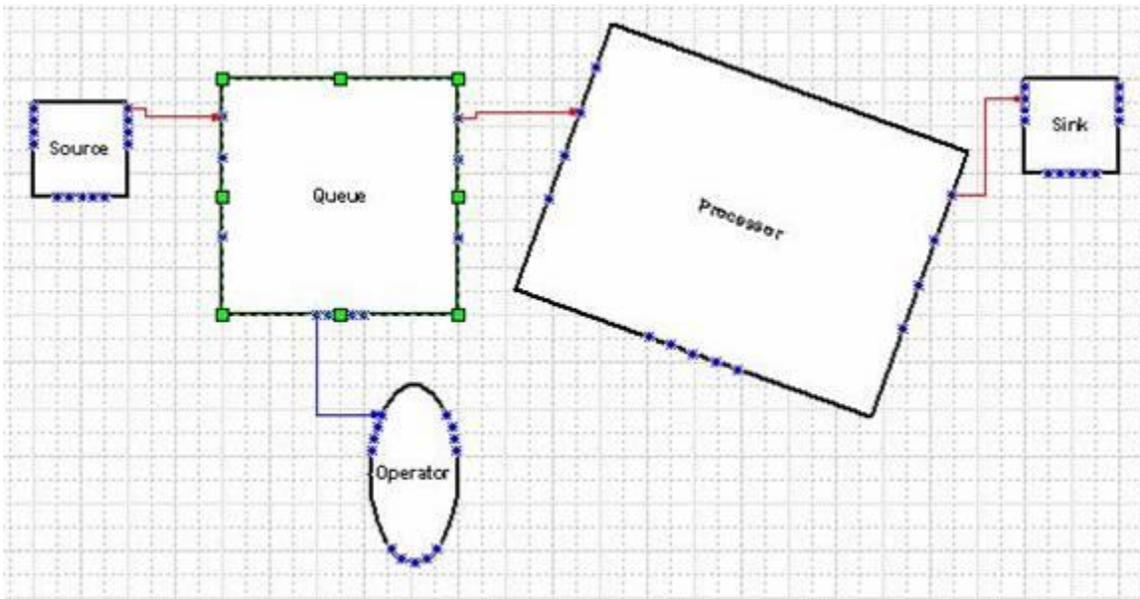
Não mova o ponto central do objeto quando estiver redimensionando-o. O ponto central é um círculo verde com um ponto preto em seu meio. Ele é usado para definir onde é o centro de rotação daquele objeto. Se você movê-lo no Visio, o objeto não será localizado corretamente no Flexsim.



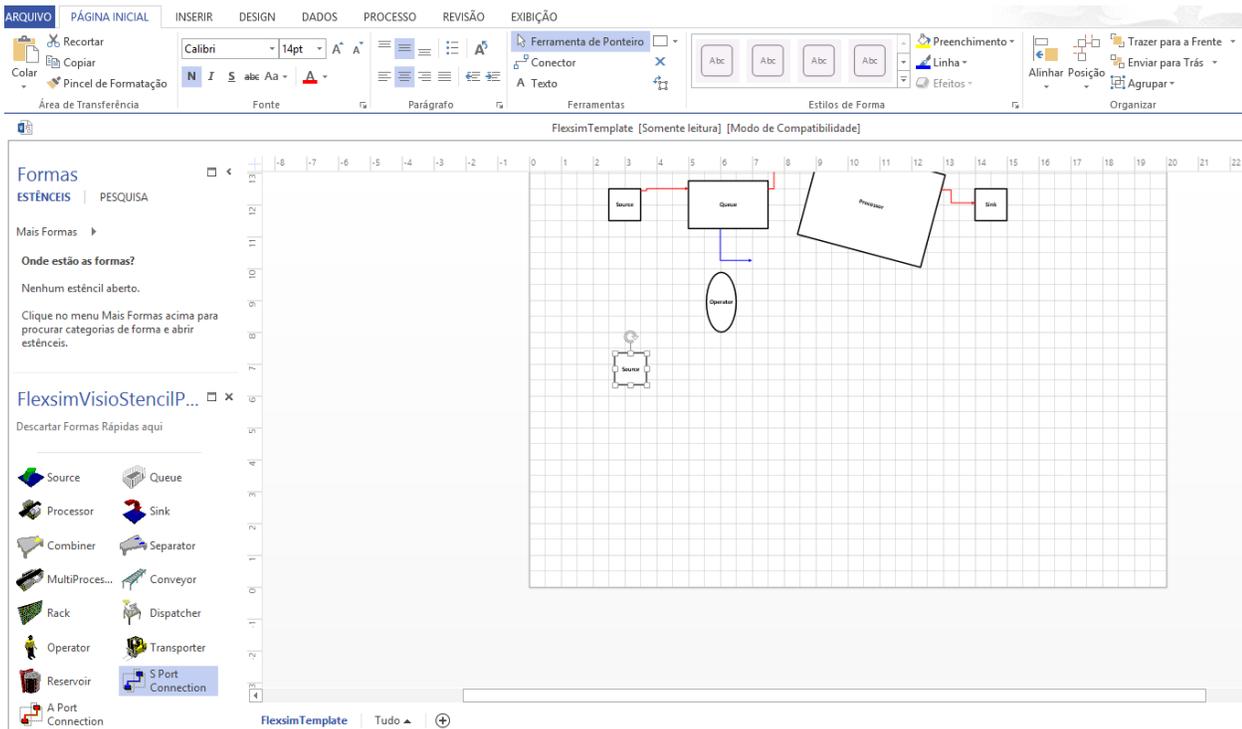
Este processador foi rotacionado dando um clique sobre ele e arrastando a ferramenta de rotação  ao redor de um dos pontos (centrais ou de canto).



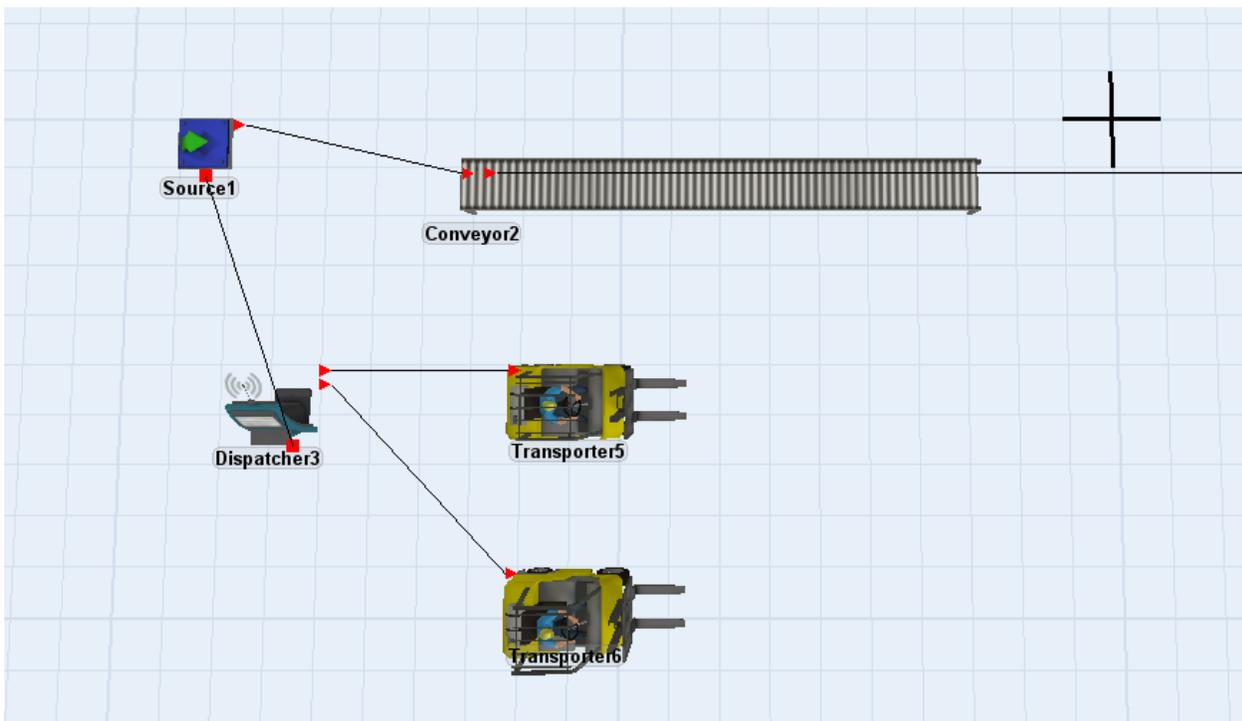
Este queue foi redimensionado ao arrastar um dos pontos laterais para baixo usando a ferramenta de seleção.  .



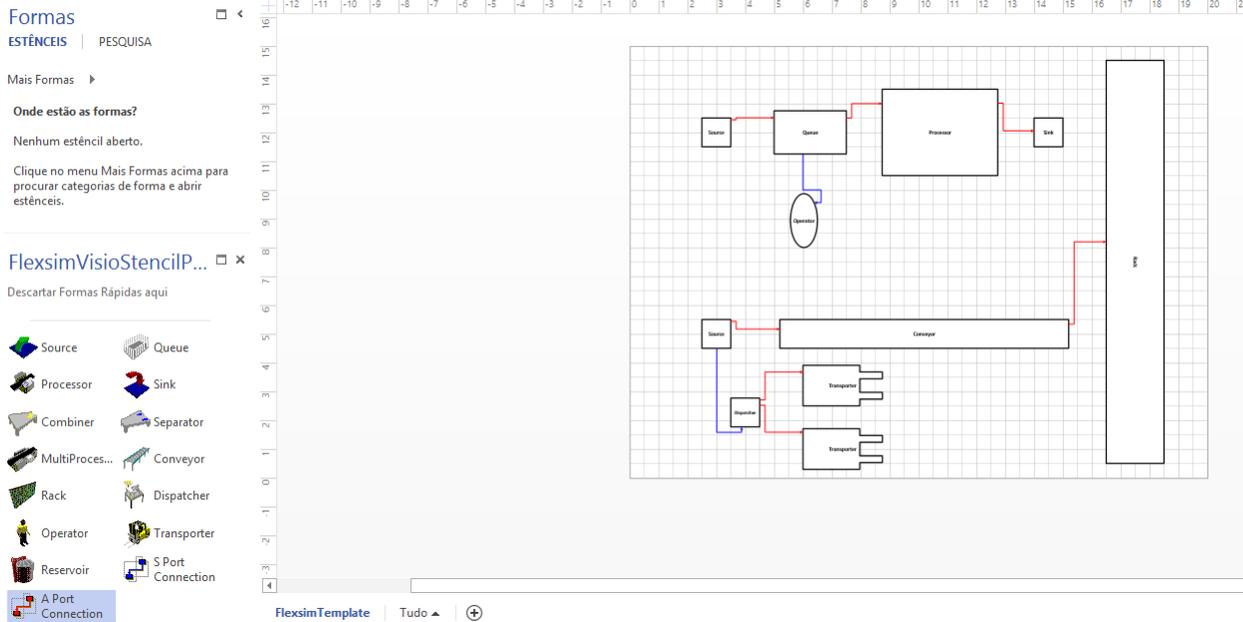
Objetos podem ser renomeados ao dar um duplo clique sobre um objeto e digitando um novo nome.



Cada objeto em seu modelo deve ter um nome único para ser importado corretamente para dentro do FlexSim. Se os objetos tiverem o mesmo nome no Visio, suas conexões não serão criadas corretamente no FlexSim. No exemplo abaixo, os Transporters tem nomes diferentes, então o Dispatcher foi conectado uma vez a cada um dos Transporter. Se os Transporters tivessem o mesmo nome, o Dispatcher seria conectado ao primeiro Transporter duplamente, quando importado do Visio para o Flexsim.

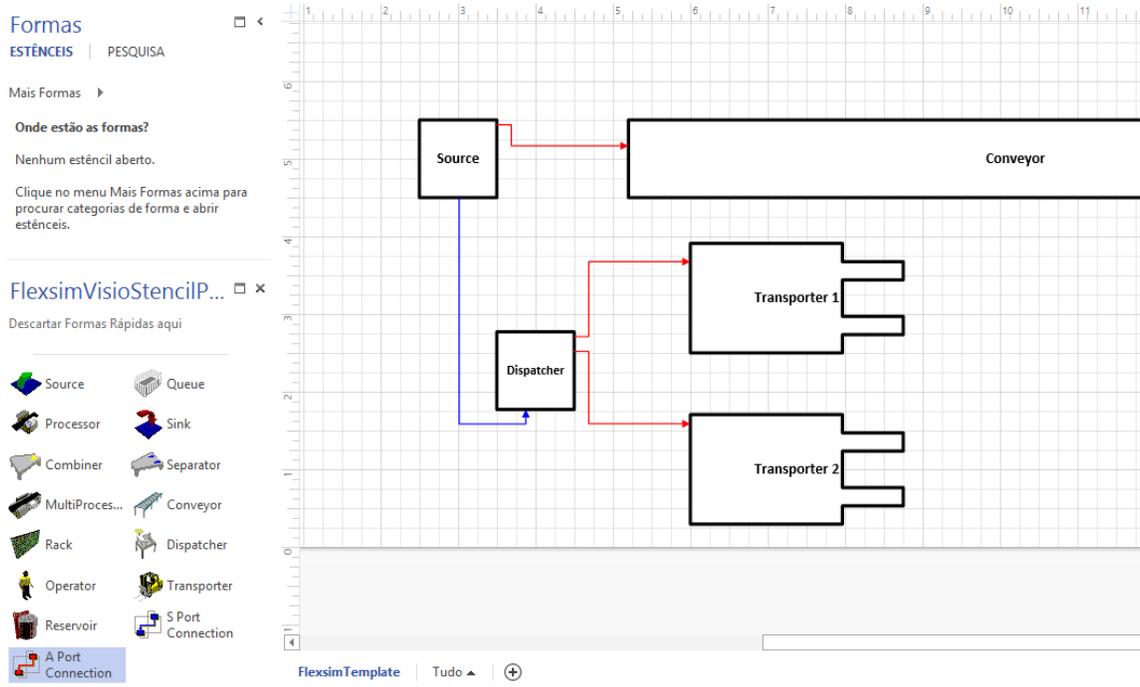


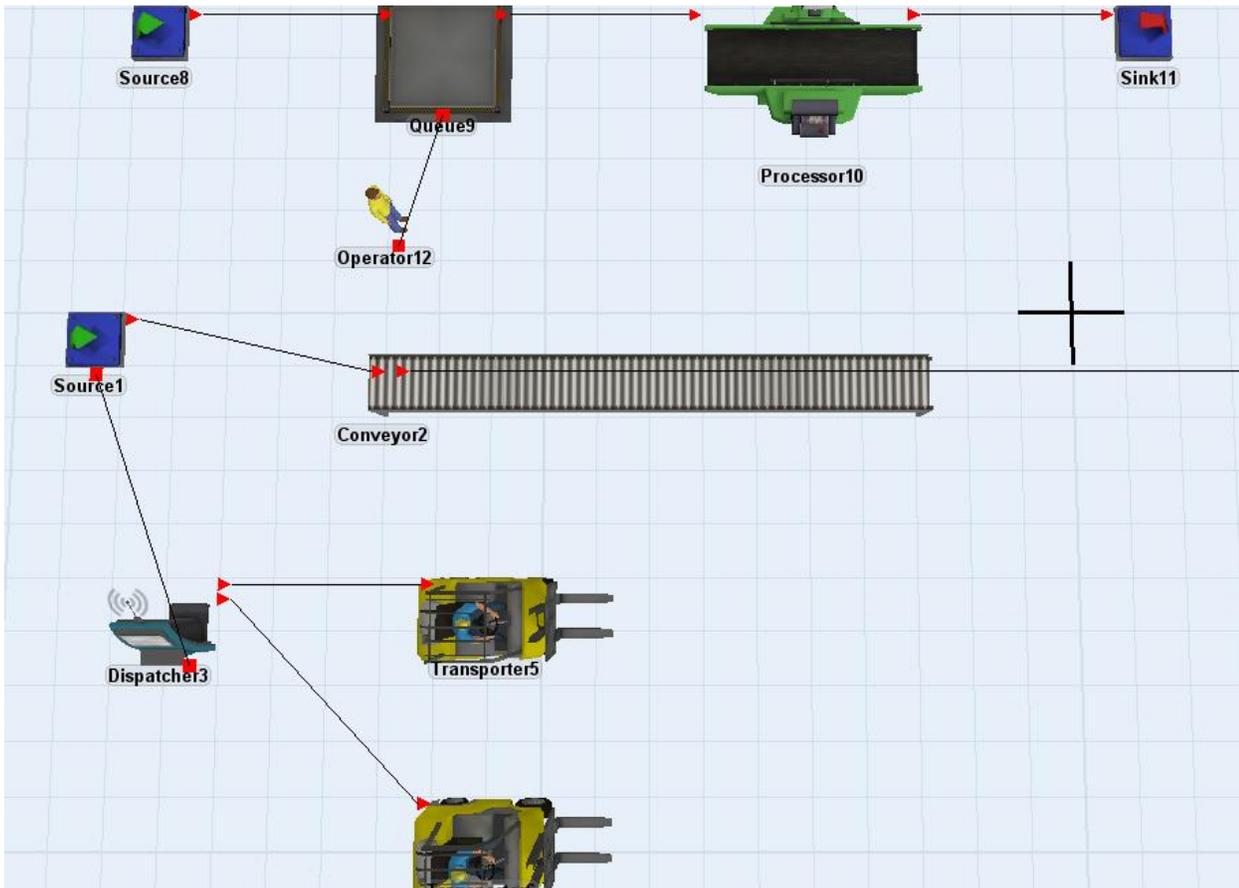
Aqui é um preview de como ficaria o fluxograma do Visio dentro do FlexSim.



Existem algumas coisas que precisam ser feitas para assegurar-se de que tudo está correto antes de exportar para o Excel e depois importar para o Flexsim.

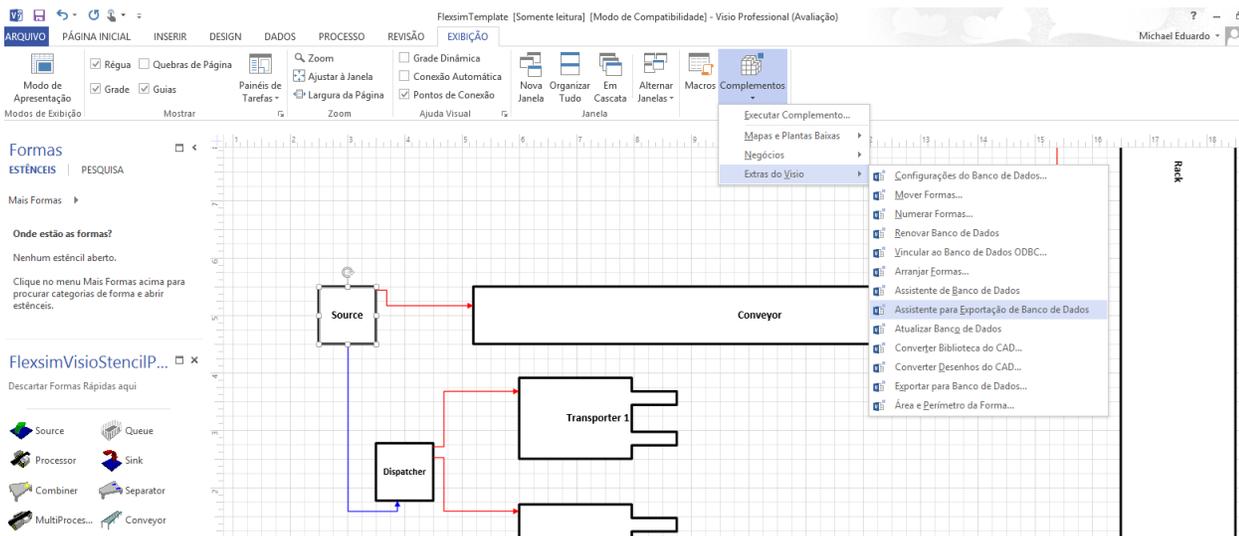
Assegure-se de que cada objeto tenha um único nome e que estejam corretamente formatado.





Agora que todos tem nomes diferentes, estamos prontos para exportá-los.

Acesse no menu Exibição - Complementos > Extras do Visio > Assistente para Exportação de Banco de Dados.



A janela abaixo irá aparecer, clique em Avançar.



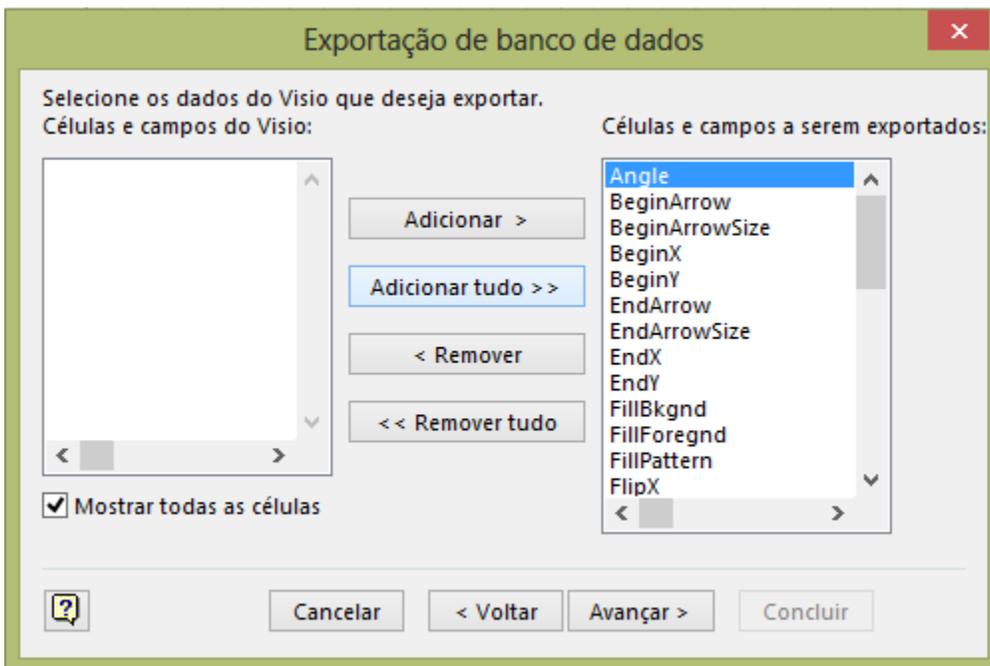
Selecione Avançar.



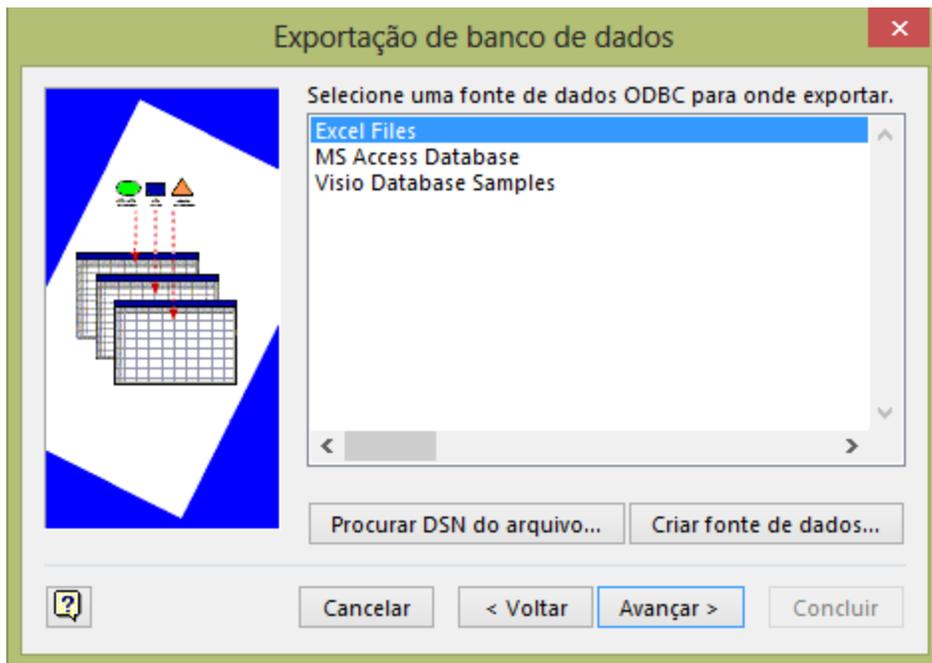
Encontre o local onde você salvou o seu arquivo do Visio e então clique em Avançar.



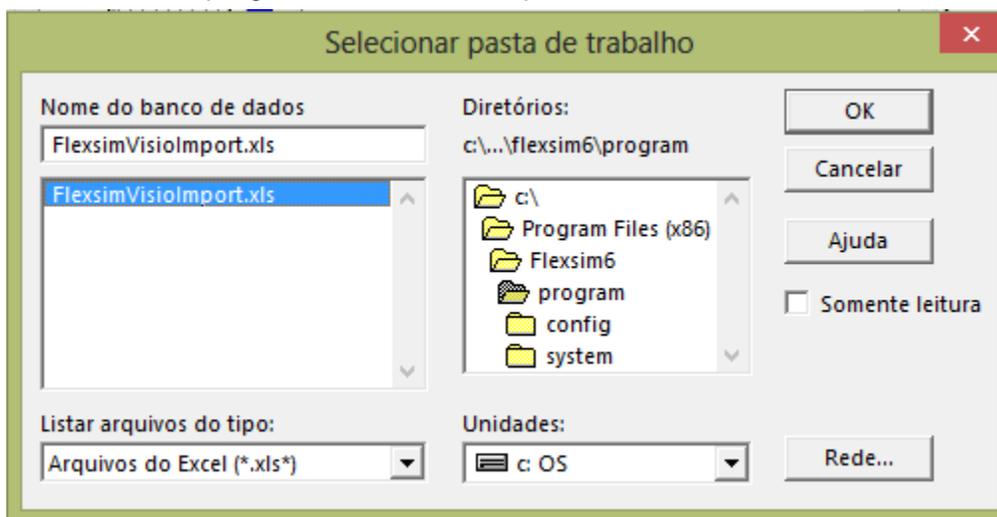
Clique em Avançar na tela acima



Selecione Adicionar Tudo e clique em Avançar na tela acima. Então, selecione o arquivo Excel como fonte de dados e então clique em Avançar.



Selecione o arquivo Excel que deve ser exportado. Deve estar em C:/Program Files/FlexSim6/program/FlexSimVisioImport.xls.



Digite o nome na tabela de nomes. Este nome será digitado novamente dentro do FlexSim para saber de qual planilha deve ser importado. Lembre-se o nome que você digitou e clique em Avançar.

Exportação de banco de dados

Especifique os detalhes da tabela de exportação:

Banco de dados: C:\Program Files (x86)\Flexsim6\program\FlexsimVisioImport.

Nome da tabela: TypeName

Detalhes da tabela

Campo de chave: ShapeKey

Tipo de chave: ShapeID

Tornar o campo de chave a chave primária da tabela

Cancel **Avançar >** Concluir

Não altere nenhum valor para os dados da tela e então clique em Avançar.

Exportação de banco de dados

Especifique os detalhes do mapeamento de exportação:

Dados do Visio: Exportar detalhes de mapeamento

Avaliar dados como: Fórmula

Nome do campo: Width

Tipo de campo: VARCHAR

Tamanho do campo:

Decimal do campo:

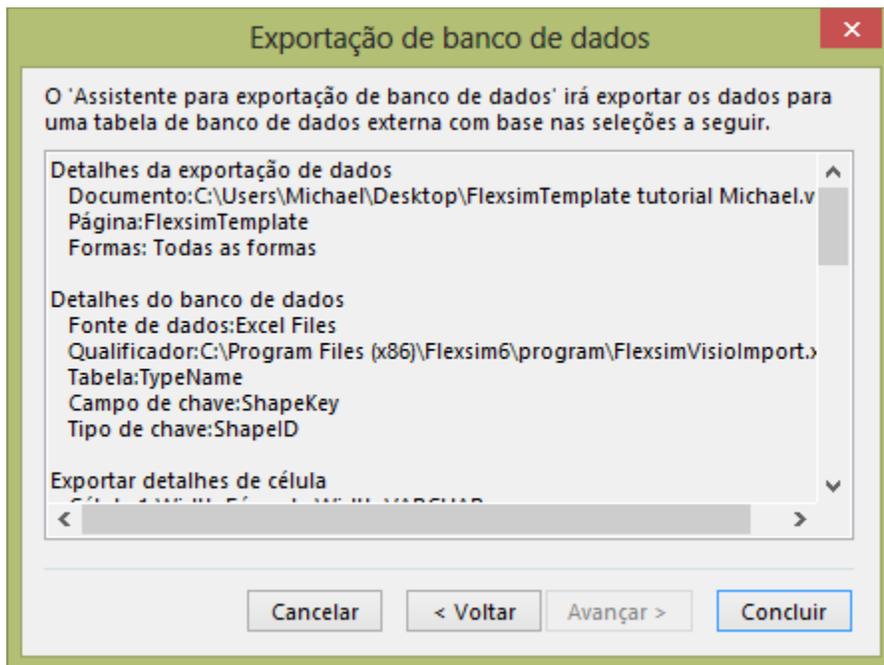
Width

Cancel **Avançar >** Concluir

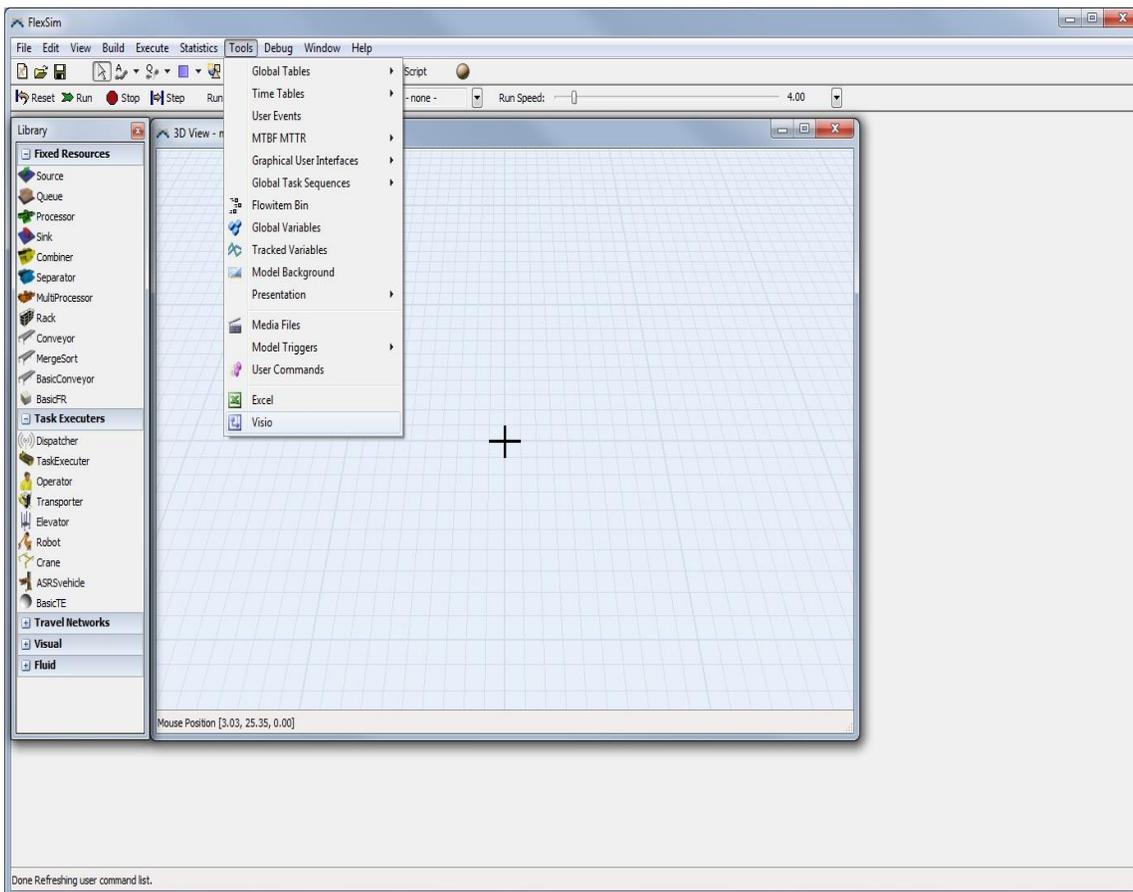
(Avançar).



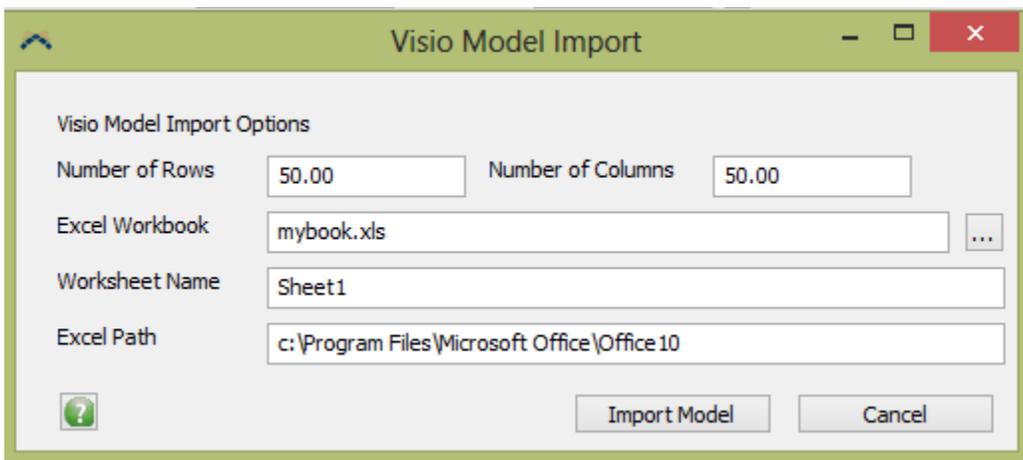
Clique em Concluir.



Agora, no FlexSim vá para Tools > Visio...

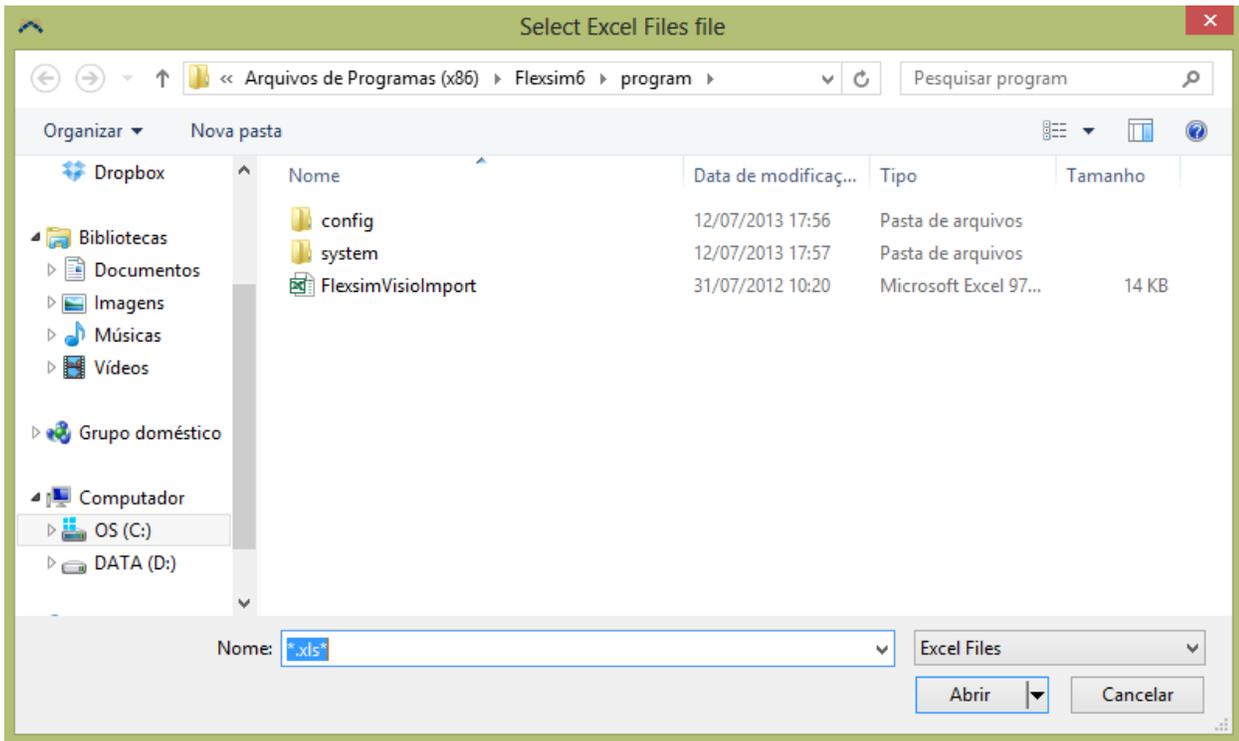


Esta caixa de diálogo irá aparecer.

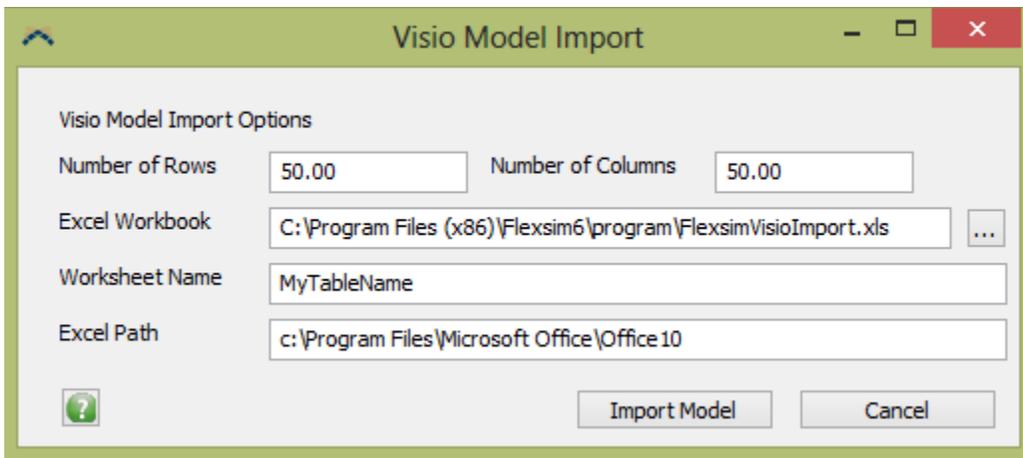


Entre com um número dentro de “Number of Rows e Number of Columns, que seja maior que o número de linhas e colunas na planilha excel que foi criada pelo Visio Database Export. Usualmente 50 é um número suficiente. Se o seu modelo não importar corretamente, você precisará aumentar este valor.

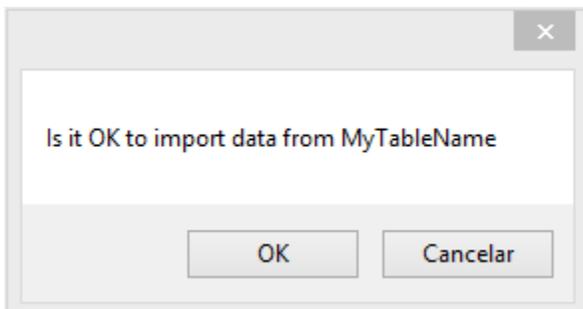
Clique no botão ... para procurar o arquivo excel correto e abri-lo.



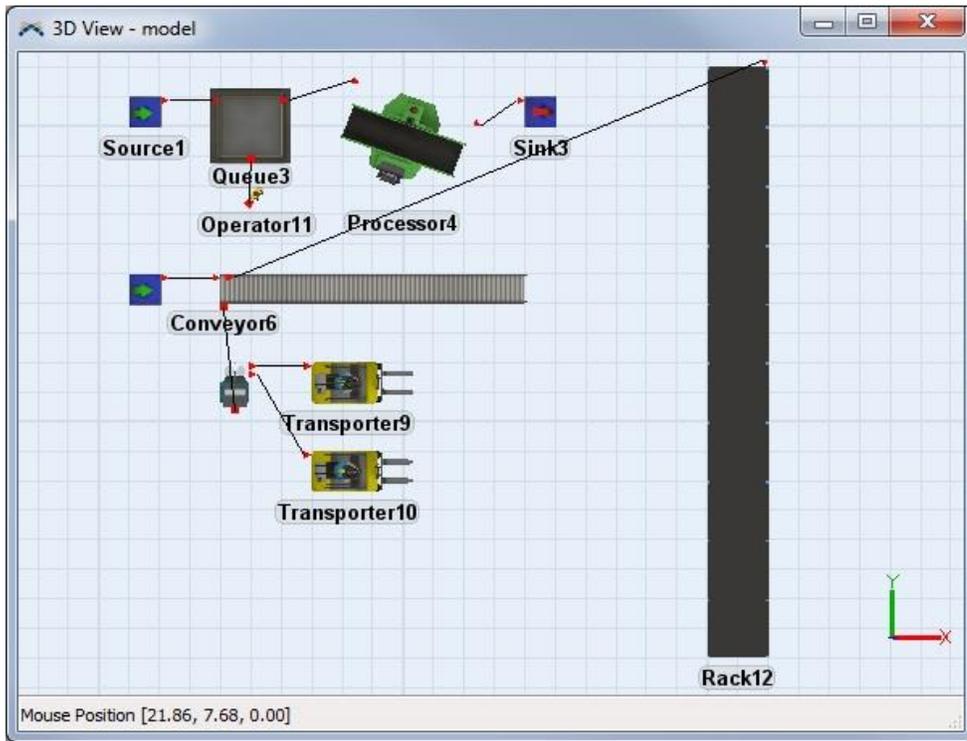
Digite o nome da tabela que foi alimentado e digitado anteriormente no campo “Worksheet Name”.



.Assegure-se que o “Excel Path” foi especificado corretamente.
Clique em “Import Model”. Esta mensagem abaixo irá aparecer.



Aguarde até o Excel abrir corretamente a planilha e então clique em Ok.
O modelo será então importado. Isto pode levar alguns minutos. Aguarde até a janela de importação do modelo do Visio ser fechada. Seu modelo será então importado para dentro do FlexSim.



3D Media

1. Importing 3D Media
2. Preparing a 3D File
3. Importing AutoCAD Drawings
4. Shape Factors
5. Shape Frames
6. Level Of Detail

Importando Media 3D

FlexSim pode importar diversas media 3D media. Estas incluem:

- .wrl, .3ds, .dxf, .stl, .skp, .dae, .obj, .ac, .x, .ase, .ply, .ms3d, .cob, .md5mesh, .irr, .irrmesh, .ter, .lxo, .csm, .scn, .q3o, .q3s, .raw, .off, .mdl, .hmp, .scn, .xgl, .zgl, .lvo, .lvs, .blend

Nota sobre importação de arquivos wrl: FlexSim irá importar apenas formatos VRML na versão 1.0, não na versão 2.0.

Nota sobre importação de arquivos stl: FlexSim irá importar apenas arquivos stl ascii, não arquivos stl binary.

Você pode importar media usando três métodos:

1. O método mais comum de importar media ao selecionar o arquivo 3D através da guia de propriedades gerais do objeto na janela Quick Properties.
2. Através da guia General do objeto.
3. Se existem alguns arquivos de media 3D que não estão sendo usadas por nenhum dos objetos no modelo por padrão, mas você quer alterar sua representação 3D dinamicamente durante a execução da simulação para usar estas medias 3D, você pode usar a janela Media Files do menu Tools para importar arquivos de media explicitamente.

As informações seguintes irão ajudar que as medias sejam importadas corretamente:

- **Shape Factors**
- **Letting Color Show Through the 3D Shape**
- **Level of Detail**

Preparando um Arquivo 3D

Nós sugerimos usar o AC3D para criação de media em 3D. Uma licença é relativamente barata e o AC3D tem muitos recursos e vem acoplado com uma instrução ao usuário que é muito fácil e rápido de aprender. Outra grande vantagem é que o Flexsim é capaz de importar diretamente os arquivos de extensão .ac criados do AC3D. Você pode pegar os arquivos AC3D do site www.inivis.com

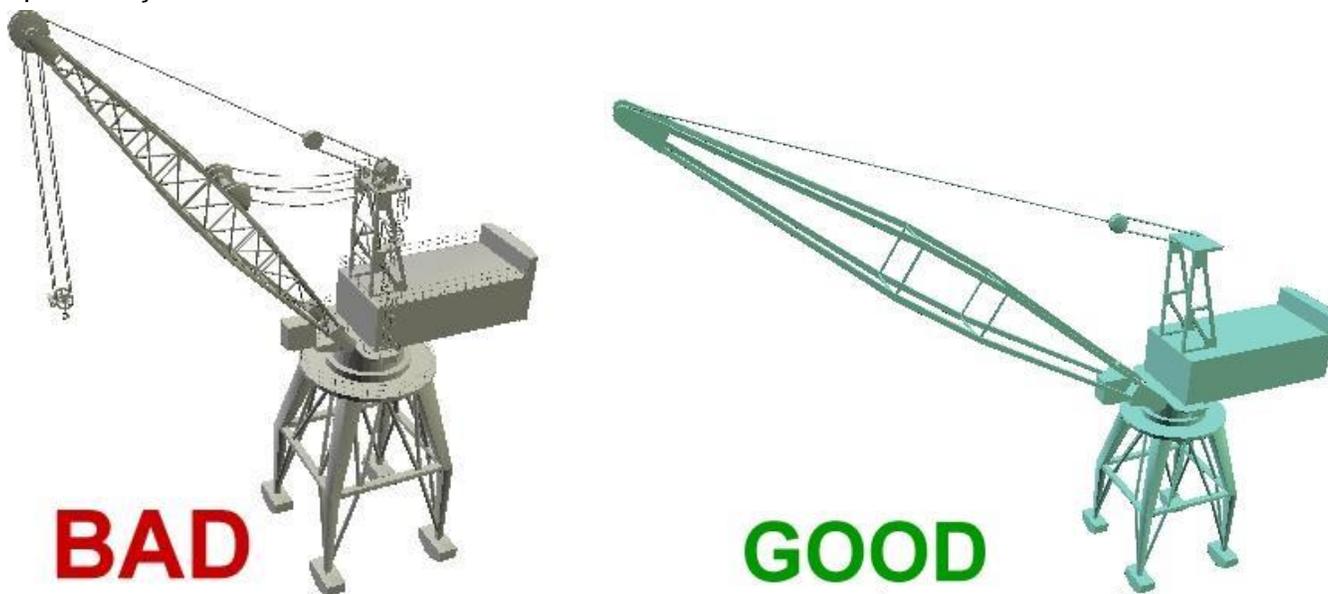
Se você não estiver usando o AC3D, nós sugerimos você exportar a media para o formato .3ds. O arquivo em formato .3ds permite a importação bem mais rápida. Com diversos formatos 3D, incluindo arquivos .3ds e .ac, você pode:

- Ter as cores dos objetos do Flexsim visualizadas através de certas partes do shape (forma)
- Ver a textura definida dos objetos do Flexsim mostradas em certas partes através de seu shape (forma)
- Auto-redimensionar os objetos do Flexsim baseado nas dimensões 3D do arquivo.

Os seguintes passos devem ser realizados quando estiver preparando um arquivo 3D para importação:

Reduzindo o número de polígonos

3ds e wrl files tipicamente incluem mais informações que o necessário. Removendo excessos de polígonos irá aumentar a performance visual do seu modelo. Como resultado, isso facilitará a construção e apresentação do modelo.



Texturas vs Polígonos

O uso de texturas bem feitas pode ajudar um modelador a reduzir o número de polígonos necessários para tornar-se um objeto do ponto de vista mais realístico. Guiaixo estão algumas fotos de baixa resolução do polígono 3D no Flexsim. Além disso, sempre que possível, você deve consolidar objetos em seu software de criação 3D e compartilhar a mesma textura entre vários objetos. Isso reduz o número de mudanças de estado OpenGL necessários, o que pode melhorar significativamente o desempenho.



Ajuste de escala

Os arquivos 3D não necessariamente precisam ser desenhados em feet ou metros e pode ser necessário ser redimensionado para trabalhar adequadamente no FlexSim. Existem três formas de ajustar a escala de um arquivo em 3D:

1. Dimensionar adequadamente o arquivo em um programa 3D
2. Dimensionar a ferramenta visual (Visual Tool) ou qualquer outro objeto de onde o arquivo em 3D será importado.
3. Use um arquivo xrl para arquivos wrl.

Se você está exportando para o 3DS, então você pode verificar se o objeto está dimensionado adequadamente no AC3D ou qualquer outro pacote que você utilizar, então importar para dentro do FlexSim, e ele automaticamente será importado no tamanho correto. Apenas assegure-se de que as unidades no AC3D são as mesmas unidades usadas no FlexSim.

Dando uma tonalidade de cor especial aos objetos

Para arquivos 3DS, você pode dar uma tonalidade de cor especial aos objetos do FlexSim sobre certas formas dos arquivos 3DS. Para fazer isso, dentro do AC3D dê ao objeto um ambiente de cor no valor `rgb:(0.235,0.235,0.243)`. Além disso, você precisa ter certeza de o objeto é exportado como o primeiro na definição de arquivo 3DS.

Dando uma textura especial aos objetos

Para arquivos 3DS, você pode dar uma textura especial aos objetos do FlexSim ao invés de utilizar a textura padrão dos arquivos 3DS. Para fazer isso, apenas adicione uma textura chamada "fstx.png" para o objeto no AC3D. Novamente, assegure-se de que o objeto foi nomeado alfabeticamente antes dos demais objetos no modelo.

Arquivos XRL

Arquivos xrl são usados para fazer importação de formas wlr conforme o objeto de onde ele está sendo importado. Arquivos .xrl devem ter os mesmos nomes que os objetos que eles modificaram:

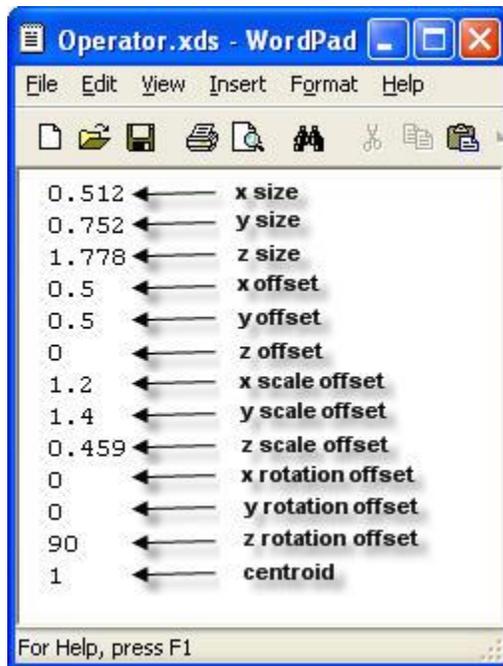
- crane.wrl – crane.yrl

Informação arquivo .xrl

Valores espaciais determinam o tamanho final da forma 3D. Valores de deslocamento são os valores necessários para obter a forma 3D zerado e dimensionado para 1,1,1. O valor baricentro é 1 ou 0 e determina se o objeto gira em torno do centro do objeto ou do canto superior esquerdo.

xrl file makeup

O arquivo .xrl é um arquivo texto composto de 13 valores separados por carriage returns. Você pode editar este arquivo usando o bloco de notas ou wordpad, como mostrado guiaixo



xrl file info

Valores espaciais determina o tamanho final da forma 3D. Os valores Offset são os valores requeridos para pegar a forma 3D zeroed e com tamanho 1,1,1. O valor centroid é 1 ou 0 e determina se o objeto rotaciona ao redor do centro do objeto ou canto esquerdo superior.

Importando arquivos do AutoCAD

Os seguintes passos devem ser realizados quando estiver preparando o arquivo .dxf do AUTOCAD para importação:

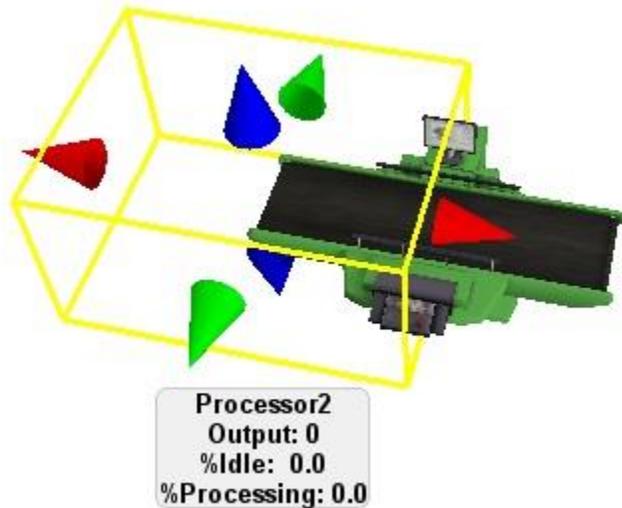
1. **Remova toda informação desnecessária:** Arquivos do AutoCAD normalmente incluem muita informação que é desnecessário para a simulação. Normalmente, toda a necessidade de uma simulação é um layout básico. Remoção de informação que é irrelevante para a simulação vai fazer o seu modelo ficar mais claro e reduzir a carga sobre a sua placa de vídeo. Como resultado, será mais fácil de construir e apresentar o modelo. Remova todas as partes do desenho que não são pertinentes para o estudo de simulação.
2. **Ajuste a escala de unidades do FlexSim:** Arquivos do AutoCAD são frequentemente configurados em polegadas. Os modelos do Flexsim são frequentemente configurados em pés ou metros. É importante que o arquivo AutoCAD esteja adaptado para funcionar adequadamente no Flexsim. Por exemplo, para converter um arquivo do AutoCAD em polegadas para um modelo do Flexsim em pés, o fator de escala será 1/12. Para determinar a quantidade de escala, siga estes passos: meça a distância conhecida em AutoCAD ("_dist"). Aplique a seguinte equação: fator de escala = Flexsim distância / ACAD distância. Para dimensionar objetos no AutoCAD siga estes passos: Selecione os objetos que deseja dimensionar. Digite Type "_scale" no command prompt ou selecione o scale comanda a partir dos menus. Especifique um ponto de referência. Digite o fator de escala calculado no command prompt.
3. **Movimente os objetos para origem:** As imagens em AutoCAD são geralmente desenhados usando um sistema de coordenadas específico. Isso geralmente significa que os objetos não estão localizados perto da origem (0,0,0). Quando um arquivo dxf. é importados para o Flexsim, ele está posicionado em um sistema de coordenadas do Flexsim de acordo com o posicionamento para o arquivo .dxf. Portanto, se o ponto de origem do seu arquivo AutoCAD for muito longe do desenho real, quando esse arquivo. dxf é importado para o Flexsim, o layout também estará muito longe da posição de origem do modelo no Flexsim. Isso pode ser bastante frustrante para o modelador. Por esta razão, mova seus objetos do AutoCAD para a origem. Faça-o, seguindo os seguintes passos:
 1. Selecione os objetos que você quer movimentar.
 2. Digite "_move" dentro do command prompt ou selecione move command do menu.
 3. Especifique um ponto de referência.
 4. Digite a localização desejada do ponto no command prompt.
4. **Explodir os componentes dos objetos.** Flexsim só pode importar formas básicas, por isso é importante explodir objetos compostos no desenho do AutoCAD em suas formas básicas. Para explodir objetos compostos em AutoCAD siga estes passos:
 1. Selecione os objetos que você quer explodir.
 2. Digite "_explode" no command prompt ou selecione explode command dos menus.
 3. Repita os passos acima até não existir mais objetos que foram explodidos.

No FlexSim:

1. Arreste um objeto Background para dentro do modelo da janela de bibliotecas abaixo de Visual.
2. Siga as etapas no wizard Background Drawing.

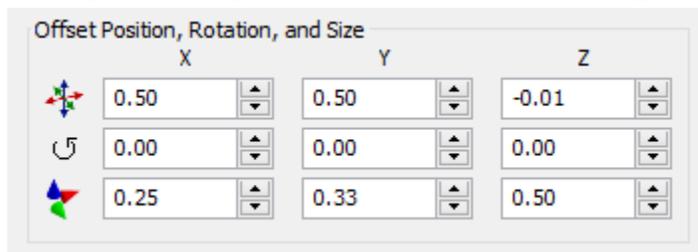
Shape Factors

Cada arquivo de media que é importado para o Flexsim tem certas escalas e configurações que podem fazer com que a forma 3D que você importou não se encaixe corretamente dentro dos limites dos objetos. Neste caso, você pode editar a shape factors dos objetos 3D para encaixar a imagem 3D dentro da caixa amarela dos limites dos objetos.



A imagem acima mostra o shape factors modificado para a forma 3D do processador. Note que o limite representado pela caixa amarela representa a verdadeira posição e tamanho do processador, mas a forma em 3D foi importada na direção x.

O Shape Factors pode ser acessado através da guia General dos objetos.



: Altera a posição da forma 3D para dentro da caixa limitadora.

: Altera a rotação da forma 3D para dentro da caixa limitadora.

: Altera o tamanho da forma 3D para dentro da caixa limitadora.

Shape Frames

Você pode criar animações em um objeto ou flowitem usando 3D frames. Isto é feito ao criar múltiplas formas em 3D. Então, dentro do Flexsim você pode informar ao objeto que shape frame será mostrado. Isto é útil se você quer alterar o visual de um objeto como um Flowitem que progride ao longo do modelo. Um exemplo disto pode ser uma linha engarrafadora. A garrafa começa vazia, então é cheia e então uma tampa é colocada no topo. Isto podem ser mostrado usando três diferentes formas em 3D.

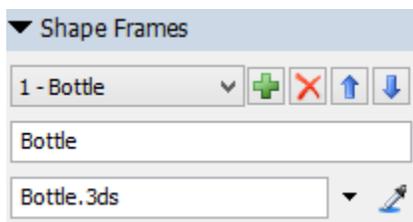
Nas versões anteriores do FlexSim 7, shape frames tinha que ser especificado pelo nome do arquivo ou pela forma 3D. Agora, shape frames são referenciados manualmente ao:

1. Adicionando um Shape Frame para um objeto.
2. Configurando um nome para o frame
3. Especificando um caminho a forma 3D.

Uma vez adicionado, shapes frames pode ser alterado ou pelo nome, ou pelo index. Veja os comandos abaixo. Shapes frames podem ser modificados em dois lugares:

- Na janela Quick Properties em Shape Frames (veja abaixo).
- Na guia General da janela de propriedades de um objeto.

Quick Properties



A seção shape frames da janela Quick Properties nem sempre é exibida para os objetos. Para um FixedResource, TaskExecuter, ou outro objeto que não seja flowitem, a seção Shape Frames será exibida somente se aquele objeto já tem um shape frames referenciado para ele através da guia General.

Flowitems no entanto, sempre tem a seção Shape Frames sendo exibida. Isto se deve ao fato de ser mais comum alterar a forma de um Flowitem conforme ele se altera através da execução do modelo.

Comandos

Para alterar o shape frame de um objeto ou para fazer com que um objeto encontre seu atual shape frame, você precisará usar os seguintes comandos:

```
setframe(current, 1);  
setframe(current, "Bottle2");  
int curframe = getframe(current);
```

Por conveniência, as triggers do objeto como OnReset, OnMessage, OnEntry/OnExit, OnLoad/OnUnload, etc tem uma pré-configuração para alterar o shape frame de um objeto.

Bibliotecas Customizadas

1. Conceitos
2. ModelLibraries Node
3. Exemplo

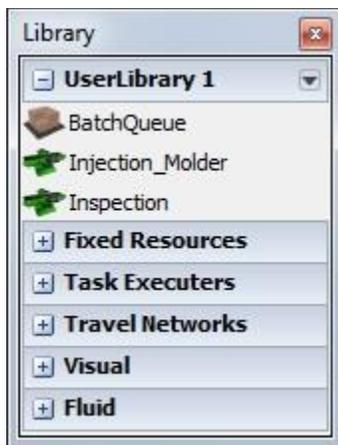
Conceitos de Bibliotecas Customizadas

Tópicos

- Overview
- Adicionando objetos a uma biblioteca customizada
- Editando a Biblioteca de Objetos
- Salvando/Removendo Bibliotecas
- Dropscrip and Droppath
- Instalação Automática
- Customizando os ícones das biblioteca de objetos

Overview

O FlexSim permite você criar e configurar bibliotecas especiais além da biblioteca padrão presente no software. Estes são referenciados como bibliotecas do usuário. Você pode criar funcionalidades definidas pelo usuário sobre os objetos, e então adicionar estes objetos para uma biblioteca para uso em outras partes de seu modelo ou em outros modelos. Você pode salvar estas bibliotecas e então carregá-las em outros projetos mais tarde. Você também pode definir um conjunto de objetos na biblioteca para ser instalado automaticamente para o seu modelo quando um novo modelo é criado ou quando você carrega a biblioteca. Você pode criar uma nova biblioteca de usuário e abrir bibliotecas customizadas do menu File.

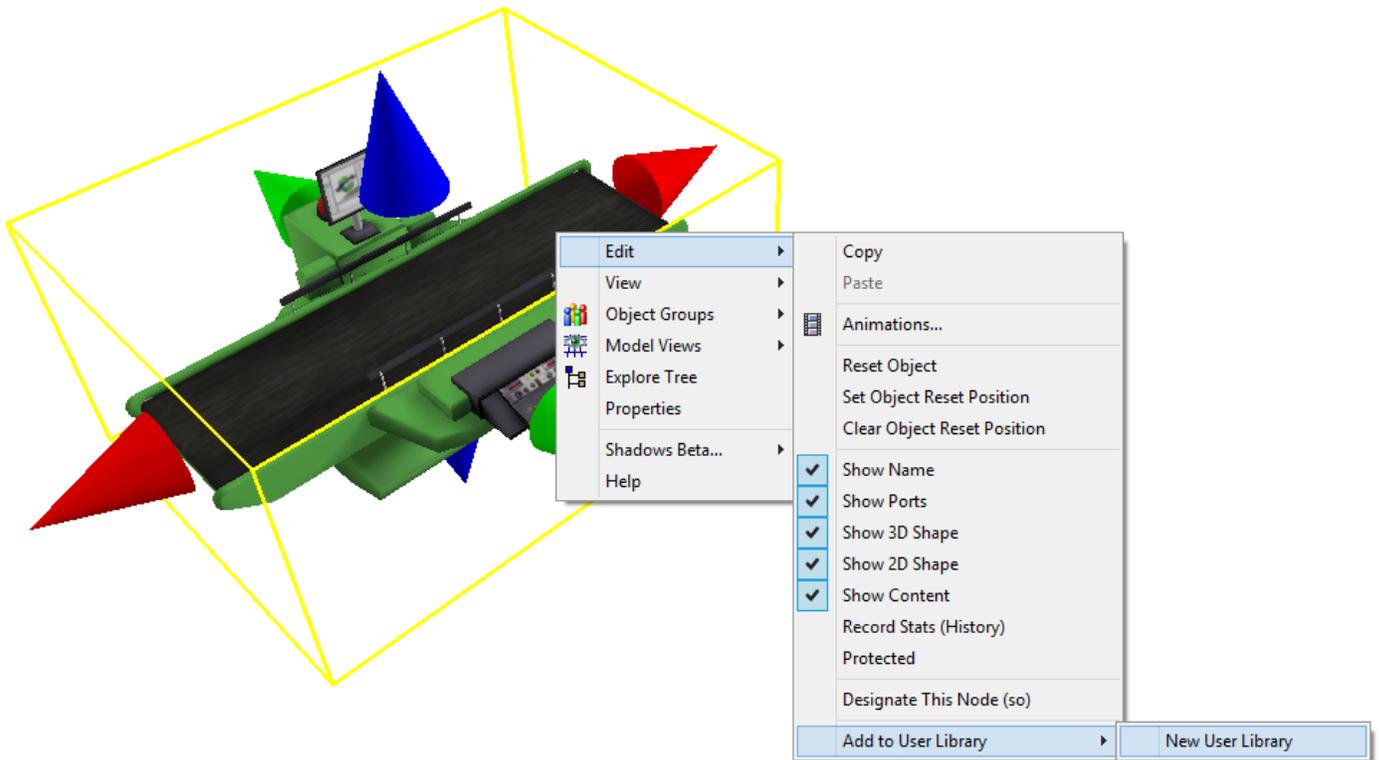


O mecanismo de biblioteca customizada, ou biblioteca do usuário, proporciona-lhe flexibilidade em muitas áreas. A funcionalidade mais utilizada das bibliotecas do usuário é a reutilização de objetos personalizados em um modelo, mas não se limitam a apenas essa funcionalidade. As bibliotecas do usuário permitem que você instale automaticamente os objetos personalizados e de dados em modelos, instale objetos e dados para o projeto principal e vista do tree, e executar código quando um objeto na grade é criado no modelo. Tudo isto deriva a funcionalidade de dois mecanismos dentro da funcionalidade da biblioteca do usuário. O primeiro é o mecanismo Droppath e Dropscrip, que permitem que você personalize o que acontece quando um objeto é criado no modelo. O segundo é o mecanismo de instalação automática, que permite a instalação de objetos ou de dados quando o usuário faz operações como a criação de um novo modelo ou abertura de um modelo. Este tópico discutirá estes dois mecanismos com mais detalhes

Adicionando objetos para um biblioteca customizada

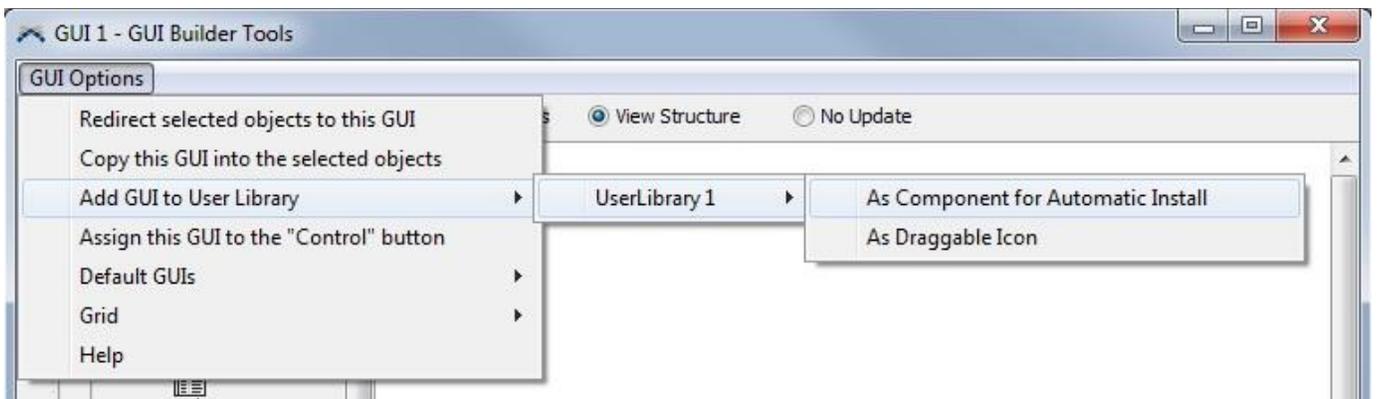
Existem diversos objetos que você pode adicionar para uma biblioteca customizada. Você irá adicionar alguns dos mais utilizados objetos padrões como o Processor ou um Queue cujas propriedades você modificou para se encaixar em uma situação de modelagem customizada. Você pode também iniciar do zero com os objetos BasicFR ou BasicTE, implementando seu comportamento customizado, e então adicioná-lo para a biblioteca. Outra possibilidade é usar o Visual Tool como um container (superfície

contenedora) do sub modelo, e adicionar o sub-modelo completo para a biblioteca. Para adicionar estes tipos de objetos para a biblioteca de usuários, clique com o botão direito do mouse sobre o objeto em sua visualização 3D e selecione **Edit > Add to User Library**.



Se você já tem uma biblioteca de usuários carregada, uma listagem destas bibliotecas irá aparecer no menu de clique com o botão direito do mouse.

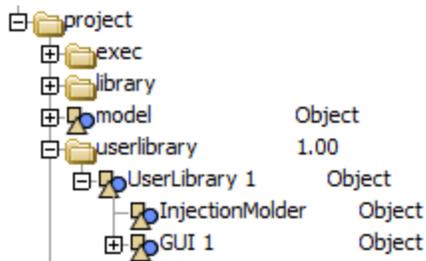
Você também pode adicionar GUIs, Global Table, Flowitems, e User Commands para uma biblioteca customizada. Você pode acessar esta capacidade da janela onde você edita estes respectivos objetos. A imagem abaixo mostra uma opção de menu no editor GUI para adicionar o GUI para uma biblioteca.



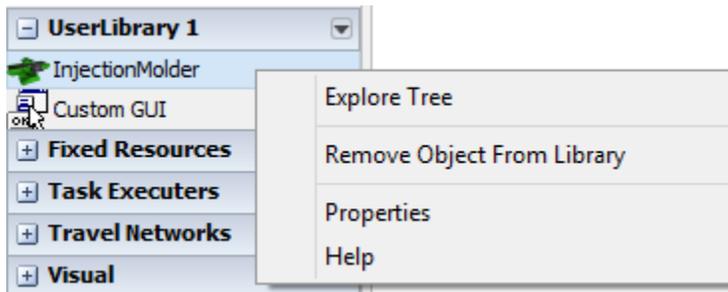
Editando os Objetos da Biblioteca

Quando um objeto é adicionado para uma biblioteca, uma cópia do objeto é criada e colocada na biblioteca. Isto significa que uma vez que eles são adicionados na biblioteca ele não tem mais relação alguma com o objeto original. Ele é uma cópia e pode ser alterado por conta própria, separado do objeto original.

A biblioteca do usuário é armazenado no Main Tree no node userlibrary.

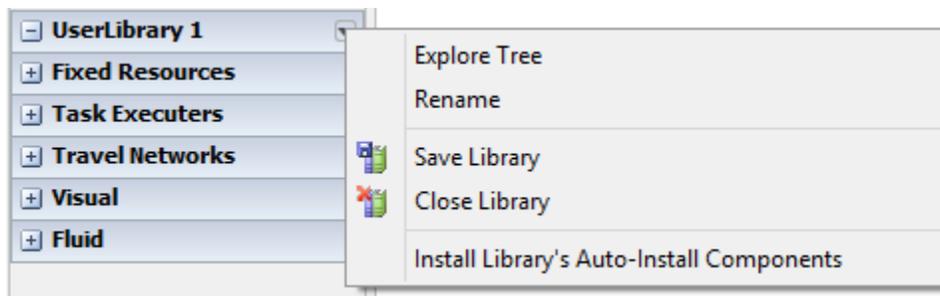


Você pode também acessar as propriedades de sua biblioteca através do menu acessado com o clique do botão direito do mouse sobre o Library Icon Grid (grade de ícones da biblioteca).



Salvando / Removendo Bibliotecas

Para salvar ou remover uma biblioteca de usuários, clique sobre a flecha em preto próximo ao User Library no Library Icon Grid.

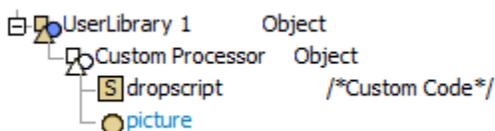


Dropscrip e Droppath

O mecanismo Droppath e Dropscrip permite que você personalize o que acontece quando um objeto é criado no modelo. Isto é feito através da criação de um objeto personalizado na biblioteca do usuário e, em seguida, adicionando ou um "Dropscrip" ou um atributo "Droppath" para o objeto e especificar os dados para o Dropscrip ou Droppath.

Dropscrip

Se um atributo Dropscrip é adicionado, em seguida, o node Dropscrip será executado como flexscript, e os valores serão passados para a função flexscript que dão informações sobre a criação.



Nota: Para que um objeto seja exibido na Library Icon Grid, ele necessita ter um atributo de imagem conforme mostrado acima em nosso objeto customizado.

Como mostra o exemplo , existem cinco variáveis de acesso que você pode usar dentro da função Dropscrip . Eles são:

parnode (1): o objeto que do qual o item foi criado sobre . Se o usuário arrastou para uma área em branco no modelo, então esta variável será 0 ou NULL. Nesse caso , geralmente você pode assumir que o usuário tem a intenção de soltá-lo no modelo. No entanto , nem sempre é o caso. Por exemplo, o usuário pode soltar o objeto em uma visão orto aberta que está vendo um subespaço do modelo, como uma VisualTool . Para assegurar uma criação segura , neste caso , você deve usar os viewfocus da vista como o ponto de criação (node ("> viewfocus +" , parnode (5))) .

parval (2), parval (3), parval (4): x, y e z são a localização do ponto onde o objeto foi deixado, respectivamente. Se o parnode (1) é NULL, então este é o local dentro do espaço do modelo (ou espaço ' viewfocus da vista). Se parnode (1) não é NULL, então este é o local dentro parnode (1).

parnode (5): a janela de vista no qual o objeto foi descartado. Use node ("> viewfocus +" , parnode (5)) para obter acesso ao viewfocus da vista.

Se dentro da função Dropscrip você gostaria de fazer uma criação padrão de um objeto, como se fosse um usuário regular que cria o objeto no mesmo local, então você pode usar o comando **dropuserlibraryobject ()**. Este comando executa a mesma funcionalidade que quando um objeto regular é retirado da biblioteca do usuário. O primeiro parâmetro é o objeto a ser criado, e os próximos cinco parâmetros são os parâmetros em x, y, z, e a vista , assim como a função Dropscrip em si. O comando retorna uma referência ao objeto que foi criado.

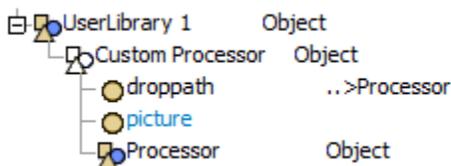
Return Value - Sua função Dropscrip também deve retornar uma referência para o objeto que você criou. Isso permite que o FlexSim faça algum trabalho extra, como a verificação para ver se havia algum node criado em C++, e em caso afirmativo, definir um sinalizador para notificá-lo de que você precisa para compilar a próxima vez que você executar o modelo.

O mecanismo Dropscrip permite máxima flexibilidade com as bibliotecas do usuário. Você pode usá-lo da forma padrão , por exemplo, para criar um objeto regular, mas também executar um código extra para inicializar a criação do objeto . Você também pode usá-lo de forma mais fora do padrão. Algumas ideias podem ser a de usar um script para adicionar uma seção de curvas ou retas para uma correia transportadora, ou para adicionar níveis ou baias de Rack. Você também pode usar uma Dropscrip para adicionar um código para a trigger de um objeto, ou para definir alguns parâmetros do objeto. Você pode adicionar esferas de colisão em um objeto, ou adicionar um conjunto de funcionalidades para a pasta de ferramentas do modelo. Há um número ilimitado de possibilidades.

Veja a seção de exemplos sobre como usar o Dropscrip node.

Droppath

O mecanismo Droppath é um método de vias indiretas. É uma maneira de se referir a um outro objeto que você deseja criar quando o ícone é criado no modelo. Este é normalmente usado com a funcionalidade de instalação automática, mas ele também pode ser usado com um ícone na grade de ícones. O método de criação de um objeto Droppath é o mesmo que para a criação de um Dropscrip. Você pode adicionar um objeto para a biblioteca, e dar-lhe os dados do objeto, mas desta vez você dá um atributo chamado "Droppath". Você não deve alternar o node como flexscript, mas deve dar dados de texto para o atributo Droppath e especificar no texto um caminho para outro objeto que deve ser criado.



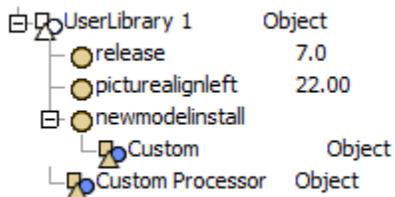
Por exemplo, você pode dar o atributo Droppath o texto: MAIN projeto :/project/ library/NetworkNode. Isto efetivamente faz com que um NetworkNode seja criado quando se arrasta o ícone de um objeto para o modelo. Você pode perguntar por que não basta adicionar um node da rede para a biblioteca do usuário, ou por que não apenas fazê-los ir para a biblioteca regular e criar um NetworkNode de lá. Para a primeira questão, se você planeja usar essa biblioteca do usuário por um longo tempo, você pode querer que o NetworkNode seja compatível com as versões futuras do Flexsim. Você pode querer usar sempre tudo o que é a versão mais recente do NetworkNode, sem a necessidade de atualizar a biblioteca do usuário a cada nova versão. Além disso, porque ter dados redundantes na sua biblioteca do usuário, se tudo isso já está na biblioteca regular? Para a última pergunta, pode ser útil ter o NetworkNode como um ícone “arrastável” em sua biblioteca, se o usuário usa este objeto muitas vezes. Ele reduz o número de cliques de mouse necessários.

O droppatch pode ser ambos um caminho absoluto *MAIN:/project/library/fixedresources/Processor* ou relativo *..>Processor*.

Instalação Automática

O mecanismo de instalação automática permite que a biblioteca do usuário instale objetos e funcionalidades no modelo quando o usuário executa certas ações como a criação de um novo modelo, ou a abertura de um modelo. A instalação automática está disponível através da interface do usuário para o GUI, Global Tables, Flowitems, e User Commands.

A maneira de fazer isso é através da criação de pastas especiais dentro de atributos no tree da biblioteca (library). Mais uma vez, vamos fazer um outro exemplo. Vamos começar a partir de nosso exemplo anterior, e em vez de ter o objeto criado quando o usuário arrasta o ícone, vamos criá-lo quando o usuário cria um novo modelo. Reduzir os atributos do objeto personalizado e expandir atributos do tree library, pressionando o botão  ao lado da library. A tree deve aparecer conforme guiaixo



Para criar uma instalação automática customizada, você pode adicionar nodes especiais para os objetos da biblioteca customizada do usuário.

newmodelinstall: Esta pasta é instalado quando o usuário cria um novo modelo. Ela também é instalada quando a biblioteca é carregada.

startupinstall: Esta pasta é instalada quando a biblioteca é carregada se o usuário tiver designado esta biblioteca como uma biblioteca para carregar durante a inicialização.

loadinstall: Esta pasta é instalada quando a biblioteca é carregada explicitamente pelo usuário.

openmodelinstall: Esta pasta é instalada quando o usuário abre um modelo existente. Você pode usar esta pasta para verificar se o modelo contém os componentes da biblioteca, e se não, instalá-los. Você também pode usá-lo para atualizar os componentes do modelo se esses componentes são de uma versão anterior da biblioteca do usuário.

Customizando os Ícones da Grade de Biblioteca

Existem alguns atributos que podem ser adicionados para os objetos em sua biblioteca customizada que irá afetar a forma que eles são mostrados pelo Library Icon Grid.

Nota: Alguns atributos visuais não serão atualizado na library icon grid sem fechar e reabrir a view.

Pictures

O atributo picture especifica o caminho para uma imagem mostrar o ícone no grid. Deixando o caminho em branco irá mostrar o nome do objeto apenas. Para dar ao seu objeto uma imagem do Processor, seu atributo picture conterá o caminho:

bitmaps\processorpicturesmall.png

Você pode também especificar o alinhamento esquerdo ao alterar o valor do node `picturealignleft`.

Largura e Altura da Grade de Ícones

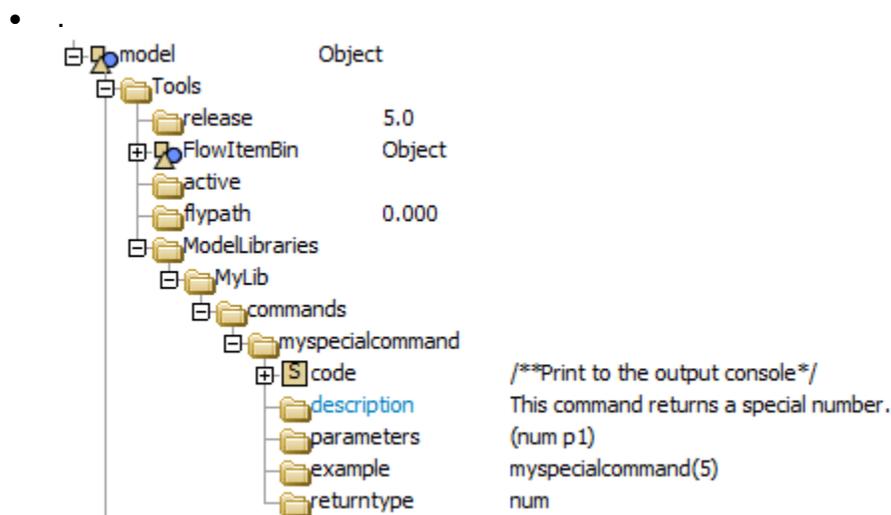
Você também pode designar o tamanho dos ícones na grade de ícones da biblioteca do usuário. Basta dar aos atributos da biblioteca o nome "cellwidth" e "cellheight". Cada atributo deve conter um número que representa a largura e altura de um ícone na grade de ícones, em pixels.

ModelLibraries Node

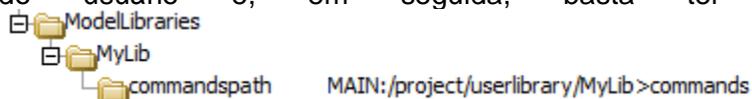
Você também pode adicionar muitas facetas para um modelo, como os comandos do usuário, variáveis globais, macros globais, etc, onde a definição é "escondida" do usuário, de modo que eles não vejam em sua janela de User Comands (Comandos do Usuário), variáveis globais, etc. Estes podem também simplesmente se remetem para estruturas em sua biblioteca, de modo que você não precise copiar continuamente coisas no modelo durante atualizações. Isto é feito através da adição de um node chamado "ModelLibraries" para a pasta Tools do modelo e dando suas subestrutura vários dados. Enquanto você não precisa ter uma biblioteca personalizada carregada para que esse recurso seja parte de um modelo, ele provavelmente será mais utilizado para bibliotecas do usuário, por isso, vamos incluir esta informação a este tópico.

Para criar essa funcionalidade, adicione um node no model/Tools, e dê-lhe o nome de "ModelLibraries". Dentro desse node, adicione outro node, e dê-lhe o nome da sua biblioteca, ou seja, MyLib. Dentro desse node, existe vários nodes que podem ser adicionados. Seu nome define seu significado para o Flexsim. Os nomes e significados estão listados guiaixo:

- **commands** - Se você adicionar um node chamado "commands", você pode colocar os comandos dentro desse node. Estes comandos serão acessíveis às chamadas do usuário, e serão documentados na documentação dos comandos, mas não serão visíveis na janela de comandos do usuário (User Comamands). A estrutura de cada comando deve ser exatamente o mesmo que a estrutura do model/Tools/UserCommands.

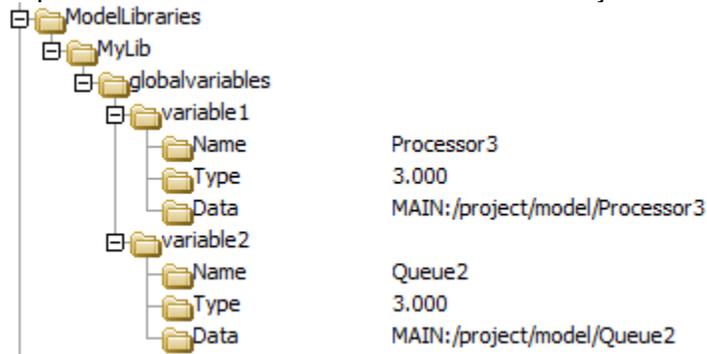


- **commandspath** – Você pode adicionar um node chamado "commandspath" e dar-lhe dados de texto que especifica o caminho para um local alternativo que mantém a definição dos comandos. Neste caso, a estrutura de destino do caminho deve ser exatamente o mesmo que a estrutura do model/Tools/UserCommands. Isso permite que você deixe suas definições de comandos na biblioteca do usuário e, em seguida, basta ter esse node referenciado a ele.



- **globalvariables** – Você pode adicionar um node chamado "globalvariables" e dar-lhe a mesma subestrutura encontrada em Tools/GlobalVariables para definir variáveis globais adicionais

disponíveis ao usuário final e “escondidas” da janela “Global Variables”.



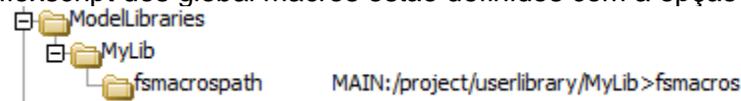
- **globalvariablespath** – Do mesmo modo que o commandspath, você pode redirecionar o local onde a global variable está definida.



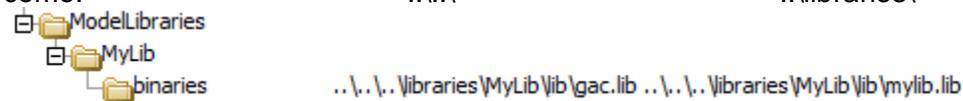
- **fsmacros** – Aqui se pode adicionar um node chamado "fsmacros", insira como dado neste node múltiplos #define macros. Estes macros estarão visíveis (em azul) para o usuário e serão adicionados como opções de auto completar.



- **fsmacrospace** – Do mesmo modo que o commandspath, você pode redirecionar o local onde o flexscript dos global macros estão definidos com a opção "fsmacrospace".



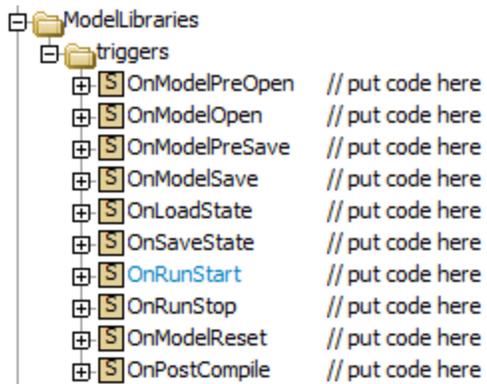
- **binaries** – Você pode adicionar um node chamado "binaries" e dar-lhe dados de texto que especifica uma lista adicional .lib ou .obj que você deseja vincular ao Flexsim. Isto só é necessário se você tem o código alterado em C++ que deve ser compilado, ou se você espera que seus usuários finais definam o código C++ que podem acessar as funcionalidade pré-compiladas. Este texto será adicionado ao campo "AdditionalLibraryDependency" durante a fase de vinculação do C++. O caminho linkado a biblioteca é especificado como \program\system\lib no diretório de instalação do FlexSim, e para referenciar um arquivo .lib no diretório da biblioteca do FlexSim, você deverá especificar um caminho como:



- **binariespath** – Do mesmo modo que o commandspath, você pode redirecionar o local onde as opções de link estão especificadas com a opção "binariespath".



- **triggers** – Você pode também adicionar diversas triggers para acionar alguns pontos. Para isso, crie um node e o nomeie como “triggers”, a então adicione diversos nodes dentro do node triggers.



As triggers válidas são:

- **OnModelPreOpen** – Acionada imediatamente depois do tree do modelo ser carregado, e antes de qualquer código na inicialização do modelo ser ativado.
- **OnModelOpen** – Acionada depois do modelo ser aberto e todos códigos de inicialização serem finalizados.
- **OnModelPreSave** – Acionada imediatamente antes de um modelo ser salvo em algum arquivo.
- **OnModelSave** – Acionada imediatamente depois do modelo ser salvo em um arquivo.
- **OnLoadState** – Acionada quando um estado for carregado.
- **OnSaveState** – Acionada quando um estado for salvo.
- **OnRunStart** – Ativada sempre que o usuário pressionar o botão “Run”, assim como quando for chamado o comando go().
- **OnRunStop** – Acionada sempre que pressionado o botão stop().
- **OnModelReset** – Acionada sempre que o usuário pressionar o botão “Reset”, assim com quando for chamado o comando resetmodel().
- **OnPostCompile** – Acionada depois que o usuário compila o modelo.

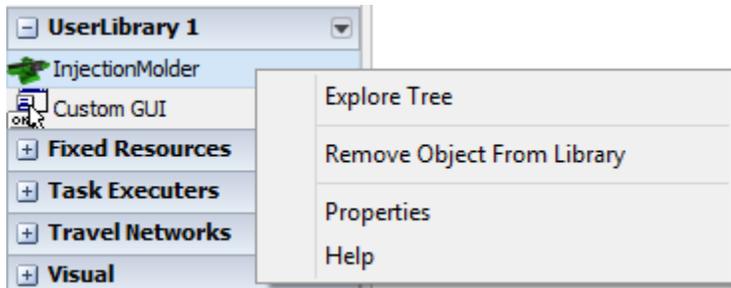
Exemplo de Bibliotecas Customizadas

Tópicos

- **Dropscrip Example**
- **Dropscrip Dynamic Parameters Example**
- **Automatic Install Example**

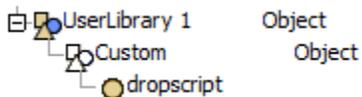
Exemplo Dropscrip

Crie uma nova biblioteca de usuário indo em File > New User Library. Clique sobre a seta em preto para baixo, próximo a mais nova biblioteca de usuário na Library Icon Grid e selecione Explore Tree.



Você deve agora ver uma tree em branco.

Insira um novo objeto para a biblioteca com o botão direito sobre ele e selecionando Node> Insert Into, ou clicando com o botão esquerdo sobre o objeto e apertando a tecla Enter. Expanda a tree da biblioteca do usuário para que você possa ver o node que você criou apertando o botão "+". Dê-lhe o nome de "Custom". Agora adicione os dados do objeto para o node com o botão direito do mouse sobre ele e selecione Node> AddObjectData. Clique no botão > para expandir o objeto. Em seguida, insira um node de atributo clicando no node do objeto e apertar a tecla Enter. Dê ao node o nome de "Dropscrip". A árvore deve aparecer conforme abaixo.



Adicione o texto ao atributo Dropscrip com o botão direito do mouse sobre ele e selecionar Node> Add Text Data. Alterne o node como flexscrip com o botão direito do mouse sobre ele e selecione Build>Toggle as FlexScript. Dê ao atributo o seguinte texto em Flexscript:

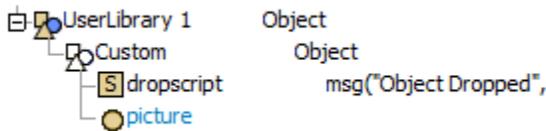
```
msg("Object Dropped",
    concat(
        "Onto Object: ",
        getname(parnode(1)), "\n",
        "X: ",
        numtostring(parval(2),2,2), "\n",
        "Y: ",
        numtostring(parval(3),2,2), "\n",
        "Z: ",
```

```

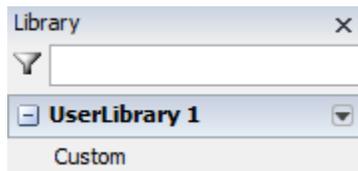
        numtostring(parval(4),2,2), "\n",
        "Onto View: ", getname(parnode(5))
    )
);

```

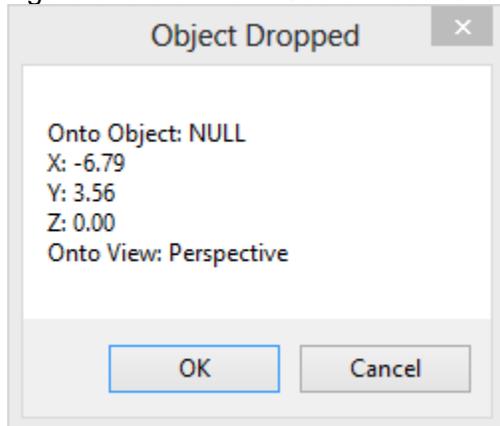
Clique fora do atributo node, então com o botão direito do mouse selecione Build>Build Node FlexScript. Finalmente, adicione um atributo de imagem para a árvore atributo do objeto com o botão direito do mouse sobre o node drop-script e selecione Node> Insert After. Nomeie o novo node como "picture". O ícone na biblioteca só vai mostrar o objeto na grade se o objeto tem um atributo de imagem. Geralmente este é um caminho para um arquivo bitmap que representa a imagem que é mostrada na grade de ícones desse objeto. No nosso caso, não nos preocuparemos com isso, então deixaremos o atributo picture em branco, o que faz com que um ícone em branco seja exibido na grade com o nome do objeto. A árvore deve aparecer assim.



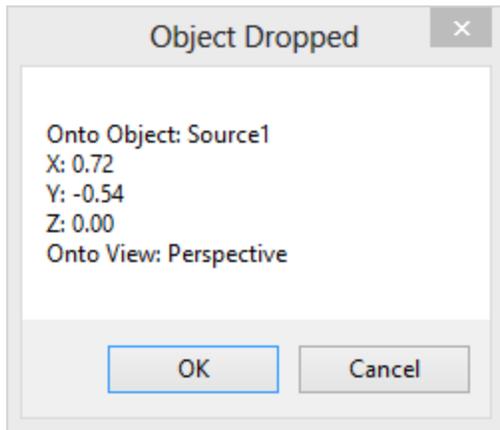
O library icon grid deve-se parecer com:



Agora arraste o item Custom da biblioteca para o modelo. A seguinte mensagem deverá aparecer



Esta mensagem é o código da Dropscrip sendo executado. Ele mostra o objeto que caiu sobre e as posições em x, y e z do local onde o item foi deixado. Você também pode deixá-lo em outro objeto. Arraste um objeto de biblioteca regular em seu modelo, em seguida, volte para a biblioteca do usuário e arraste o objeto personalizado para o objeto no modelo. A seguinte mensagem deve aparecer.



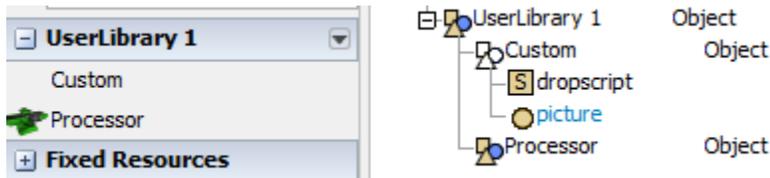
Desta vez, a mensagem mostra que ela foi deixada no Source3 . Os parâmetros x , y e z estão agora em relação ao objeto de origem (source)

Exemplo Parâmetros Dropscrip Dynamic

Neste exemplo, nós teremos um objeto que é criado através da User Library que terá seu tamanho definido por uma Global Table.

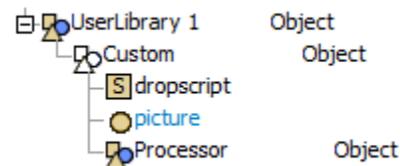
Crie um nova User Library conforme descrito no exemplo do Dropscrip.

Arraste e solte um objeto Processor da grade de biblioteca de objetos (Library Icon Grid) para dentro da superfície de simulação em 3D. Clique com o botão direito do mouse sobre o Processor e selecione Edit > Add to User Library > UserLibrary 1 para adicionar a nossa biblioteca.



Nós queremos dinamicamente alterar o tamanho do nosso Processor conforme ele é arrastado para dentro da superfície de simulação. Para executar isso, nós iremos usar o Custom Objeto com código em nosso código Dropscrip. Nós iremos copiar nosso processor para dentro do nosso custom object como um attribute node.

Copie o objeto Processor (clique com o botão esquerdo mouse e selecione Ctrl + C). Clique com o botão esquerdo do mouse sobre o Custom Object e abra os atributos dos objetos na tree se ainda não estiver aberto, pressionando o >. Clique na tecla enter para criar um novo attribute node em nosso Custom Object. Clique com o botão esquerdo do mouse sobre o novo node e pressione Ctrl + P para copiar o objeto Processor. Você pode então deletar o objeto original Processor.



Crie uma nova GlobalTable através da ferramenta Tools > Global Tables > Add. Nomeie a tabela como SizeTable. Faça com que a sua GlobalTable se pareça conforme segue:

SizeTable	Columns	Rows
Custom	sx	10.00
	sy	2.00
	sz	2.00

Os valores desta tabela irão definir o tamanho de nosso processor.

De volta ao Custom Objeto, atualize o código Dropscrip para o seguinte:

```
treenode ontoObj = parnode(1);
double x = parval(2);
double y = parval(3);
double z = parval(4);
treenode ontoView = parnode(5);

treenode newProcessor = dropuserlibraryobject(node("../>Processor", c), ontoObj, x, y,
z, ontoView);
setnodenum(spatialsx(newProcessor), gettablenum("SizeTable", 1, 1));
setnodenum(spatialsy(newProcessor), gettablenum("SizeTable", 1, 2));
setnodenum(spatialsz(newProcessor), gettablenum("SizeTable", 1, 3));
return newProcessor;
```

Clique com o botão direito do mouse no node Dropscrip e selecione Build > Build Node FlexScript.

Agora crie um custom object (objeto customizado) ao arrastar e soltar o objeto da Library Icon Grid para a superfície de simulação em 3D. Note que o tamanho do objeto foi alterado pelo nosso Dropscrip.



Nota: Esta é a forma mais apropriada de alterar dinamicamente os parâmetros ou variáveis de um objeto de biblioteca customizada quando é solto dentro da superfície em 3D (embora não importa onde o objeto criado está contido). Se você adicionou um node Dropscrip diretamente para o Processor, o Processor não será criado. Se você tentou e criou o objeto no node Dropscrip do Processor, você fará com que o FlexSim fique em um loop infinito e pare.

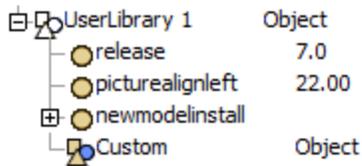
Exemplo de Instalação Automática

Para este exemplo, nós teremos o objeto Processor do exemplo Dropscrip Dynamic Parameters sendo instalado automaticamente quando o usuário criar um novo modelo.

Abra a GlobalTable Size Table. Clique sobre o botão  e selecione **Add to UserLibrary 1 > As Auto-Install Component**

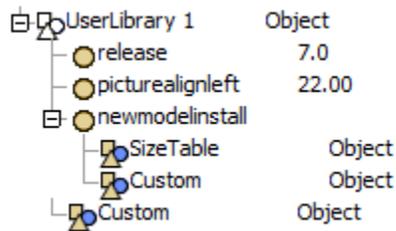
Abra a árvore (Tree) User Library 1's ao clicar na seta próximo ao User Library na Library Icon Grid e selecione Explore Tree.

Expanda a árvore (tree) de atributos da biblioteca ao pressionar no botão  próximo ao UserLibrary 1. O tree deve parecer-se com algo abaixo.



Pressione o botão + próximo ao newmodelinstall para expandir o node. Note que o SizeTable já aparece dentro de seu node. Isto é importante conforme nosso Custom object Dropscrip se refere para esta GlobalTable.

Crie um subnode de newmodelinstall clicando com o botão esquerdo do mouse e pressionando Enter. Copie e cole o Custom object dentro deste novo node.



Agora você pode pressionar o botão “New” sobre a barra de ferramentas do File > New Model. Você notará que existe um objeto Processor1 na superfície de simulação 3D e o nossa SizeTable na GlobalTables. A pasta newmodelinstall armazena um conjunto de objetos que serão “soltadas” dentro do modelo quando um novo modelo é criado. Os objetos são “soltados” nesta mesma funcionalidade que é executada quando você na verdade arrasta um objeto da grade biblioteca de objetos (library icon grid). Neste caso do objeto Dropscrip, o script é executado conforme um objeto for solto no ponto (0,0,0) no modelo.