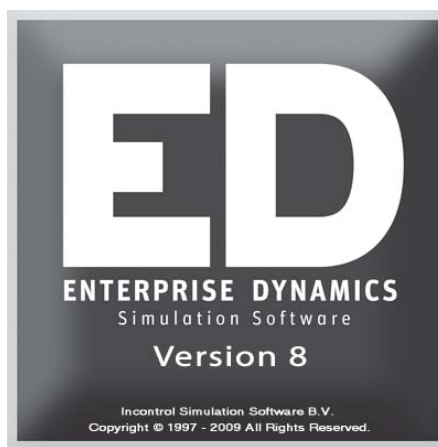


# BIJLAGEN TUTORIAL



Copyright © 2009, Incontrol Simulation Solutions  
Papendorpseweg 77  
3528 BJ Utrecht  
The Netherlands  
[www.IncontrolSim.com](http://www.IncontrolSim.com)



## Bijlage 1

## De menustructuur

Nb. Een *cursief* gedrukte optie is meer voor de gevorderde gebruiker van belang.

<b><i>File Menu</i></b>	<b><i>Uitleg</i></b>
New Model	Sluit het huidige model en opent een leeg model.
Open Model...	Sluit het huidige model en opent een bestaand model.
<i>Merge Model...</i>	<i>Opent een model en plaatst dit model in het geselecteerde atoom of in het huidige model. Het bestaande model wordt dus niet vernietigd.</i>
Save Model	Slaat het huidige model onder dezelfde bestandsnaam op.
Save Model As...	Slaat het huidige model op met een nieuwe bestandsnaam. Het oude model blijft bestaan.
<i>Add Atom to Library...</i>	<i>Laadt een atoom en plaatst het aan het einde van de bibliotheek.</i>
<i>Save TreeAtom</i>	
<i>Save Atom As...</i>	<i>Toont een scherm waarmee een atoom uit de bibliotheek geselecteerd kan worden. Vervolgens kan het geselecteerde atoom onder een nieuwe naam worden opgeslagen.</i>
<i>Import</i>	<i>Hiermee kan een 2D/VR (VR=Virtual Reality) icoon of een VR geluid worden geïmporteerd. De icoon of het geluid wordt toegevoegd aan de lijsten met iconen of geluiden die de gebruiker aan een atoom kan toewijzen.</i>
Print 2D Layout...	Drukt het 2D scherm af op de standaardprinter.
Print Setup	Hiermee kunnen de instellingen van de standaardprinter worden aangepast.
Preferences	Deze optie bevat een aantal bladen voor het aanpassen van standaard instellingen van Enterprise Dynamics.
<i>Startup Script</i>	<i>Hiermee kan het Startup Script worden aangepast. Dit script wordt elke keer dat Enterprise Dynamics start uitgevoerd. De aanpassingen moeten in de programmeertaal 4DScript van Enterprise Dynamics worden geformuleerd.</i>
Exit	Hiermee wordt Enterprise Dynamics afgesloten.

<b><i>Model Menu</i></b>	<b><i>Uitleg</i></b>
Create	Toont het zogeheten model tree (onderdeel van de bibliotheek) en het model lay-out scherm. Door atomen vanuit de model tree in de model lay-out te slepen, is een eigen model te bouwen.
Layout window	Toont het model layout scherm. Atomen kunnen of in dit scherm worden gecreëerd door atomen te slepen vanuit de bibliotheek of Menubar.
<i>Sublayout Window</i>	<i>Toont eveneens een Layout, maar dan een hiërarchische laag dieper. Bijvoorbeeld de inhoud van een Composition Container.</i>
Model Tree	Toont de model tree. Deze 'model tree' geeft een hiërarchisch overzicht van de atomen in het model
Library Tree	Deze optie toont de bibliotheek met atomen.

<b>Simulate Menu</b>	<b>Uitleg</b>
Run Control	Toont het Run Control scherm, gebruikt voor het starten, stoppen en instellen van de snelheid van een simulatierun.
Clock	Deze optie toont de klok.
Run	Start een run van het model via Ctrl-F11.
Stop	Onderbreekt een run van het model via Ctrl-F10.
Stop and Reset	Onderbreekt en reset het model via Ctrl-F9.
History	<p>Toont het History instellingenscherm. De History zorgt ervoor dat er tijdens een simulatierun data wordt bewaard waarmee na afloop via de menu-optie Results   Graphs grafieken kunnen worden opgesteld van die run.</p> <p>Er wordt alleen History bijgehouden als de optie 'general history' aangevinkt is. Deze optie kan ook aan- en uitgezet worden door middel van de rode "Record" knop op het Run Control scherm.</p> <p>Verder dienen de specifieke atomen waarvan History moet worden bijgehouden, hier geselecteerd te worden. Alleen bij zeer kleine modellen kan voor de optie "All on" gekozen worden. Bij grotere modellen leidt deze optie tot grote vertragingen en grote databestanden.</p>

<b>Results Menu</b>	<b>Uitleg</b>
Summary Report	Toont het Summary Report, dat een overzicht geeft van de basisstatistieken van alle atomen in het model.
Graphs	Toont diverse grafieken van atomen zoals wachtrijen, histogrammen en taartdiagrammen. Deze kunnen alleen worden gemaakt als de historie voor het betreffende atoom aangezet is.

<b>Experimentation</b>	<b>Uitleg</b>
Experiment Wizard	Hier definiëren en voeren we experimenten uit: in de eerste stap worden de runlengte, het aantal runs e.d. opgegeven, waarna de definitie van de uitvoervariabelen volgt. Als laatste vindt de daadwerkelijke simulatie van het experiment plaats
Analyze Results	Toont de resultaten van het bovenstaande experiment in diverse overzichten. Deze rapporten zijn zelf in te stellen en te exporteren naar bv. Excel.

<b>Tools Menu</b>	<b>Uitleg</b>
<i>Atom Editor</i>	<i>Met behulp van de Atom Editor is zowel de functionaliteit als het uiterlijk van een atoom aan te passen. Dit is een zeer krachtig hulpmiddel waarmee de werking van bestaande atomen volledig te veranderen is en waarmee eigen atomen ontwikkeld kunnen worden. Binnen de Atom Editor wordt gebruik gemaakt van de programmeertaal 4DScript.</i>
<i>4DScript Interact</i>	<i>Een scherm waarin 4DScript code kan worden ingevoerd, waarna direct uitvoering plaatsvindt.</i>
<i>Text Editor</i>	<i>Een simpele text editor, qua functionaliteit vergelijkbaar met MS Notepad (MS Kladblok).</i>
<i>Debugger...</i>	<i>Opent het Debugger scherm. Met de Debugger kun je de 4DScript code volgen terwijl hij wordt uitgevoerd en daarmee sneller programmeerfouten opsporen.</i>
<i>CAD Import Wizard</i>	<i>Optionele component om CAD tekeningen in te lezen.</i>
<i>GUI Builder...</i>	<i>GUI betekent Graphical User Interface: hiermee kan de gebruiker zijn eigen invoerscherm ontwerpen.</i>
<i>Scenario Manager</i>	<i>Opent de Scenario Manager. Hiermee is het mogelijk om meerdere simulatieruns in te plannen en in een keer uit te voeren. Ga naar huis en vind de resultaten de volgende morgen.</i>
<i>View Atom Labels</i>	<i>Deze optie laat alle labels zien van het geselecteerde atoom (en alle atomen die zich in dat atoom bevinden). Labels zijn variabelen en attributen die de gebruiker aan een atoom kan toewijzen.</i>
<i>Autofit</i>	<i>De autofit functie analyseert een dataset en zoekt de meest passende kansverdeling.</i>

<b>Display</b>	<b>Uitleg</b>
<i>2D Model Layout</i>	<i>Opent het 2D Layoutscherm.</i>
<i>2D Model View</i>	<i>Opent het 2D animatiescherm. Let op! Het is in dit scherm niet mogelijk om atomen toe te voegen aan het model. Daarvoor is het Model lay-out scherm nodig.</i>
<i>2D Model Subview</i>	<i>Opent hetzelfde scherm als de 2D Model View optie, alleen toont het slechts de inhoud van het geselecteerde atoom.</i>
<i>2D Background Color</i>	<i>Met deze optie kunt u de achtergrond kleur van het huidige actieve 2D modelling scherm.</i>
<i>3D Model View</i>	<i>Opent het 3D animatiescherm</i>
<i>3D Model Subview</i>	<i>Opent hetzelfde scherm als de 3D Model View optie, alleen toont het slechts de inhoud van het geselecteerde atoom.</i>
<i>3D Background Color</i>	<i>Opent een kleuren selectiescherm. Hiermee kan de achtergrondkleur van het 3D scherm worden ingesteld. Deze kleur wordt ook door het VR scherm gebruikt.</i>







<b>Search Menu</b>	<b>Uitleg</b>
Search Text or Atom	Opent het zoekscherm.
TreeAtom in 2D Model view	Selecteert in het model view window het atoom dat op dit moment geselecteerd is in het Tree view.
AnimAtom in Tree view	Selecteert in de Tree view het atoom dat op dit moment geselecteerd is in de Model view.
Mother of TreeAtom in Tree view	Selecteert in de Tree view het moeder atoom van het op dit moment geselecteerde atoom.


















<b>Window Menu</b>	<b>Uitleg</b>
Close all windows	Sluit alle schermen
4DScript Overview	Toont een overzicht van alle 4DScript expressies en een uitleg van hun syntax. Dit scherm is ook te openen door op de F2 knop te drukken.
Error Monitor	<i>Opent een scherm waarin de fouten staan die de gebruiker in 4DScript heeft gemaakt.</i>
Tracer	<i>Opent het Tracer scherm. Het is mogelijk om met behulp van 4DScript berichten naar dit scherm weg te schrijven.</i>
Layers	<i>Hiermee kan gekozen worden welke atomen zichtbaar zijn</i>
Resources Manager	<i>Opent een scherm waarin alle iconen zijn weergegeven die voor de atomen gebruikt kunnen worden.</i>
Graph Window	<i>Opent de meest recente grafiek. Het is in dit scherm niet mogelijk om nieuwe grafieken te produceren.</i>






























<b>Help Menu</b>	<b>Uitleg</b>
Help Overview	Biedt toegang tot de complete handleiding, samengesteld uit de drie volgende menu-onderdelen
Quick start	Hier vind je documenten die een aantal nieuwe features van Enterprise Dynamics beschrijven.
Tutorials	Bevat diverse Tutorials in pdf-format
Add-ins	<i>Verschaft hulp over de GUI-Builder en OptQuest Nb. OptQuest is een algemeen optimalisatieprogramma en dient apart te worden aangekocht. Deze optie kan dus ontbreken!</i>
Example Wizard	Opent de Example Wizard. Vanuit de Example Wizard kunnen veel van de voorbeeld modellen die meegeleverd worden bij Enterprise Dynamics gevonden en geopend worden.
About Enterprise Dynamics	Hierin staat informatie over de gebruikte versie en van Enterprise Dynamics en is ook het licentie nummer terug te vinden.
























## Speed buttons in de Main Toolbar

Er zijn meerdere speed buttons beschikbaar in de Main Toolbar. Verschillende van deze buttons zijn voor standaard acties op bestanden bijvoorbeeld het opslaan van een model. Andere buttons open speciale windos zoals de Model Layout en de Run Control. De meeste buttons zijn echter voor het snel plaatsen van een atoom in het Model Layout Window. Daarnaast zijn er nog buttons voor tools bijvoorbeeld de Autofit tool of een tool om de Labels die op een atoom staan weer te geven.




File actions	<i>Uitleg</i>
	Creëer een nieuw model.
	Open een bestaand model.
	Bewaar het huidige model.
	Save library dialog.
	Automatically creates a preregister file for all functions used in the library. Can be used to prevent errors when loading a library.
	Maak een afdruk van het huidige beeld in de Model Layout.





Window actions	<i>Uitleg</i>
	Toon de Model Layout en Library Tree.
	Toon de Model Layout waarin het model gebouwd kan worden.
	Toon het 2D model scherm. In dit 2D Model scherm kunnen alleen de instellingen van de bestaande atomen gewijzigd worden. Er kunne geen nieuwe atomen toegevoegd worden.
	Toon het 3D model scherm.
	Toon het Layer scherm. Vanuit dit scherm maak je verschillende lagen (layers) aan in je Model en stel je in wat de huidige laag is. Per laag kun je de volgende instellingen in en uitschakelen; atomen zijn zichtbaar, atomen zijn instelbaar in grote, atomen zijn selecteerbaar en atomen kunnen wel of niet verwijderd worden.
	Toon de Library Tree.
	Toon de Model Tree.
	<i>Toon de Atom Editor.</i>
	Toon de Run Control.
	Toon de Clock.
	<i>Toon de GUI Builder.</i>
	Toon de 4DScript Interact window.
	Toon de 4DScript overview.
	Toon de text file editor.
	Toon de Summary Report.
	Toon de Graph window.
	Toon de Help.





Voeg atomen toe	Toelichting
Basic modeling	
	Voeg een <b>Source</b> atoom toe aan het Model.
	Voeg een <b>Queue</b> atoom toe aan het Model.
	Voeg een <b>Server</b> atoom toe aan het Model.
	Voeg een <b>Sink</b> atoom toe aan het Model.
	Voeg een <b>Container</b> atoom toe aan het Model.
	Voeg een <b>Node</b> atoom toe aan het Model.
Processes	
	Voeg een <b>Assembler</b> atoom toe aan het Model.
	Voeg een <b>Unpack</b> atoom toe aan het Model.
	Voeg een <b>Splitter</b> atoom toe aan het Model.
	Voeg een <b>Multiserver</b> atoom toe aan het Model.
Product transformation	
	Voeg een <b>Single transform</b> atoom toe aan het Model.
	Voeg een <b>Multiple transform</b> atoom toe aan het Model.
Storage	
	Voeg een <b>Warehouse</b> atoom toe aan het Model.
Transport	
	Voeg een <b>Accumulating conveyor</b> atoom toe aan het Model.
	Voeg een <b>Non-Accumulating conveyor</b> atoom toe aan het Model.
	Voeg een <b>Transporter</b> atoom toe aan het Model.
	Voeg een <b>Dispatcher</b> atoom toe aan het Model.
	Voeg een <b>Destinator</b> atoom toe aan het Model.
	Voeg een <b>Portal Crane</b> atoom toe aan het Model.
	Voeg een <b>Robot</b> atoom toe aan het Model.
Network	
<i>Atomen die nodig zijn bij het bouwen van een Netwerk voor een Advanced Transporter of een Operator.</i>	
	Voeg een <b>Network node</b> atoom toe aan het Model.
	Voeg een <b>Node Manipulator</b> atoom toe aan het Model.
	Voeg een <b>Network Controller</b> atoom toe aan het Model.
Operators	
	Voeg een <b>Operator</b> atoom toe aan het Model.
	Voeg een <b>Team</b> atoom toe aan het Model.
	Voeg een <b>Call Operator</b> atoom toe aan het Model.
	Voeg een <b>Free Operator</b> atoom toe aan het Model.
Time	
	Voeg een <b>Arrival List</b> atoom toe aan het Model.
	Voeg een <b>User Events</b> atoom toe aan het Model.

Tools	
	Voeg een <b>Composition Container</b> atoom toe aan het Model.
	Voeg een <b>Empirical Distribution</b> atoom toe aan het Model.
Availability	
	Voeg een <b>Availability Control</b> atoom toe aan het Model.
	Voeg een <b>MTBF MTTR availability</b> atoom toe aan het Model.
	Voeg een <b>Time schedule availability</b> atoom toe aan het Model.
Flow control	
	Voeg een <b>Lock</b> atoom toe aan het Model.
	Voeg een <b>Unlock</b> atoom toe aan het Model.
	Voeg een <b>Condition Control</b> atoom toe aan het Model.
	Voeg een <b>Notify Router</b> atoom toe aan het Model.
Visualization	
	Voeg een <b>Text Box</b> atoom toe aan het Model. Zet hiermee statische tekst in het 2D Model.
	Voeg een <b>Bitmap Box</b> atoom toe aan het Model. Voeg een plaatje toe aan het 2D Model
Results	
	Voeg een <b>Monitor</b> atoom toe aan het Model.
	Voeg een <b>Status indicator</b> atoom toe aan het Model.
	Voeg een <b>Status monitor</b> atoom toe aan het Model.
	Voeg een <b>Status histogram</b> atoom toe aan het Model.
	Voeg een <b>Status monitor stacked bar</b> atoom toe aan het Model.
	Voeg een algemeen <b>Circle diagram</b> atoom toe aan het Model.
	Voeg een algemeen <b>Histogram</b> atoom toe aan het Model.
	Voeg een <b>Scatterplot</b> atoom toe aan het Model.
Data	
	Voeg een <b>Table</b> atoom toe aan het Model.
<i>Data atomen voor het maken van een DDE verbinding met andere applicaties.</i>	
	Voeg een <b>Word</b> atoom toe aan het Model.
	Voeg een <b>Excel</b> atoom toe aan het Model.
	Voeg een <b>Database</b> atoom toe aan het Model.



Tools	<i>Uitleg</i>
	Toon de Autofit tool.
	Toon het Label scherm. Dit scherm toont een tabel waarin de Labels en hun waarde van het geselecteerde atoom in de Model Layout en van de atomen die hierin bevat zijn weergeeft.
	<i>Toont een scherm waarin eenvoudig attributen en corresponderende oproepfuncties toegevoegd kunnen worden aan atomen.</i>

Debugging tools	<i>Uitleg</i>
	Toon de Tracer Window.
	Toon de Error Monitor.
	Toon de Watches.
	Toon de Debugger.

Zoeken	<i>Uitleg</i>
	Vindt het geselecteerde Tree atoom in de Model Layout.
	Vind het atoom geselecteerd in de Model Layout, d.w.z. de AnimAtom, in de Model Tree
	Vind het moeder atoom van het geselecteerde Tree Atoom.
	Gereedschap waarmee een stuk tekst, code of een atoom gezocht kan worden in een specifieke omgeving bijvoorbeeld de library of in het Model.

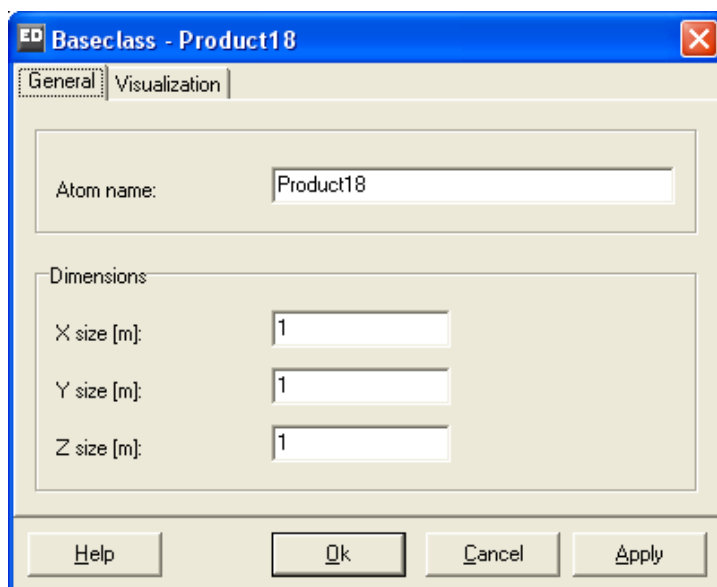
## Bijlage 2            Beschrijving van enkele atomen

---

1	Het Product atoom .....	11
2	Het Source atoom.....	12
3	Het Server atoom .....	19
4	Het Queue atoom .....	23
5	Het Sink atoom .....	25
6	Het Container atoom .....	26
7	Het Assembler atoom.....	28
8	Het Unpack atoom .....	30
9	Het Multiservice atoom.....	32
10	Het Lock atoom.....	34
11	Het Unlock atoom .....	37

# 1 HET PRODUCT ATOOM

---



**Figuur 1-1: Het Product atoom**

Het Product Atoom wordt gebruikt om de fysieke stromen in Enterprise Dynamics te modelleren. Deze stromen kunnen bestaan uit producten, goederen, documenten of personen. De volgende instellingen kunnen worden aangepast:

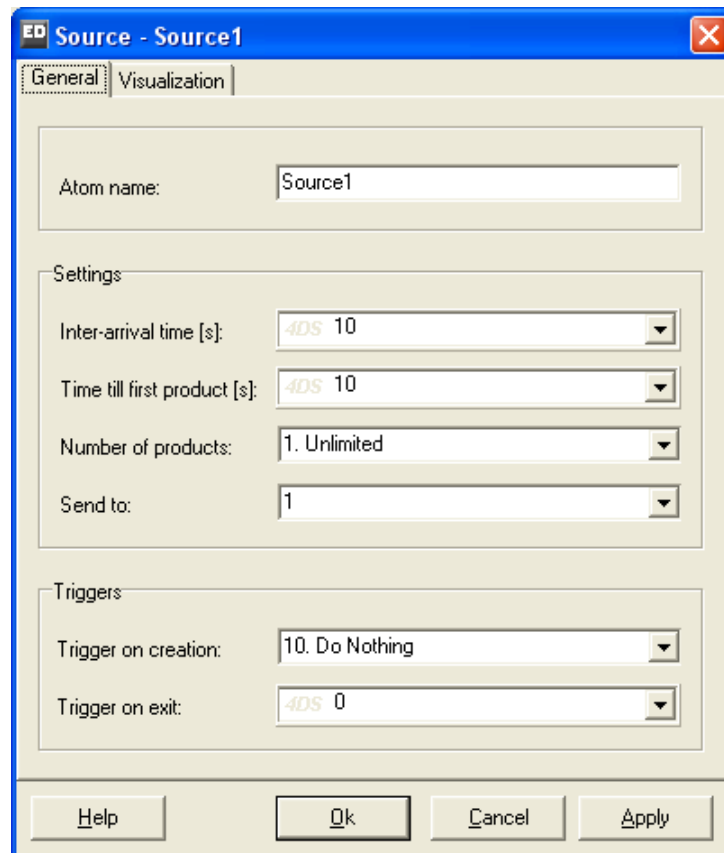
Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Size X*  
De afmeting van het atoom in de x richting (lengte in meters).
- *Size Y*  
De afmeting van het atoom in de y richting (breedte in meters).
- *Size Z*  
De afmeting van het atoom in de z richting (hoogte in meters).

Op het Visualization tabblad:

- *2D Icon*  
Het figuur dat het Product atoom weergeeft in het 2D scherm.
- *Show 2D Icon*  
Hiermee kan worden opgegeven of het 2D Icon wordt getoond. Als de optie is aangevinkt (standaardinstelling) wordt het icoon getoond.
- *3D Icon*  
Het figuur dat het Product atoom weergeeft in het 3D scherm.
- *Color*  
De kleur van het atoom.

## 2 HET SOURCE ATOOM



Figuur 2-1: Het Source atoom

Het Source Atoom zorgt ervoor dat atomen, meestal producten, met een bepaalde intensiteit het model binnen komen en fungeert dus als product- of klantgenerator. Meestal is dit atoom het eerste onderdeel van een model.

De volgende instellingen kunnen worden aangepast:

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Inter-arrival time*  
De tussenaankomsttijd (t.a.t.). Dit is de tijd die er tussen twee binnenkomsten van een Product Atoom aanwezig dient te zijn. Deze tijd wordt gemeten in seconden en kan constant zijn, maar ook via een kansverdeling worden opgegeven. Bekijk via het pulldownmenu (klikken op de driehoek rechts) een aantal mogelijke kansverdelingen met parameters en ingevulde waarden.
- *Time till first product*  
Aankomstijdstip van het eerste product. Na deze aankomst geldt de kansverdeling van de *Inter-arrival time*. Defaultwaarde is 10 seconden; geef -wanneer alle producten dezelfde tussenaankomsttijd hebben- hier dezelfde uitdrukking als bij de *Inter-arrival time*.

- *Number of products*

Met deze optie is het aantal producten uit de Source in te stellen. Er zijn twee mogelijkheden:

1. Unlimited (standaardinstelling)
2. Generate maximum 100 products

Het is nu dus mogelijk het precieze aantal aankomsten in het model te kiezen. Dit is vergelijkbaar met de functionaliteit van het –meer algemene- Lockatoom.

- *Send to*

Hier wordt het nummer van het uitgangskanaal opgegeven waardoor andere atomen (meestal producten) dit atoom verlaten. Het resultaat van deze optie moet altijd een getal tussen 1 en het totale aantal uitgangskanalen van het atoom opleveren. Als het resultaat 0 is, dan wordt er nooit verzonden. Als een atoom geblokkeerd wordt omdat de invoerkanalen van de ontvangende kanalen gesloten zijn, dan wordt het 'send to' statement pas opnieuw geëvalueerd indien er een toestandswijziging heeft plaatsgevonden waardoor verzending mogelijk is.

Er kan door de gebruiker een getal worden ingevuld, of gekozen worden uit één van de onderstaande opties:

- 1: *Specific channel: always send to channel 1.*

Hierbij zal het Product Atoom altijd naar een vast uitgangskanaal worden gestuurd.

- 2: *An open channel (First channel first): search, starting from the first channel, and send to the first open channel found.*

Hierbij wordt het Product Atoom naar het eerste open kanaal gestuurd dat Enterprise Dynamics tegenkomt, startend vanaf het eerste uitgangskanaal naar twee en hoger.

- 3: *An open channel (Last channel first): search, starting from the last channel, and send to the first open channel found.*

Het product wordt naar het eerste open kanaal gestuurd dat Enterprise Dynamics tegenkomt, startend vanaf het laatste uitgangskanaal naar het voorlaatste en lager.

- 4: *A random open channel: choose a random channel from all the open output channels.*

Er wordt door Enterprise Dynamics een willekeurig kanaal uit de open kanalen gekozen.

Bij lange simulaties leidt dit tot gelijke bezettingsgraden van bijvoorbeeld een groep servers waar naar verwezen wordt

- 5: *By percentage: 90% of products go to channel 1, the remaining percentage go to channel 2.*

Hierbij wordt een bepaald percentage van de producten naar het ene kanaal gestuurd en de rest naar een ander kanaal. De gebruiker kan de kanalen en het percentage opgeven.

- 6: *By atom name: if the atom name of the 1st atom in the queue matches Atom-Name then send to channel 1 else 2.*

Hierbij worden de atomen doorgestuurd op basis van hun naam. Als de naam overeenkomt met de naam die de gebruiker opgeeft worden de producten naar

kanaal **1** gestuurd en anders naar kanaal **2**. De kanaalnummers en de atoomnaam kunnen door de gebruikers worden aangepast.

- 7: *By label value (direct): the channel number is written directly on the label named **LabelName** of the **1st** atom in the queue. If the label value is 0 then send to channel **1**.*

De atomen worden doorgestuurd op basis van de waarde van een label met een door de gebruiker opgegeven naam. De waarde van de variabele komt direct overeen met die van het uitgangskanaal, behalve als de waarde nul is, dan wordt een gespecificeerde uitgang gebruikt. Let erop dat wanneer er gezocht wordt naar een label dat op het atoom niet aanwezig is, dit ook de waarde nul oplevert.

- 8: *By label value (conditional): if the value on the label named **LabelName** of the **1st** atom in the queue is < the value **1** then send to channel **1** else **2**.*

Ook hier is de waarde van een opgegeven label bepalend voor het te gebruiken uitgangskanaal. Als de waarde van het atoom kleiner is dan **1**, wordt het atoom naar kanaal **1** gestuurd, anders naar uitgang **2**. Alle waarden en de vergelijking (kleiner dan, groter dan en gelijk aan) zijn aanpasbaar.

- 9: *By label text: if the text on the label named **LabelName** of the **1st** atom in the queue matches **text** then send to channel **1** else **2**.*

Als de waarde van een opgegeven label gelijk is aan een bepaalde tekst wordt het atoom naar kanaal **1** gestuurd, anders naar kanaal **2**. De tekst en de kanaalnummers zijn aanpasbaar.

- 10: *Conditional statement: If **1** is > than 0 then send to channel **1** else send to channel **2**.*

Als een bepaalde waarde groter is dan een andere waarde wordt het atoom naar kanaal **1** gestuurd, anders naar kanaal **2**. De vergelijking en de kanaalnummers zijn aanpasbaar.

- 11: *By icon name: if the icon name of the **1st** atom in the queue matches **IconName** then send to channel **1** else **2**.*

Als de naam van de icoon van het atoom overeenkomt met een opgegeven naam wordt het atoom naar kanaal **1** gestuurd, anders naar kanaal **2**. De icoonnaam en de kanaalnummers zijn aanpasbaar.

- 12: *By icon number: if the icon number of the **1st** atom in the queue is = the value **1** then send to channel **1** else **2**.*

Als het icoonnummer van het atoom **gelijk is** aan **1** dan wordt het atoom naar kanaal **1** gestuurd, anders naar kanaal **2**. De vergelijking en de kanaalnummers zijn aanpasbaar.

- 13: *Round robin: all outputchannels are used in rotation. If channel is closed, then wait till open.*

Alle uitgangskanalen worden om de beurt gebruikt. Als een kanaal niet geopend is, wacht Enterprise Dynamics tot het kanaal opengaat.

- 14: *Lowest queue: Send to the channel connected to the atom with the lowest queue.*

Het atoom wordt naar het uitgangskanaal gestuurd met de kortste wachtrij. Bij een keuze uit gelijke wachtrijen wordt het uitgangskanaal met het laagste nummer gekozen.

- 15: *Largest queue: Send to the channel connected to the atom with the largest queue.*

Het atoom wordt naar het kanaal gestuurd met de langste wachtrij. Bij een

keuze uit gelijke wachtrijen wordt het uitgangskanaal met het laagste nummer gekozen.

- 16: *Lookup table: Send to the channel specified in row 1 column 2 of global table named **table1**.*

Stuurt het atoom naar het kanaal dat is opgegeven in de 1e rij en de 2e kolom van een tabel. De rij- en kolomnummers en de tabelnaam zijn aan te passen. Let op dat de tabel zich als apart atoom in het model moet bevinden!

- 17: *Round robin if available: all output channels are used in rotation if channel is available. If channel is closed, then next available channel is chosen.*

Alle kanalen worden om de beurt gebruikt, maar als het aan de beurt zijnde kanaal gesloten is, wordt er naar het volgende kanaal gegaan.

- 18: *Matching icon number or empty: Sends to a queue containing products of same icon. If no icons match, then sends to first empty queue starting with last output channel.*

Stuurt atomen zo door dat ze steeds in een rij terechtkomen van atomen met hetzelfde icoonnummer. Wanneer hetzelfde icoonnummer niet voorkomt, wordt het atoom naar de eerste lege Queue gestuurd, waarbij Enterprise Dynamics begint te zoeken bij het laatst gebruikte kanaalnummer.

- 19: *Lowest queue of next two atoms: Sends to the output channel connected to the lowest queue, where lowest queue takes into account the next TWO atoms.*

Enterprise Dynamics stuurt het atoom naar het kanaal waarvan de inhoud tezamen met de inhoud van het daaropvolgende atoom (via uitgangskanaal 1!) het kleinste is. Er wordt dus twee atomen ver gekeken om deze inhoud te bepalen. Een typisch voorbeeld hiervan is een atoom dat naar drie gescheiden rijen wil sturen met daarachter een server. Met deze regel wordt vermeden dat producten doorgestuurd worden naar een server die aan een bediening bezig is, terwijl de overige servers leeg zijn.

- 20: *By user: enter your own 4DScript expression resulting in a value between 1 and the number of channels: 1. You can press the small button for the 4DScript editor.*

Een door de gebruiker geschreven 4DScript code die het uitgangskanaal oplevert. Door op het kleine vierkante knopje te drukken naast de tekst, verschijnt de 4DScript editor.

- 21: *Random channel: randomly choose a channel. If the channel is open then send to it, otherwise choose again when any channel opens.*

Er wordt door Enterprise Dynamics een willekeurig kanaal gekozen. Wanneer dit kanaal geopend is, wordt het product naar dat kanaal gezonden. Wanneer dit kanaal echter gesloten is, wordt er pas opnieuw geloot wanneer een van de andere uitgangskanalen open gaat.

- *Trigger on Creation*

Het commando in dit invoerscherm wordt uitgevoerd als een atoom het model binnenkomt. Naast een zelfgeschreven 4DScript expressie kan er ook uit één van de onderstaande opties worden gekozen:

- 1: *Assign label: products are assigned a label named **LabelName** with a value of **1**.*  
De producten krijgen een label met een opgegeven naam en een bepaalde waarde. De labelnaam en de waarde zijn instelbaar.
- 2: *Auto Name: a counter is added to the end of each product's name.*  
Aan de naam van de producten wordt een teller toegevoegd. Het eerste product heet dan bijvoorbeeld Product1 en het tweede Product2.
- 3: *Random icons: products are assigned a random icon number between **2** and **6**.*  
Aan elk product wordt een willekeurig icoon toegewezen met een icoonnummer tussen twee opgegeven waarden.
- 4: *Set Size: product dimensions are set to: **X= 50 cm, Y= 40 cm, Z= 30 cm**.*  
De afmetingen van het product worden aangepast aan de opgegeven waarde.
- 5: *Random Size: product dimensions are randomly set within the following ranges: **X= 50 to 100 cm, Y= 50 to 100 cm, Z= 50 to 100 cm**.*  
De afmetingen van een product worden aangepast aan willekeurig gekozen waarden binnen een opgegeven bereik.
- 6: *Set Color: products are set to the **colorpurple**.*  
De kleur van een product wordt veranderd in een opgegeven kleur. Let er op dat een kleur in 4DScript opgegeven wordt door het woord color direct te laten volgen door de kleur. Dus colorpurple is het commando voor paars. In plaats van het commando colorpurple kan ook het nummer van de kleur worden opgegeven.
- 7: *Random color: products are assigned a random color.*  
De producten krijgen een willekeurige kleur toegewezen.
- 8: *Random Size and Color: products are assigned a random color and its dimensions are randomly set within the following ranges: **X=50 to 100 cm, Y= 50 to 100 cm, Z= 50 to 100 cm**.*  
De producten krijgen zowel een willekeurige kleur als een willekeurige maat (binnen opgegeven marges).
- 9: *Outline: display the products as a simple outline, not its icon.*  
Zorgt ervoor dat de icoon van een product niet meer zichtbaar is, maar nog slechts zijn omtrek.
- 10: *Do Nothing.*  
Er gebeurt niets. Dit is de standaardinstelling.



- *Trigger on Exit*

Het commando in dit invoerscherm wordt uitgevoerd op het moment dat een product het atoom verlaat. Het is hierbij mogelijk om zelf een 4DScript expressie op te geven of gebruik te maken van de voorgedefinieerde expressies. De vraagtekens betekenen steeds dat een gebruiker zelf iets kan invoeren.

Er kan uit de volgende mogelijkheden gekozen worden:

1: *setlabel([?],?,i).*

Met dit 4DScript commando wordt er een label toegevoegd aan het atoom dat de Source verlaat. De code is: *setlabel([labelnaam],waarde,i).*

Voorbeeld

Om aan het product een label 'Voltooid' toe te wijzen met de waarde 1 gebruikt je de volgende code: *setlabel([Voltooid],1,i).*

De letter i verwijst naar het *involved* atoom. Dit is het atoom dat de handeling (het verlaten van de Source) ondergaat. Wanneer er een label op de Source zelf geplaatst zou moeten worden, kan de i vervangen worden door een c (van *current*).

Voorbeeld

*Setlabel([bewerkingstijd], uniform(25,45),i)* definieert een label bewerkingstijd op het product met een uitkomst uit de uniforme verdeling tussen 25 en 45 seconden. Deze uitkomst kan later bijvoorbeeld op een server als bewerkingstijd aangeroepen worden.

2: *set(name(i),[?]).*

Verandert de naam van het atoom dat de Source verlaat. Het '?' moet vervangen worden door de naam die het atoom moet gaan krijgen.

3: *set(icon(i),?).*

Verandert het icoon van het atoom in het icoon met het nummer '?'.

4: *set(icon(i),iconbyname([?])).*

Verandert het icoon in het icoon met de naam '?'.

5: *set(color(i),coloryellow).*

Verandert de kleur van het atoom in een opgegeven kleur. De kleuren in Enterprise Dynamics zijn op te geven met hun nummer of door het commando *color* direct gevolgd door de kleur, dus bijvoorbeeld *colored*.

6: *setsize(?,?,?,i).*

Verandert de maat van het atoom in de maat x,y,z.

7: *setloc(?,?,?,i).*

Geeft het atoom de nieuwe locatie x,y,z

8: *freeoperators(atombyname([Team],model),i).*

Zorgt dat de Operator atomen weer gebruikt kunnen worden. Vervang 'Team' door de naam van het Teamatoom. Deze mogelijkheid zal slechts door gevorderde gebruikers worden toegepast.

9: *if(=(?,?),?,?).*

Een voorwaardelijke vergelijking. Bij invullen van een voorbeeldcode ontstaat: *if(=(stelling1,stelling2),opdracht1,opdracht2).*

Als stelling 1 en 2 gelijk zijn, wordt opdracht1 uitgevoerd, anders opdracht2. Opdracht2 kan ook weg gelaten worden.

10: *if*(=(*label*([?],*i*),?),?,?).

Een voorwaardelijke vergelijking, waarbij gekeken wordt naar de waarde van een label.

Voorbeeld

Op de volgende manier kunnen we –indien het label Afkeur een 1 heeft- alle afgekeurde producten rood kleuren en alle goedgekeurde producten groen:

*if*(=(*label*([Afkeur],*i*),1),*set*(*color*(*i*),*colorred*),*set*(*color*(*i*),*colorgreen*)).

11: *if*(*comparetext*(*name*(*i*),[?]),?,?).

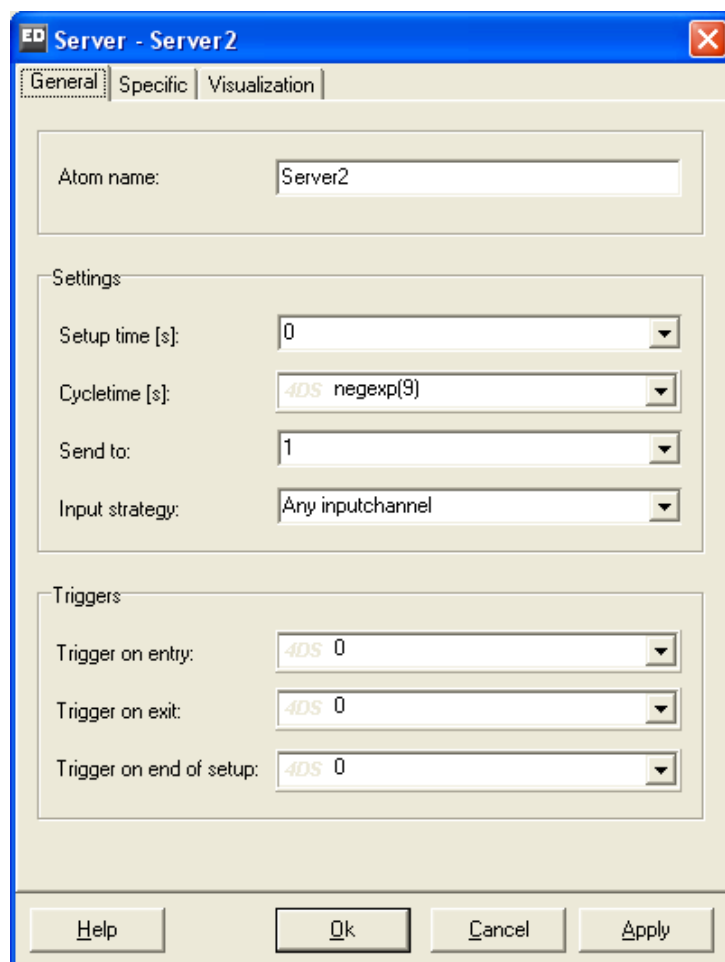
Een voorwaardelijke vergelijking waarbij gekeken wordt naar de naam van het atoom. De werking is verder gelijk aan die van mogelijkheid 10.

Voor een meer uitgebreide uitleg over de 4DScript commando's kan er in bijlage 3 of de helpbestanden van Enterprise Dynamics gekeken worden.

Op het Visualization tabblad

- *Icon*  
Het figuur dat het Source atoom weergeeft in het 2D scherm.
- *3D Icon*  
Het figuur dat het Source atoom weergeeft in het 3D scherm.

### 3 HET SERVER ATOOM



**Figuur 3-1: Het Server atoom**

De Server wordt gebruikt voor het modelleren van operaties die tijd kosten zoals het bewerken van producten op een machine of het afrekenen van een klant bij een kassa.

De Server kan dus een machine, een loket, een bediende of een andersoortige bewerkingsplaats voorstellen. Naast bewerkingstijden kunnen zaken als omsteltijden of de gelijktijdige bewerking van meerdere producten gedefinieerd worden.

De volgende waarden zijn instelbaar:

Op het General tabblad:

- *Atom Name*  
De naam van het atoom.
- *Setup time*  
Hier kan insteltijd of omsteltijd gedefinieerd worden: de tijd die nodig is voordat aan de eigenlijke bewerking kan worden begonnen. Denk hierbij aan het schoonmaken van machines, anders instellen van machines voor nieuwe producten enzovoorts. Door op het driehoekje te drukken verschijnt er een lijst met opties, waaronder het instellen voor ieder product of alleen bij producten van een ander type. Naast die opties is het ook mogelijk eigen 4DScript op te stellen.

- *Cycletime*  
De tijd die nodig is om het product te bewerken. Door op het pijltje te klikken verschijnen er een aantal voorgedefinieerde 4DScript regels.  
Belangrijk: indien er sprake is van groepsgewijze bewerking van producten (de zgn. batchverwerking) heeft de bewerkingstijd betrekking op de hele groep en niet op ieder individueel product. Bovendien start de bewerking pas indien de groep compleet is.
- *Send to*  
Hier wordt het kanaal ingevuld waarheen het product gestuurd moet worden. Voor meer uitleg: zie de toelichting bij Send to van de Source.
- *Input strategie*  
Hiermee wordt de toegang tot een atoom vanuit voorafgaande atomen (met uitgangskanalen naar dit bewuste atoom) geregeld. Door de inputstrategie worden zowel één of meerdere ingangen vrijgegeven alsook via een bepaalde volgorde bekeken via welk kanaal als eerste producten mogen worden ontvangen.  
De inputstrategie is het beste te vergelijken met de regelstrategie van een verkeersregelinstallatie waarbij onafhankelijk van het verkeersaanbod vanuit de toestand Rood overgeschakeld wordt naar Groen voor een of meerdere toevoerwegen en waarbij tevens de voorrang van afwikkeling van deze toevoerwegen is bepaald.

Let op: de eerste drie inputstrategieën openen alle ingangskanalen, de laatste twee openen steeds één ingangskanaal!

1: *Any inputchannel.*

Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met het laagste nummer als ingangskanaal voorrang. Zolang er dus steeds producten via het eerste kanaal binnenkomen, komen de andere kanalen niet aan bod!

2: *Largest queue.*

Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met de grootste rij of inhoud voorrang. Merk op dat in het geval van meerdere even lange wachtrijen steeds wordt gekozen voor het ingangskanaal met het laagste nummer.

3: *Longest waiting.*

Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met de hoogste gemiddelde verblijftijd voorrang. In het geval van meerdere atomen met eenzelfde gemiddelde verblijftijd steeds wordt gekozen voor het ingangskanaal met het laagste nummer. Merk op dat dit dus niet per definitie betekent dat de wachtrijen in aantallen ongeveer even lang worden, zoals bij de vorige optie wel het geval was.

4: *Round robin.*

Opent eerst het eerste ingangskanaal en wacht op toezending van een product via dit ingangskanaal. Bij de volgende cyclus komt het tweede ingangskanaal aan de beurt enz. Wanneer het laatste ingangskanaal geweest is, wordt de procedure met het eerste ingangskanaal hervat.

5: *Channel 1.*

Hierbij wordt een vast ingangskanaal opgegeven dat gebruikt moet worden. In dit geval mogen dus alleen producten via ingangskanaal 1 naar binnen. Merk op dat bij het eerste product dat binnenkomt deze regel nog niet geldt en alle kanalen openstaan.

- *Trigger on Entry*

Hier kan een commando worden ingevuld dat wordt uitgevoerd als een product de Server binnenkomt.

Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.

- *Trigger on Exit*

Hier kan een commando worden ingevuld dat wordt uitgevoerd als een product de Server verlaat.

Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.

- *Trigger on end of setup*

Hier kan een commando worden ingevuld dat wordt uitgevoerd als de Setup time verlopen is. Er zijn enkele voor gedefinieerde regels maar ook kan een eigen regel geschreven worden in 4DScript.

Op het Specific tabblad:

- *Batch (B)*

Hier kan de batchgrootte worden opgegeven. Standaard staat deze op 1.

- *Batch rule*

Hier wordt opgegeven hoe de batches moeten worden samengesteld. Er zijn drie opties:

- 1: *B in, 1 out (the first).*

Zodra het aantal producten dat de Server is binnengekomen gelijk is aan de batchgrootte, wordt het voorste product doorgestuurd naar het volgende atoom. De andere producten verdwijnen.

- 2: *B in, B out.*

Zodra het aantal producten dat de Server is binnengekomen gelijk is aan de batchgrootte, worden de producten doorgestuurd naar het volgende atoom. De Server laat pas weer producten toe als alle producten uit de batch de Server verlaten hebben.

- 3: *1 in, B out (copies of in).*

Elke keer wanneer een product de Server binnenkomt, verlaten er zoveel producten het atoom als er in het invoerscherm Batch zijn opgegeven. De producten zijn allemaal kopieën van het atoom dat de Server binnenkwam. De Server laat pas weer een product binnen als alle andere producten het atoom verlaten hebben.

- *Busy time*

Wanneer deze optie is aangevinkt, wordt er bij de “Mean Time Between Failure” opties alleen uitgegaan van de tijd dat de Server daadwerkelijk in gebruik is geweest en niet van de totale simulatietijd.

- *MTTF*

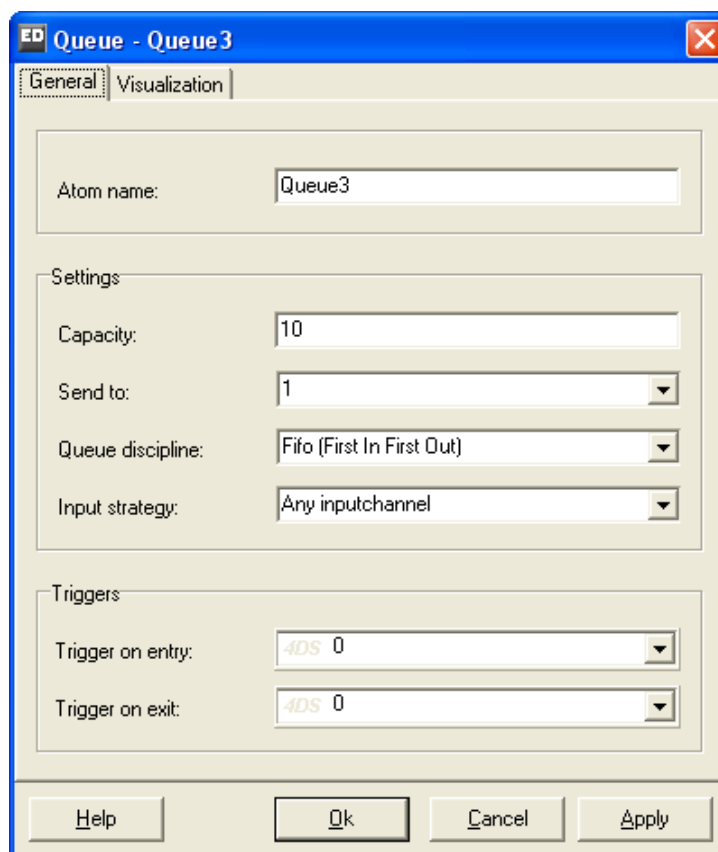
Dit staat voor Mean Time To Failure, de gemiddelde tijd die verstrijkt vanaf een reparatie tot de volgende storing. In het invoerveld kan deze gemiddelde tijd tussen twee storingen aan de Server worden opgegeven. De tijd wordt opgegeven in seconden.

- *MTTR*  
Staat voor Mean Time To Repair. In het invoerveld kan de gemiddelde tijd worden opgegeven die het kost om de Server weer te laten functioneren.
- *MCBF*  
Staat voor Mean Cycles Between Failure. In het invoerveld kan het gemiddeld aantal cycli worden opgegeven tussen twee storingen. Bij MCBF zit er dus niet een bepaalde tijd tussen twee storingen, maar een bepaald aantal batches.  
Nb. wanneer zowel het invoerveld MTBF en MCBF zijn ingevuld zullen er op beide momenten storingen optreden.
- *MTTR for cycles*  
Dit is het Mean Time To Repair invoerveld dat moet worden ingevuld indien er een bepaald aantal batches tussen een storing zit.
- *Trigger on Breakdown*  
Hier kan een commando worden ingevuld dat wordt uitgevoerd direct voor de start van een Breakdown periode van de Server.  
Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.
- *Trigger on Repair*  
Hier kan een commando worden ingevuld dat wordt uitgevoerd direct voor de start van de Repair periode van de Server.  
Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Server atoom weergeeft in het 2D scherm.
- *3D Icon*  
Het figuur dat het Server atoom weergeeft in het 3D scherm.
- *Main color*
- *Second color*

## 4 HET QUEUE ATOOM



Figuur 4-1: het Queue atoom

Het Queue atoom zit producten in een wachtrij als het volgende atoom bezet is. De volgende instellingen kunnen worden aangepast:

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Capacity*  
De capaciteit van de Queue. Als er zich evenveel producten in de wachtrij bevinden als er zijn opgegeven bij de capaciteit kunnen er geen nieuwe producten meer in de wachtrij worden geplaatst.
- *Send to*  
Hier wordt opgegeven naar welk uitgangskanaal de producten gestuurd worden. Voor meer uitleg: zie de Send to van het Source atoom.
- *Queue discipline*  
De wijze waarop de producten in de wachtrij geplaatst worden. Er zijn daarbij de volgende instellingen mogelijk:
  - 1: *First in first out.*  
Hierdoor worden de atomen in de volgorde van binnenkomst in de Queue geplaatst.
  - 2: *Last in first out.*  
Hierdoor worden de producten steeds vooraan in de wachtrij geplaatst als ze

binnenkomen. De producten verlaten de Queue dus in de omgekeerde volgorde: de laatst aangekomene vertrekt als eerste.

3: *Random.*

Hierbij worden de producten steeds op een willekeurige plaats in de wachtrij geplaatst.

4: *Sort by **Label** Ascending.*

Hierbij worden de producten met de laagste waarde voor een bepaald label vooraan in de wachtrij geplaatst.

Let op! Wanneer de producten niet goed gesorteerd worden heeft dit waarschijnlijk te maken met een spatie voor of achter de naam van het label.

5: *Sort by **Label** Descending.*

Hierbij worden de producten met een hoogste waarde voor een bepaald label vooraan in de wachtrij geplaatst.

Let op! Wanneer de producten niet goed gesorteerd worden heeft dit waarschijnlijk te maken met een spatie voor of achter de naam van het label.

6: *User defined.*

Hierbij worden de producten op een door de gebruiker opgegeven positie in de wachtrij geplaatst.

- *Input Strategy*

Bij dit invoerscherm kan worden opgegeven welk ingangskanaal gebruikt moet worden.

Voor meer uitleg: zie de inputstrategie van het Server atoom.

- *Trigger on Entry*

Het commando uit dit invoerveld wordt uitgevoerd als een product de Queue binnenkomt.

Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom.

- *Trigger on Exit*

Het commando uit dit invoerveld wordt uitgevoerd als een product de Queue verlaat.

Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom.

Op het Visualization tabblad:

- *Icon*

Het figuur dat het Queue atoom weergeeft in het 2D scherm.

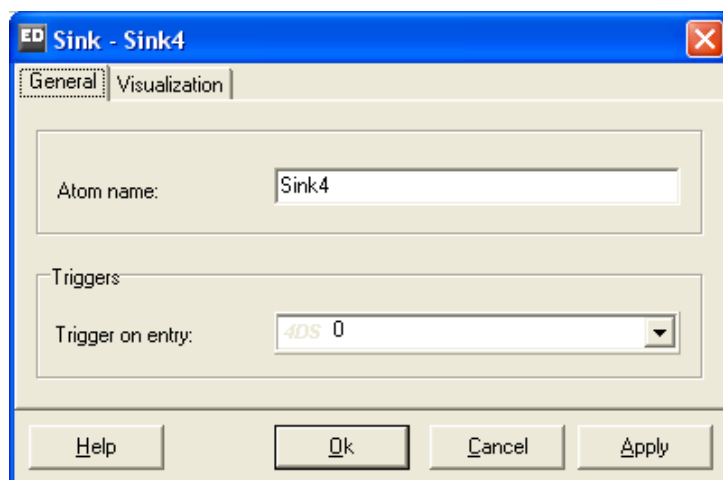
- *3D Icon*

Het figuur dat het Queue atoom weergeeft in het 3D scherm.



## 5 HET SINK ATOOM

---



**Figuur 5-1: Het Sink atoom**

Dit atoom zorgt ervoor dat producten het model weer kunnen verlaten. De volgende instellingen kunnen worden aangepast:

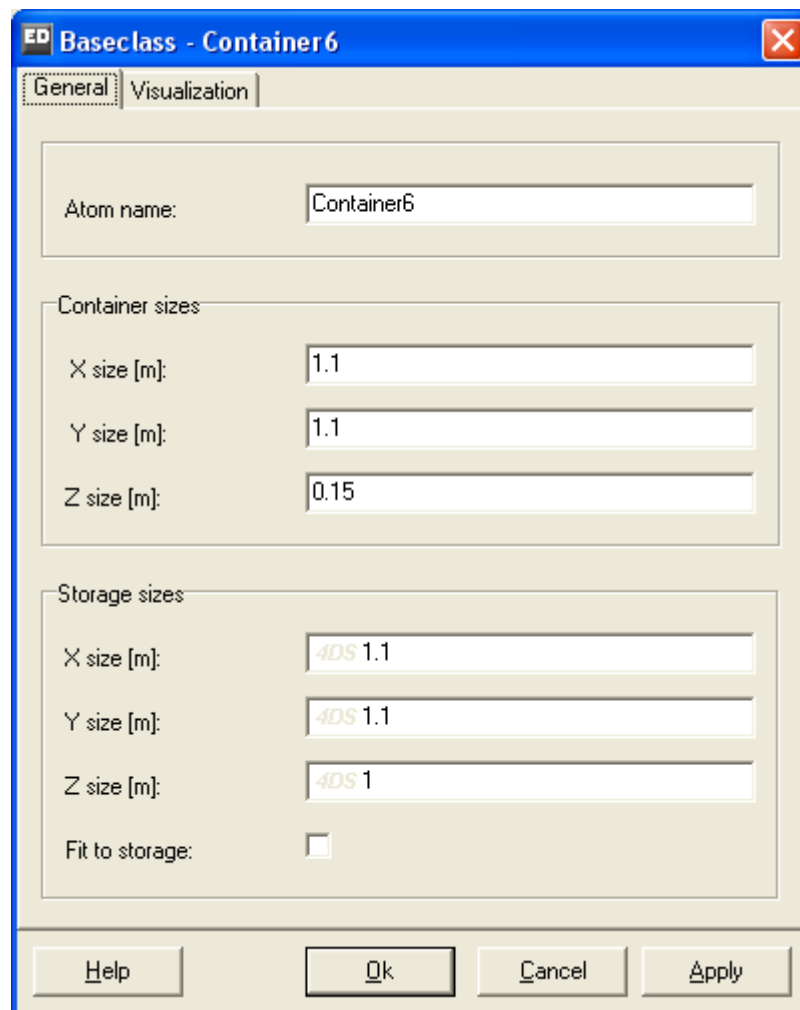
Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Trigger on Entry*  
Het commando dat in dit invoerscherm is ingevuld wordt uitgevoerd zodra een product de Sink binnenkomt.  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Sink atoom weergeeft in het 2D scherm.

## 6 HET CONTAINER ATOOM



The screenshot shows a software window titled "Baseclass - Container6". It has two tabs: "General" and "Visualization". The "General" tab is selected. Inside the window, there are three main sections:

- Atom name:** A text field containing "Container6".
- Container sizes:** Three text fields for dimensions in meters:
  - X size [m]: 1.1
  - Y size [m]: 1.1
  - Z size [m]: 0.15
- Storage sizes:** Three text fields for storage dimensions in meters:
  - X size [m]: 4DS 1.1
  - Y size [m]: 4DS 1.1
  - Z size [m]: 4DS 1

Below these fields is a checkbox labeled "Fit to storage:" which is currently unchecked. At the bottom of the window are four buttons: "Help", "Ok", "Cancel", and "Apply".

**Figuur 6-1: Het Container atoom**

Het Container atoom is speciaal ontworpen om andere atomen in te bewaren of op te plaatsen zoals een doos of pallet. Voor het Container atoom zijn een aantal standaard opties ontworpen die visualisatie vergemakkelijken. Zo zijn er speciale 3D icons gecreëerd en kan de maat van een product automatisch worden aangepast aan de maat van de Container. Het container atoom wordt, net als het Product atoom, in principe via een Source in het model geplaatst (of via het Arrival List atoom). Met behulp van een Assembler atoom kunnen producten dan in de Container worden geplaatst.

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Container X size*  
De maat van de container, gemeten in de X richting (lengte).
- *Container Y size*  
De maat van de container, gemeten in de Y richting (breedte).

- *Container Z size*  
De maat van de container, gemeten in de Z richting (hoogte).
- *Storage X size*  
Hier wordt de ruimte (in de X richting) opgegeven die nodig is om een binnenkomend product op te bergen. Wanneer er meer ruimte wordt opgegeven als het product zelf nodig heeft, ontstaat er lege ruimte tussen de producten.
- *Storage Y size*  
Hier wordt de ruimte (in de Y richting) opgegeven die nodig is om een binnenkomend product op te bergen. Wanneer er meer ruimte wordt opgegeven als het product zelf nodig heeft, ontstaat er lege ruimte tussen de producten.
- *Storage Z size*  
Hier wordt de ruimte (in de Z richting) opgegeven die nodig is om een binnenkomend product op te bergen. Wanneer er meer ruimte wordt opgegeven dan het product (zie het Productatoom!) zelf nodig heeft, ontstaat er lege ruimte tussen de producten.
- *Fit to storage*  
Door deze optie aan te vinken wordt de maat van een product zo aangepast, dat deze precies overeenkomt met de gedefinieerde Storage size.

### **Let op!**

Er is sprake van 3 soorten afmeting: de afmeting van het product (zie Productatoom), de afmeting van de Container (Container size) en de Storage size. Met de Storage size kan de manier van verpakken (visueel) geregeld worden.

Voorbeeld:

Een pallet, Container met Container size 1x1x0.15, zal een product van grootte van 1x1x1 4 hoog stapelen als de Assembler 4 producten op een pallet plaatst.

Wanneer echter Fit to Storage aangevinkt is met Storage size 0.5x0.5x0.5 geeft dit keurig 4 dozen op een laag!

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Container atoom weergeeft in het 2D scherm.
- *3D Icon*  
Het figuur dat het Container atoom weergeeft in het 3D scherm.
- *Color*  
De kleur van het atoom.

## 7 HET ASSEMBLER ATOOM

ED Assembler - Assembler7

General Channels Visualization

Atom name: Assembler7

Settings

Cycletime [s]: 4DS lognormal(10,2)

Send to: 1

Column reference b.o.m.: 4DS 1

Display contents: ☒

Pack contents: ☒

Triggers

Trigger on entry chn. 1: 4DS 0

Trigger on entry chn. 2-n: 4DS 0

Trigger on exit: 4DS 0

Miscellaneous

Edit table

Help Ok Cancel Apply

**Figuur 7-1: Het Assembler atoom**

Dit atoom voegt atomen uit meerdere bronnen samen. Hierbij kunnen de atomen die in een ander atoom geplaatst worden, bewaard blijven of vernietigd worden.

Naast het nabootsen van echte assemblage werkzaamheden, is dit atoom ook goed bruikbaar voor het inpakken van producten in dozen of pallets en zelfs voor het samenstellen van orders.

Via het eerste ingangskanaal komt steeds de pallet, doos of order binnen en via de andere kanalen komen de producten binnen. Afhankelijk van de instellingen worden deze producten vernietigd of in het atoom geplaatst dat via het eerste ingangskanaal is binnengekomen.

- *Atom name*  
De naam van het atoom.

- *Cycletime*  
De tijd die nodig is om de producten samen te voegen. Deze tijd start op het moment dat alle benodigde atomen bij de Assembler zijn aangekomen en slaat op de totale assemblagehandeling en niet op ieder product afzonderlijk! Door op het pijltje te klikken verschijnen er een aantal voorgedefinieerde 4DScript regels.
- *Send to*  
Hier wordt opgegeven naar welk uitgangskanaal de producten gestuurd worden. Voor meer uitleg: zie de Send to van het Source atoom.
- *Column reference b.o.m. (Bill of Material)*  
Hier wordt opgegeven welke kolom uit de b.o.m. moet worden gebruikt. Er kan een getal worden ingevuld, maar ook een 4DScriptcommando dat een getal oplevert. Zodra het eerste atoom via het eerste ingangskanaal binnenkomt wordt dit veld uitgelezen en wordt vastgelegd welke kolom wordt uitgelezen als de andere atomen de Assembler binnenkomen.

#### Bill of Material (b.o.m.)

Het Assembler atoom bevat tevens een tabel met daarin de Bill of Material (b.o.m.) (klik op de button Edit Table). In de b.o.m. staat per ingangskanaal hoeveel atomen er nodig zijn om het eindproduct te assembleren. Deze b.o.m. bevat een aantal rijen en kolommen. Het aantal rijen is gelijk aan het aantal ingangskanalen. Het aantal kolommen kan de gebruiker zelf instellen, maar staat standaard op 1.

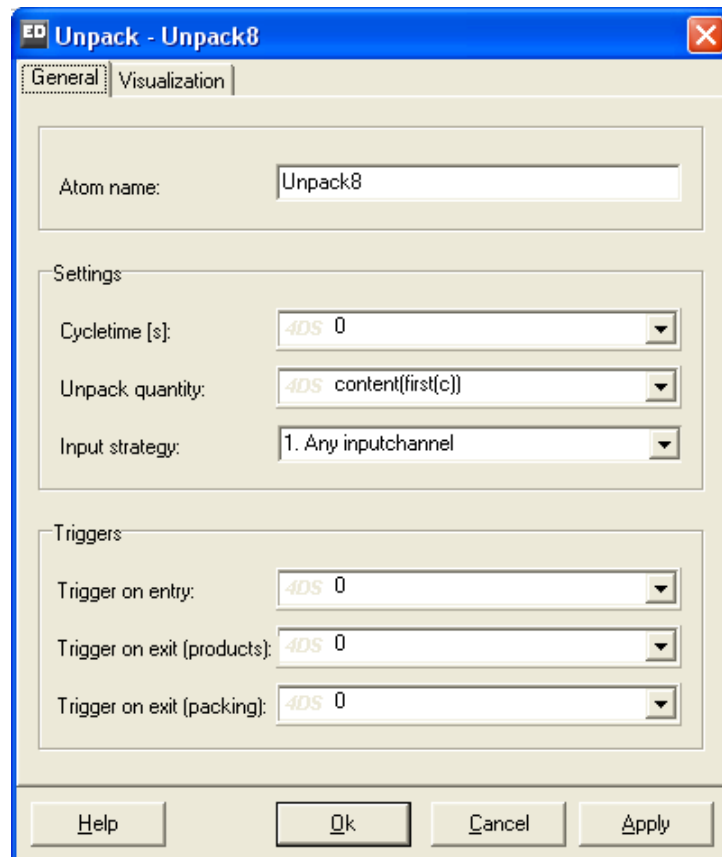
Voor elk type product dat word geassembleerd kan daardoor een eigen kolom worden aangemaakt. Op het moment dat een product via het eerste ingangskanaal binnenkomt, wordt beslist welke kolom in de b.o.m. wordt gebruikt voor de rest van de atomen die binnenkomen.

- *Display contents*  
Als deze keuzebox is aangevinkt wordt de inhoud van de Assembler getoond.
- *Pack contents*  
Als deze keuzebox is aangevinkt worden de atomen (lees: producten) in het atoom dat via kanaal 1 binnenkwam (lees: hoofdproduct of Container) geplaatst. Is de keuzebox niet aangevinkt, dan worden alle atomen die niet via kanaal 1 zijn binnengekomen vernietigd.
- *Trigger on entry channel 1*  
Het commando dat in dit invoerscherm is ingevuld wordt uitgevoerd zodra een product de Assembler binnenkomt via het eerste ingangskanaal.  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom
- *Trigger on entry channel 2..n*  
Het commando dat in dit invoerscherm is ingevuld wordt uitgevoerd zodra een product de Assembler binnenkomt via één van de overige ingangskanalen (dus niet kanaal 1)  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom
- *Trigger on exit*  
Het commando uit dit invoerveld wordt uitgevoerd als een product de Assembler verlaat.
- Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom.

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Assembler atoom weergeeft in het 2D scherm.

## 8 HET UNPACK ATOOM



**Figuur 8-1: Het Unpack atoom**

Het Unpack atoom wordt gebruikt om producten uit een Container atoom te halen. Na het uitpakken wordt het Container atoom door het tweede uitgangskanaal gestuurd en de producten door het eerste uitgangskanaal.

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Cycletime*  
De tijd die nodig is om de container leeg te halen. Deze tijd start op het moment dat de container het atoom binnenkomt en is de lostijd per container (dus niet per apart product). Door op het pijltje te klikken verschijnen er een aantal voorgedefinieerde 4DScript regels  
Voor meer uitleg: zie de cycletime van het Server atoom.
- *Unpack quantity*  
Hier wordt het aantal atomen dat moet worden uitgepakt opgegeven. Zodra dit aantal is bereikt wordt de container doorgestuurd. De standaard instelling `content(first(c))` zorgt ervoor dat alle producten uit de container worden gehaald. De 3 andere voorgedefinieerde opties zijn:  
`duniform(1,10)`  
`label(['?'],first(c))`

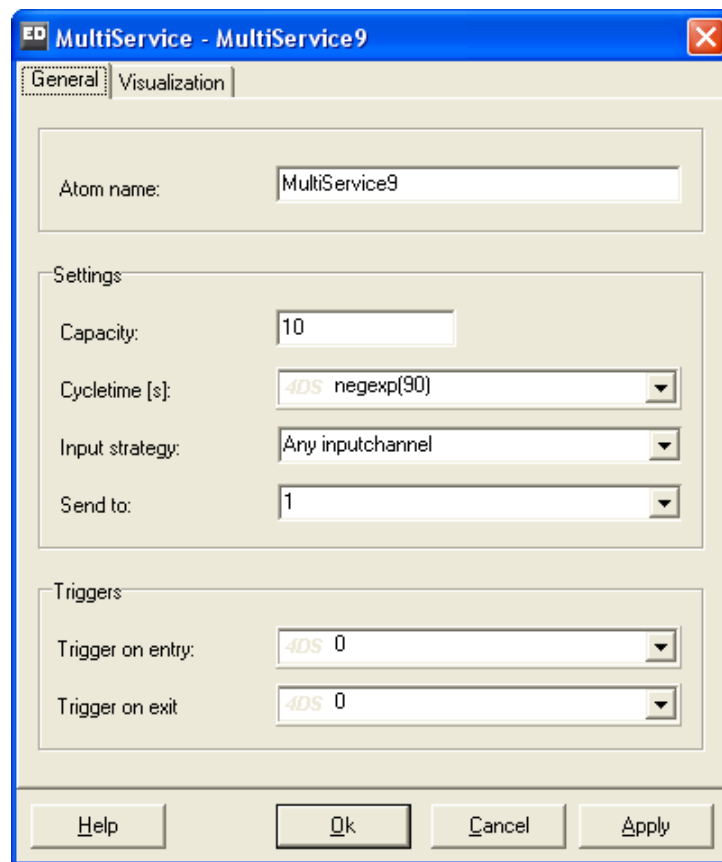
if(=(?,?),?,?)

- *Input strategie*  
Bij dit invoerscherm kan worden opgegeven welk ingangskanaal gebruikt moet worden.  
Voor meer uitleg: zie de inputstrategie van het Server atoom.
- *Trigger on entry*  
Het commando dat in dit invoerscherm is ingevuld wordt uitgevoerd zodra een container het Unpack atoom binnenkomt.  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom
- *Exit trigger products*  
Het commando uit dit invoerveld wordt uitgevoerd als een product het Unpack atoom verlaat.  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom.
- *Exit trigger packaging*  
Het commando uit dit invoerveld wordt uitgevoerd als een container het Unpack atoom verlaat.  
Voor meer uitleg: zie bij de Trigger on Exit van het Source atoom

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Unpack atoom weergeeft in het 2D scherm

## 9 HET MULTISERVICE ATOOM



**Figuur 9-1: Het MultiService atoom**

Het Multiservice atoom heeft dezelfde basisfuncties als meerdere, afzonderlijke Server atomen. Waar een Server atoom slechts 1 product tegelijk kan bewerken, kan de Multi-Service meerdere producten tegelijk bewerken. De instellingsmogelijkheden zijn niet zo uitgebreid als bij de normale server, maar alle basisfuncties zijn aanwezig.

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Capacity*  
Hier wordt het aantal producten opgegeven dat gelijktijdig bewerkt kan worden.

Zie voor onderstaande opties de beschrijving van het Server atoom:

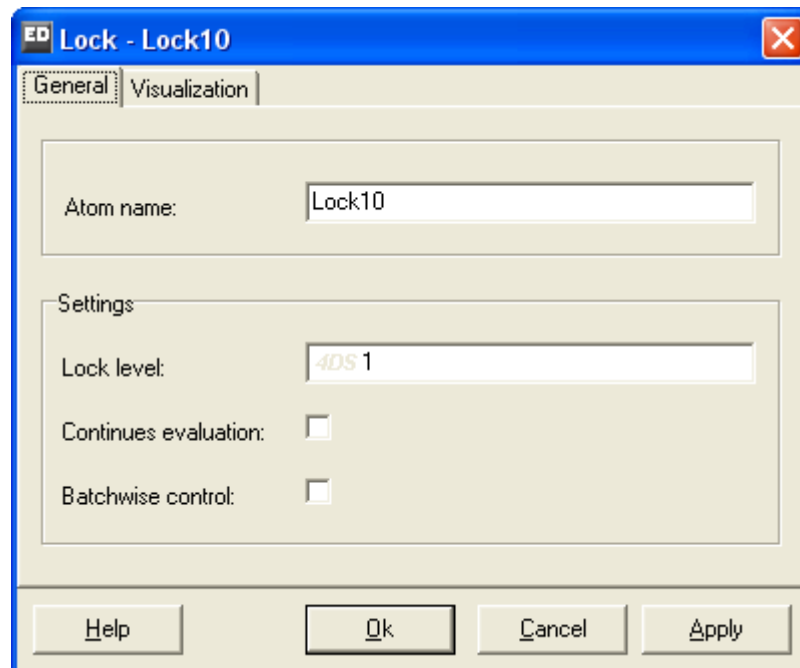
- *Cycletime*
- *Input strategy*
- *Send to*
- *Trigger on Entry*
- *Trigger on Exit*



Op het Visualization tabblad:

- *Icon*  
Het figuur dat het MultiService atoom weergeeft in het 2D scherm.
- *2D display*  
De manier waarop de producten in 2D op het atoom te zien zijn:
  - 1: *Standard display*  
De producten in de MultiService zijn niet zichtbaar, maar er wordt met tekst aangegeven hoeveel producten er aanwezig zijn.
  - 2: *Move left right*  
De producten zijn zichtbaar en bewegen van links naar rechts.
  - 3: *Move right left*  
De producten zijn zichtbaar en bewegen van rechts naar links.
  - 4: *Level vertical*  
In plaats van de producten is er een gekleurd vlak zichtbaar dat hoger wordt als er meer producten in het MultiService atoom aanwezig zijn.
  - 5: *Level horizontal*  
In plaats van de producten is er een gekleurd vlak zichtbaar dat naar rechts schuift als er meer producten in het MultiService atoom aanwezig zijn.
  - 6: *Line up content*  
De producten zijn zichtbaar en worden onder elkaar weergegeven.

## 10 HET LOCK ATOOM



Figuur 10-1: Het Lock atoom

Wanneer het Lock atoom alleen gebruikt wordt, fungeert het als een toegangspoort waar slechts het aantal producten genoemd in het Lock Level mogen passeren.

Wanneer het Lock atoom tezamen met het Unlock atoom in gebruik is, dient het om het maximaal aantal producten in een bepaald deel van het model te beheersen. De hoeveelheid onderhanden werk (Work in Process) tussen het model begrensd door de Lock en Unlock kan dan maximaal het zgn. Lock Level bedragen.

Vanwege de samenhang worden Unlock atoom hierna besproken.

Op het General tabblad:

- *Atom name*  
De naam van het atoom.
- *Lock level*  
Zodra het aantal atomen tussen Lock en Unlock het hier opgegeven niveau bereikt, sluit de input van de Lock en stopt dus de toevoer.  
Wanneer er geen Unlock toegevoegd is, sluit de input zonder ooit weer te openen
- *Continues evaluation*  
Als deze optie aan staat, wordt het veld Lock Level elke keer uitgelezen wanneer een product het Lock atoom binnenkomt, in plaats van alleen na een reset van het model.  
*Alleen* wanneer het veld Lock Level een 4DScript-commando bevat (en dus niet alleen een getal) kan het nodig zijn deze optie in te schakelen.
- *Batchwise control*  
Wanneer het Lock Level bereikt wordt, wordt de toevoer via het Lockatoom gesloten en pas weer geopend wanneer alle tussenliggende producten via de Unlock

verdwenen zijn. Hierdoor ontstaat een vorm van batch-verwerking met het Lock Level als batchgrootte.

Op het Visualization tabblad:

- *Icon*  
Het figuur dat het Lock atoom weergeeft in het 2D scherm.

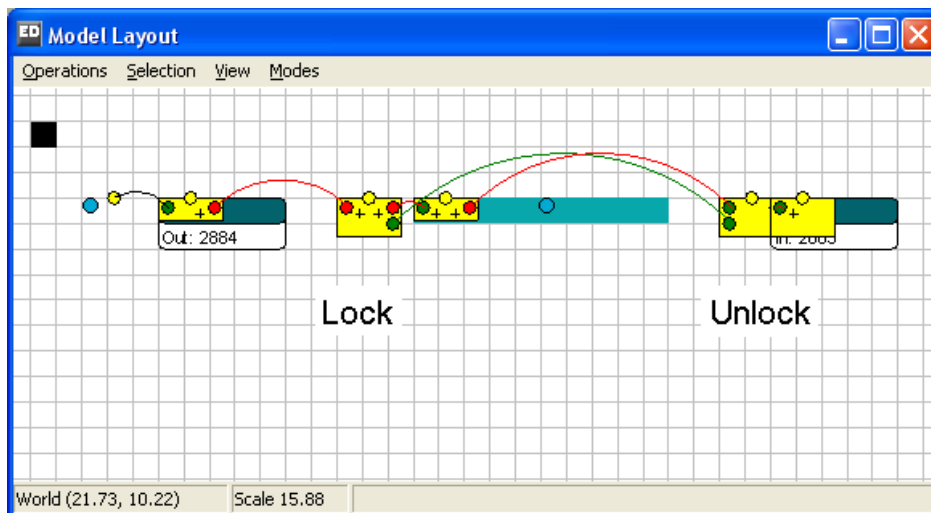


## 11 HET UNLOCK ATOOM

Dit atoom kent geen instellingen en wordt alleen in combinatie met het Lock atoom gebruikt. Elke keer wanneer een product het Unlock atoom verlaat, laat het Lock atoom een nieuw product in het systeem toe (tenzij de optie “Batchwise control” is ingeschakeld bij het Lock atoom, dan moeten eerst alle aanwezige producten het Unlock atoom gepasseerd zijn).

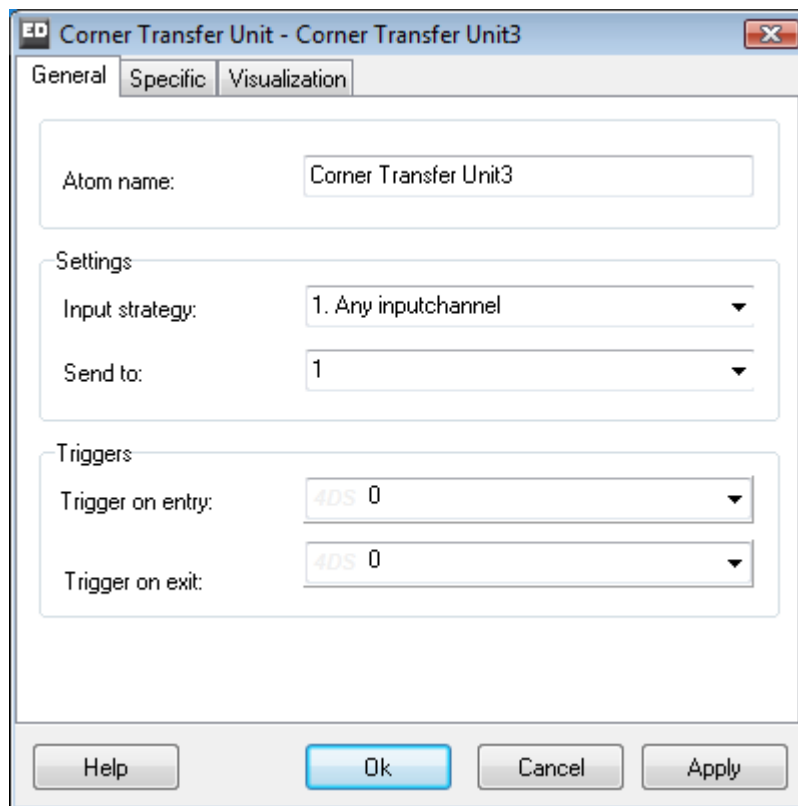
In onderstaand voorbeeld controleren de Lock en Unlock de hoeveelheid producten op de conveyor. Zie hoe beide atomen ‘op normale wijze’ in de productroute geplaatst worden en hoe de Lock en Unlock via hun tweede uitvoer- resp. invoerkanaal verbonden zijn.

De Unlock zelf heeft geen scherm waar iets ingevuld hoeft te worden. Een Lock atoom kan met meerdere Unlock atomen verbonden zijn.



**Figuur 11-1: Weergave van de 2D Model Layout met een Lock en Unlock atoom**

## 12 HET CORNER TRANSFER UNIT ATOOM



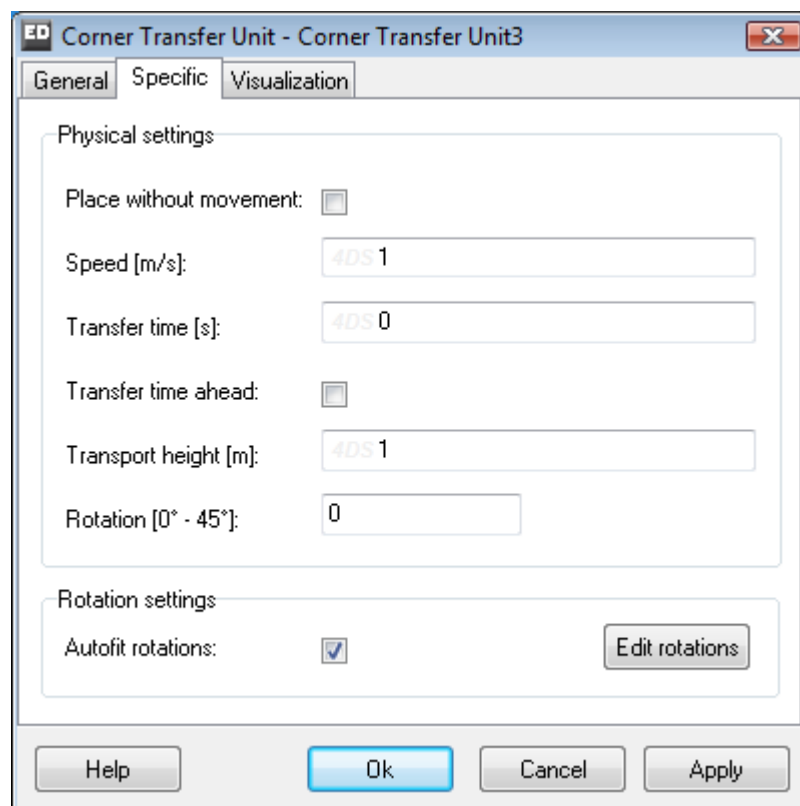
**Figuur 12-1: invoerscherm van het Corner Transfer Unit atoom**

Een Corner Transfer Unit is een intersectie tussen elk type lopendebanden. Anders dan het Turn Table atoom verandert dit atoom niet de oriëntatie van binnenkomende producten in de transport richting van de uitgaande lopende band. Producten houden hun oriëntatie terwijl ze door de Corner Transfer Unit naar links, naar rechts of rechtdoor worden getransporteerd. Als producten om een hoek getransporteerd moeten worden kost het vaak extra transport tijd, hierin wordt de snelheid verlaagd, het aandrijvingsmechanisme verandert en versnelt in de nieuwe transportrichting. Er zijn in totaal vier richtingen waarvan producten binnen kunnen komen of naartoe getransporteerd kunnen worden. Minimaal moet een van die vier richtingen verbonden zijn aan een atoom via welke producten binnen kunnen komen en één richting moet verbonden zijn aan een atoom waarnaar producten getransporteerd kunnen worden. Als er meerdere in of uitgaande richtingen zijn dan is er een strategie nodig waarmee bepaald wordt welke producten eerst getransporteerd worden of die bepaalt in welke richting de producten getransporteerd dienen te worden.

**Opmerking:** dit atoom is onderdeel van de ED Logistics bibliotheek. De bibliotheek van de ED Logistics editie bevat ook een Turn Table atoom, die binnenkomende producten roteert in de nieuwe transport richting voordat deze in die richting doorgestuurd worden.

- *Atom name*  
De naam van het atoom.

- *Input strategy*  
Een regel die bepaalt in welke volgorde de beschikbare producten getransporteerd worden door de Corner Transfer Unit, d.w.z. als er uit meerdere richtingen producten beschikbaar zijn om getransporteerd te worden dan zal deze strategie bepalen uit welke richting het product gehaald wordt om als eerste te transporteren. Men kan een voorgedefinieerde strategie kiezen of zelf een strategie schrijven in een 4DScript expressie.
- *Send to*  
Een regel die bepaalt via welke output kanaal de producten doorgestuurd moeten worden. A rule that determines to which output channel a Product atom needs to be sent. Men kan een voorgedefinieerde strategie kiezen of zelf een strategie schrijven in een 4DScript expressie.
- *Trigger on entry*  
Hier kan een commando worden ingevuld dat wordt uitgevoerd als een product de Corner Transfer Unit binnenkomt.  
Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.
- *Trigger on exit*  
Hier kan een commando worden ingevuld dat wordt uitgevoerd als een product de Corner Transfer Unit verlaat.  
Voor meer uitleg: zie de toelichting bij Trigger on Exit van de Source.



**Figuur 12-2: invoer parameters van de Corner Transfer Unit, Specif tabblad**

- *Place without movement*  
Als deze optie van de Corner Transfer Unit is aangevinkt dan zal er geen beweging te zien zijn als een product de Corner Transfer Unit binnenkomt. Binnenkomende producten zullen direct in het centrum van de Corner Transfer Unit geplaatst worden. Deze optie is vooral geschikt als producten de Corner Transfer Unit binnenkomen via bijvoorbeeld een Portal Crane atoom of een Robot atoom.
- *Speed [m/s]*  
De snelheid waarmee producten de Corner Transfer Unit binnenkomen en waarmee ze verplaatst worden naar het centrum van de Corner Transfer Unit. De snelheid waarmee producten de Corner Transfer Unit verlaten wordt bepaald door het atoom verbonden aan het output kanaal waardoor het product wordt doorgezonden.
- *Transfer time [s]*  
De tijd die nodig is om de snelheid af te bouwen, het aandrijvingsmechanisme te veranderen en de snelheid in de nieuwe transportrichting op te bouwen als een product een hoek om wordt getransporteerd.
- *Transfer time ahead*  
Als deze optie is aangevinkt dan wordt de *Transfer time* ook als nodig geacht voor producten die rechtdoor getransporteerd worden.
- *Transport height [m]*  
De lengte van de poten van de transportband (in meters).
- *Rotation [+/- 45°]*  
Standaard worden transportbanden horizontal in uw scherm geplaatst, waarbij de producten van links naar rechts getransporteerd worden. Het is echter mogelijk de transportbanden te roteren om zo een weergave te krijgen die overeenkomt met de werkelijkheid.

**Opmerking:** Door de weergave van het Corner Transfer Unit atoom is de mogelijkheid voor rotatie beperkt tot +/- 45°. Toch kan hiermee elk transport over een hoek van 90°, -90° en 180° gemodelleerd worden.

- *Autofit rotations*  
Als deze optie is aangevinkt zal de Corner Transfer Unit Atom op reset trachten de rotatie te achterhalen van de atomen die verbonden zijn aan haar input en output kanalen. Deze informatie heeft de Corner Transfer Unit nodig om de rotatie van de producten die door de Corner Transfer Unit getransporteerd worden te berekenen.

**Opmerking:** De Corner Transfer Unit zal met de Autofit optie niet in alle gevallen in staat zijn de rotatie te herkennen van de atomen die verbonden zijn aan haar input en output kanalen. In dat geval kan de optie uitgezet worden en kan handmatig de rotatie tabel aangepast worden.



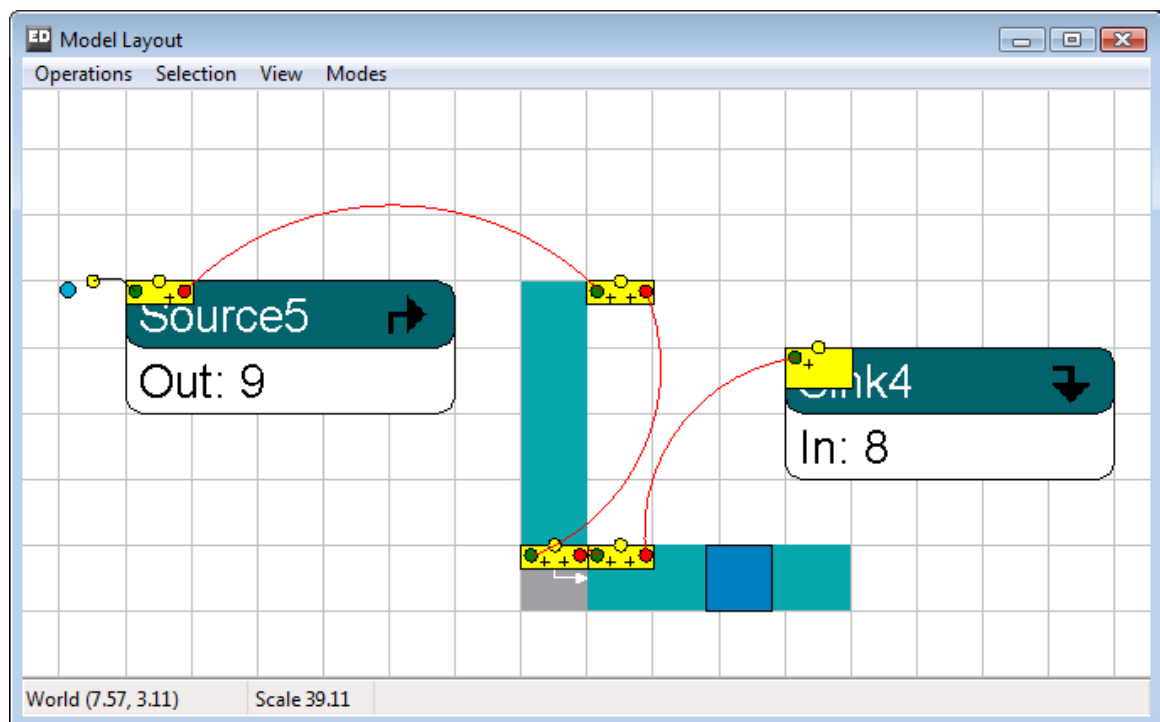
- *Edit rotations*

Klik op deze knop om de rotatie tabel aan te passen. Pas de “motion flags” aan voor de input en output motion directions. Type een 1 voor positieve x/y richtingen en -1 voor negatieve x/y richtingen.

**Opmerking:** Het aantal rijen in de rotatie tabel wordt op Reset aangepast aan het aantal input en output kanalen. Kanalen die niet verbonden zijn aan een atoom zullen een foutmelding geven dit ter voorkoming van het ontstaan van onverwacht gedrag tijdens de simulatie.

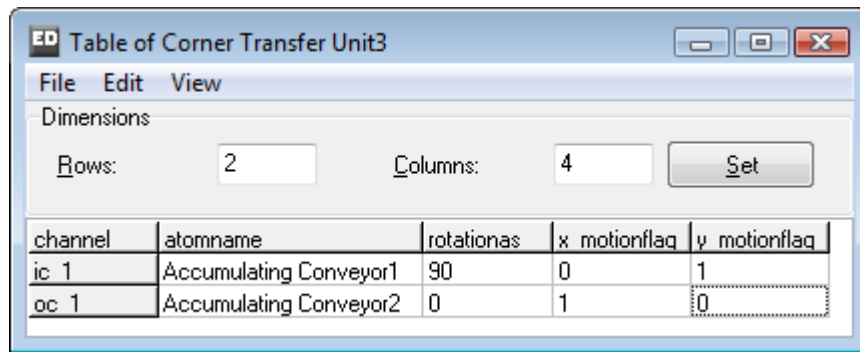
**Opmerking:** Zet voor elke rij 1 “motion flag”, voor een beweging langs de x-as of voor een beweging langs de y-as.

Voorbeeld: Verbind een Sourcemet een 90° geroteerde Accumulating Conveyor (rollenband). De conveyor transporteert de producten naar een Corner Transfer Unit atoom. De producten worden vervolgens door de Corner Transfer Unit getransporteerd naar een volgende Accumulating Conveyor met een rotatie van 0°, daarna verlaten de producten het model via een Sink.



**Figuur 12-3: 2D Model Layout voorbeeld Corner Transfer Unit atoom**

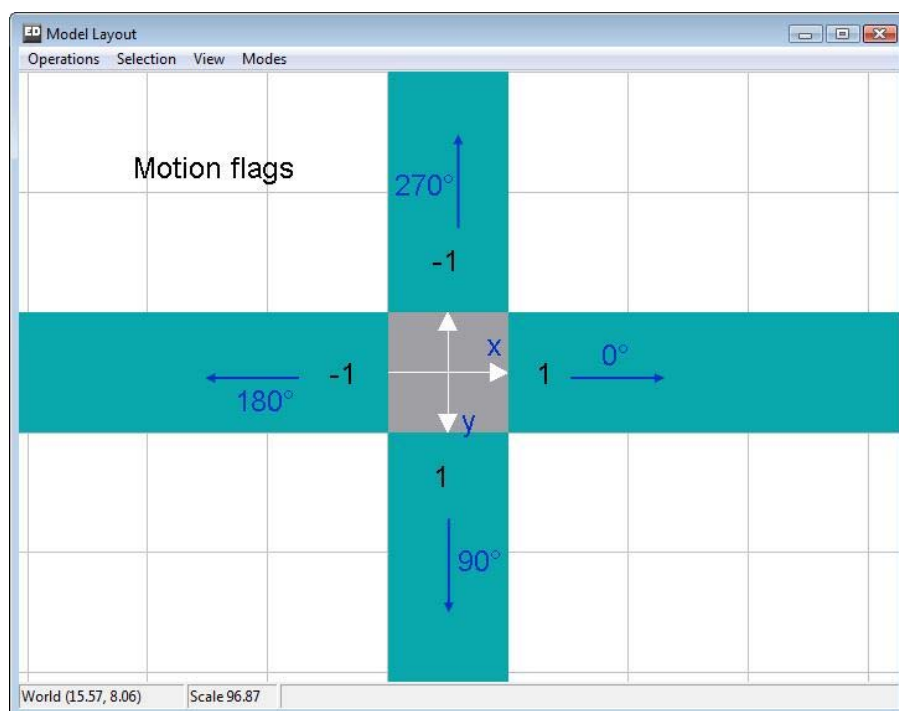
Zet de x-motion flag op 0 en de y-motion flag op 1 voor producten die binnenkomen via een input kanaal die verbonden is aan een atoom met rotatie 90° rond zichzelf. Zet x-motion flag op 1 en y-motion flag op 0° voor producten die door een output kanaal gaan die verbonden is aan een atoom met rotatie 0° rond zichzelf.



channel	atomname	rotationas	x motionflag	y motionflag
ic 1	Accumulating Conveyor1	90	0	1
oc 1	Accumulating Conveyor2	0	1	0

**Figuur 12-4: De rotatie table**

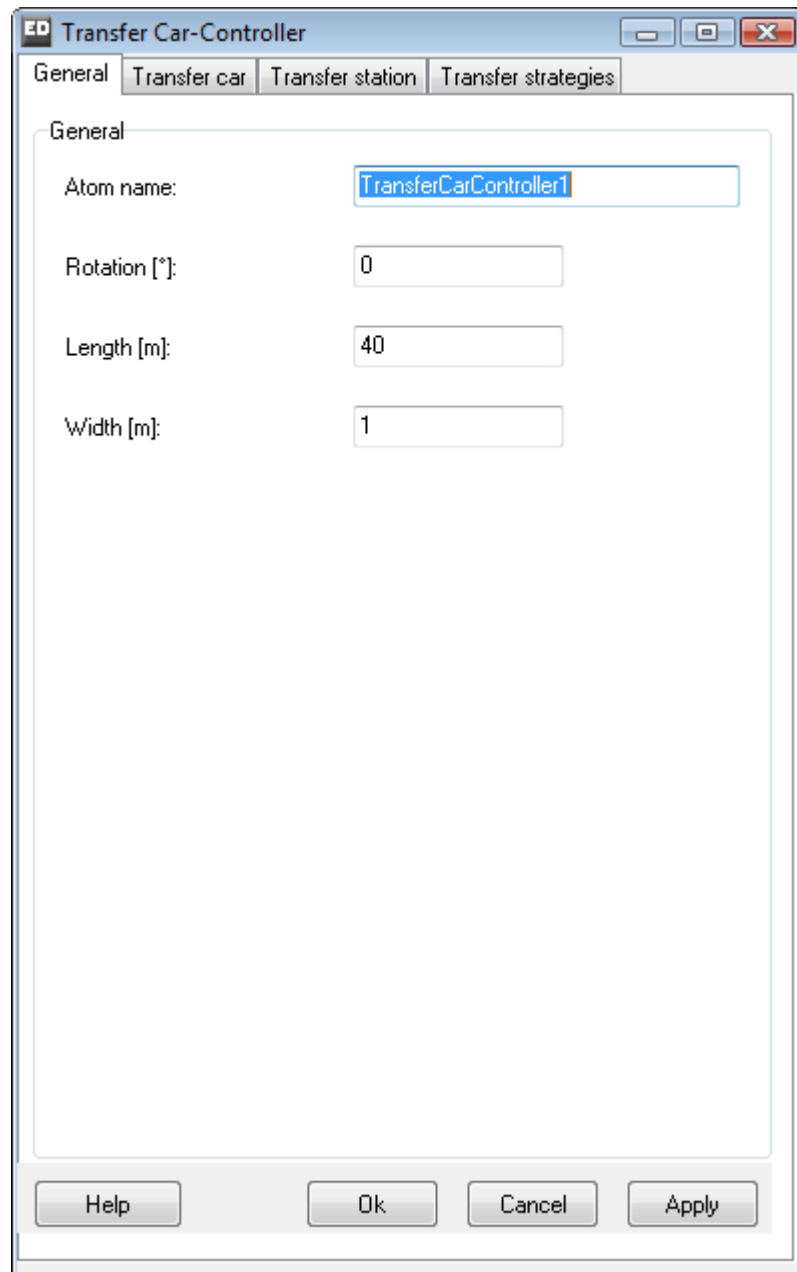
Als de 'Autofit rotations' optie van het Corner Transfer Unit atoom is aangevinkt, dan wordt de rotatie tabel automatisch gevuld. Het resultaat van de aanpassingen zijn zichtbaar in de 2D Model Layout. Witte pijlen tonen de berekende transport richtingen.



**Figuur 12-5: Motion flags**

## 13 HET TRANSFER CAR ATOOM

---



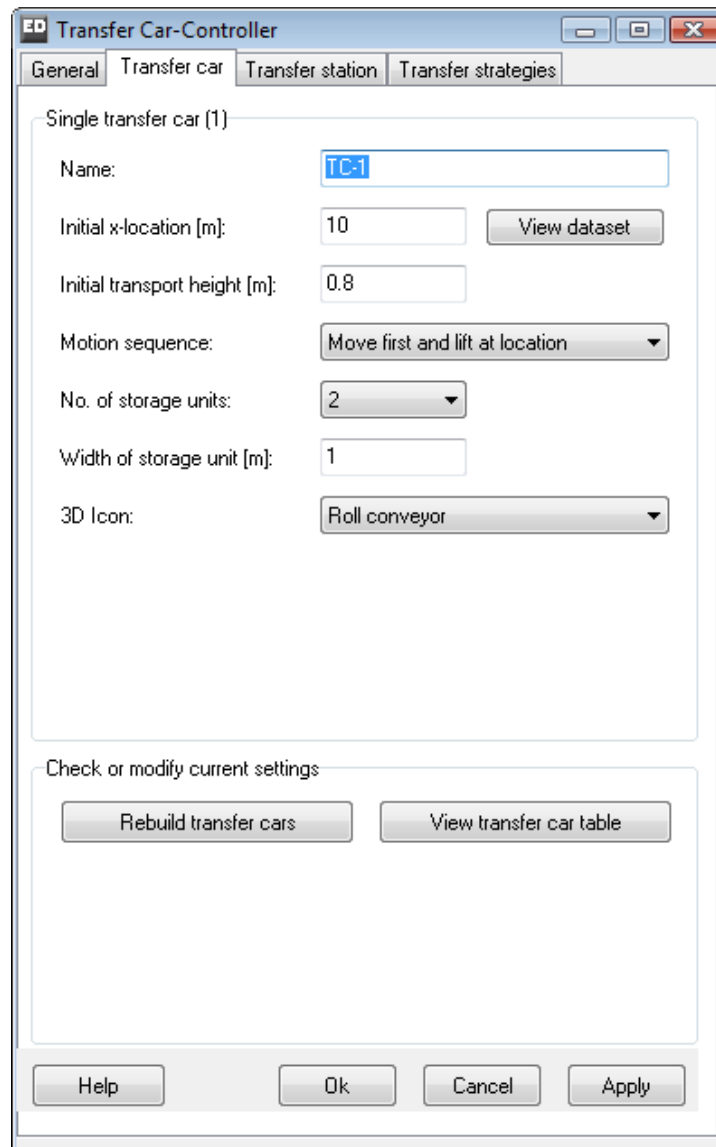
**Figuur 13-1: invoerscherm van het Transfer Car atoom**

Een Transfer Car system is een reëel gebonden enkel spoor of dubbelspoor voor snel en min of meer lineair transport en/of voor het combineren van meerdere input en output locaties van materiaal afhandeling. In het bijzonder voor lange afstanden geven Transfer Cars de voorkeur boven conveyors, dit om bijvoorbeeld de inbound van verschillende magazijn gangen met elkaar te verbinden. Het Transfer Car system bestaat uit een veel gebonden wagen, de verbindings Stations en een Controller om het gedrag van het systeem te coördineren. De wagen zelf kan één of meerdere producten tegelijk verplaatsen, for example to swap Products at a particular station. Products have to be

conveyed from the input station to the storage units and after the linear transport movement they have to be conveyed to the output station as well. Sometimes there are altitude differences between input and output station. In this case the Car has to compensate the vertical distance. Each transportation direction has its own behavior regarding acceleration, maximum speed and deceleration. The Stations are located on the left and right side along the rails. A Station can consist of one or more ports. Each port can be separately served by the Car. There are 3 types of ports. Pick-ports are intended to absorb incoming Products. Place-ports are designated to deliver Products to the car. Pick-and-Place-ports can do both. The Controller coordinates interaction between loading, transportation and unloading taking into account all predefined strategy.

**Important:** The Transfer Car system is made of several sub atoms not intended to be dragged into the model. Drag the TransferCarController atom into your model, to open the GUI with all the edit fields to build or rebuild a complete Transfer Car system.

- *Atom name*  
The name given to the atom.
- *Rotation [°]*  
At default the Transfer Car system is positioned from left to right in your screen. However you can also rotate the atom to display the flow of atoms over the Transfer Car system matches reality.
- *Length [m]*  
The length of the rail systems (x-axis) to connect all the Stations within the range of the Transfer Car.
- *Width [m]*  
The width of the rail system also defines the y-dimension of the Transfer Car.



**Picture 13-2: The Transfer Car parameters**

- *Name*  
The name of the Transfer Car.
- *Initial x-location [m]*  
The initial position of the car along the rails on start of simulation.
- *Initial transport height [m]*  
The initial vertical transport height of the car on start of simulation.
- *Motion sequence*  
A rule that determines how the Transfer Car atom coordinates the movements of lifting and driving. There are 2 strategies available: 2D:  
  - 1: *Move left right*  
The products are visible and move from the left to the right.

2: *Move first and lift at location*

Move and lift together.

- *No. of storage units*

A Transfer Car can have up to three Storage units in order to move several Products at the same time.

**Note:** The strategies to control a Transfer Car system are more complex if the Transfer Car has more than one Storage unit, as you have to control the interaction between the several storage units.

- *Width of storage unit [m]*

The x-dimension of every Storage unit. Defines together with the width [m] of the rail system and the No. of storage units the x-dimension of the Transfer Car.

- *3D Icon*

A 3D icon that can be used to visualize the Transfer Car atom in 3D. Change this icon to switch between simple box, roll conveyor, belt conveyor or chain conveyor.

- *Rebuild transfer cars*

Click this button to rebuild the Transfer Car and all storage units with the parameters of the edit fields.

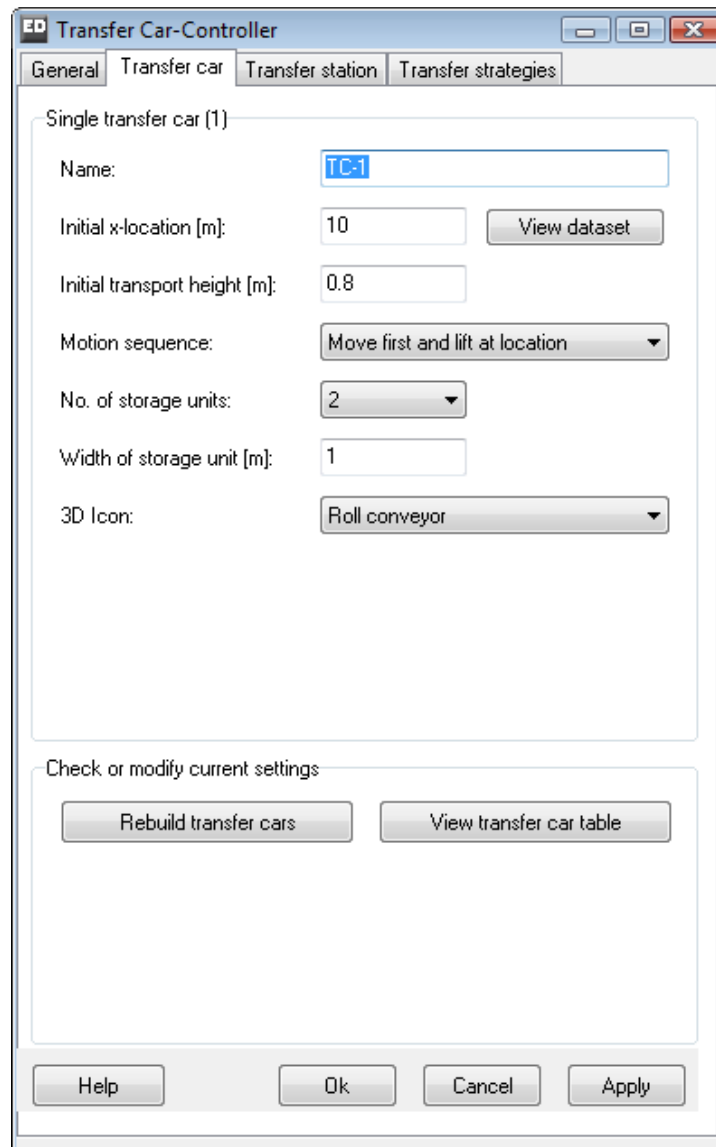
- *View transfer car table*

Click on this button to view the table with the detailed parameters of all Transfer Cars.

**Note:** Because of the high number of Transfer Car parameters some of them are defined on a separate Transfer Car GUI.

- *View dataset*

Click on this button to edit detailed parameters of the Transfer Car on a separate Transfer Car GUI.



**Picture 13-2: The Transfer Car parameters**

- *Name*  
The name of the Transfer Car.
- *Initial x-location [m]*  
The initial position of the car along the rails on start of simulation.
- *Initial transport height [m]*  
The initial vertical transport height of the car on start of simulation.
- *Motion sequence*  
A rule that determines how the Transfer Car atom coordinates the movements of lifting and driving. There are 2 strategies available: 2D:  
  - 1: *Move left right*  
The products are visible and move from the left to the right.

2: *Move first and lift at location*

Move and lift together.

- *No. of storage units*

A Transfer Car can have up to three Storage units in order to move several Products at the same time.

**Note:** The strategies to control a Transfer Car system are more complex if the Transfer Car has more than one Storage unit, as you have to control the interaction between the several storage units.

- *Width of storage unit [m]*

The x-dimension of every Storage unit. Defines together with the width [m] of the rail system and the No. of storage units the x-dimension of the Transfer Car.

- *3D Icon*

A 3D icon that can be used to visualize the Transfer Car atom in 3D. Change this icon to switch between simple box, roll conveyor, belt conveyor or chain conveyor.

- *Rebuild transfer cars*

Click this button to rebuild the Transfer Car and all storage units with the parameters of the edit fields.

- *View transfer car table*

Click on this button to view the table with the detailed parameters of all Transfer Cars.

**Note:** Because of the high number of Transfer Car parameters some of them are defined on a separate Transfer Car GUI.

- *View dataset*

Click on this button to edit detailed parameters of the Transfer Car on a separate Transfer Car GUI.



The screenshot shows the 'Transfer Car' dialog box with the 'Speed' tab selected. The dialog is organized into four main sections, each with three dropdown menus for Speed, Acceleration, and Deceleration. All dropdowns are currently set to '4DS 1'. The sections are: Empty drive, Loaded drive, Convey speed, and Vertical drive. The bottom of the dialog features 'Ok', 'Cancel', and 'Apply' buttons.

**Picture 13-3: The detailed Transfer Car dataset**

The following additional parameters can be changed on page General of the detailed Transfer Car GUI:

- *Description*  
A describing text.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Transfer Car atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.
- *Trigger on exit*  
A rule that determines what kind of action needs to be executed when a Product

atom exits the Transfer Car atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

The following additional parameters can be changed on page Transfer Car of the detailed Transfer Car GUI:

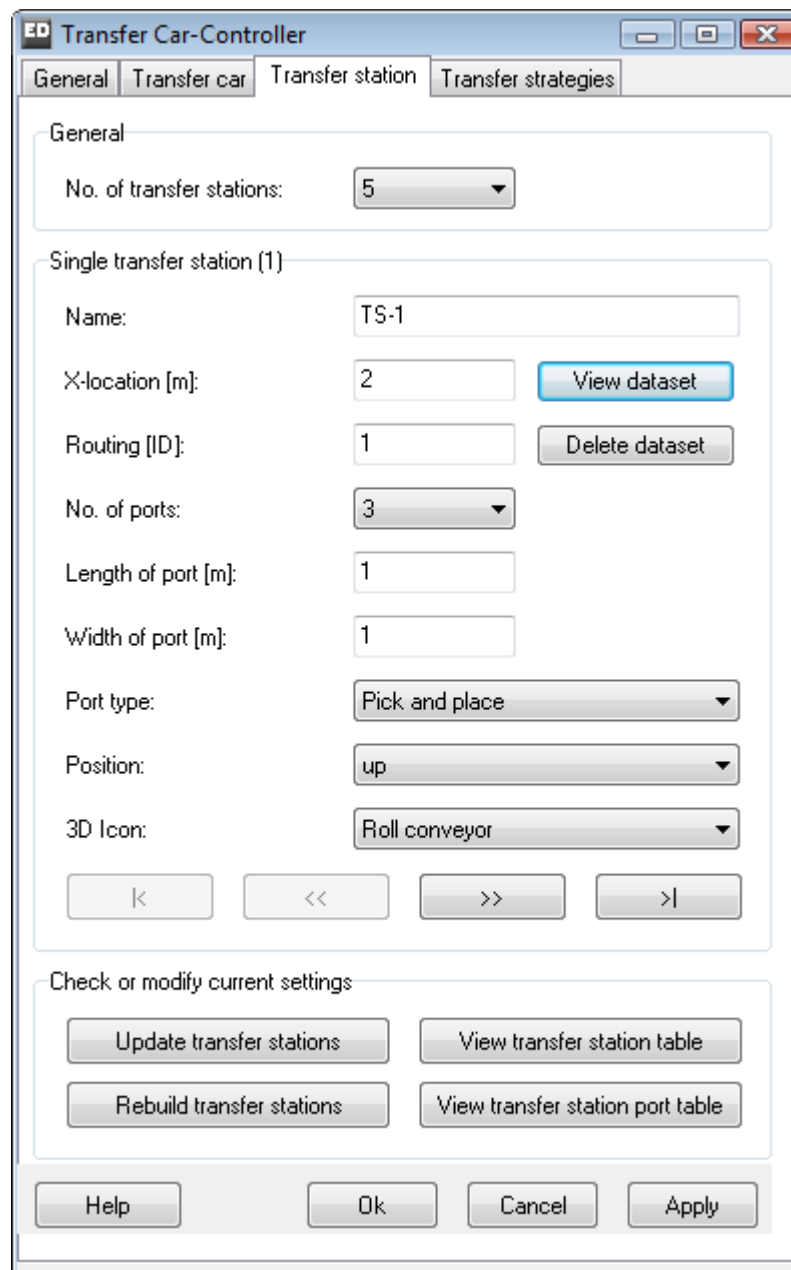
- *ID*  
The index of the Car.

The following additional parameters can be changed on page Speed of the detailed Transfer Car GUI:

- *Empty Drive*  
Speed [m/s], Acceleration [m/s<sup>2</sup>] and Deceleration [m/s<sup>2</sup>] of empty Transfer Car along rails direction.
- *Loaded Drive*  
Speed [m/s], Acceleration [m/s<sup>2</sup>] and Deceleration [m/s<sup>2</sup>] of loaded Transfer Car along rails direction.
- *Convey Drive*  
Speed [m/s], Acceleration [m/s<sup>2</sup>] and Deceleration [m/s<sup>2</sup>] of Transfer Car when loading or unloading goods from or to Stations.
- *Vertical Drive*  
Speed [m/s], Acceleration [m/s<sup>2</sup>] and Deceleration [m/s<sup>2</sup>] of Transfer Car when balancing altitude differences between input and output station.

The following additional parameters can be changed on page Visualization of the detailed Transfer Car GUI:

- *Color*  
The color of the Transfer Car.



**Picture 13-4: The Transfer Station parameters**

- *No. of transfer stations*  
The Transfer Car system can have up to 20 Stations.

**Note:** Use the arrow buttons to switch between the parameters of every individual Station.

- *Name*  
The name of the Transfer Station.
- *X-location [m]*  
The position of the Station along the rail track.

- *Routing [ID]*  
The index of the Station that is used to define the sequence of transportation inside the Routing plan on page Transfer strategies.
- *No. of ports*  
Each Station can have up to 6 Ports with an individual Port ID from 1 to 6.
- *Length of port [m]*  
The size on the x-axis.
- *Width of port [m]*  
The size on the y-axis.
- *Port type*  
There are 3 types of ports. Pick-Ports are intended to absorb incoming Products. Place-Ports are designated to deliver Products to the car. Pick-and-Place-Ports can do both.
- *Position*  
Select up to place a Station on the left side of the rail track. Select down to place a Station on the right side of the rail track.
- *3D Icon*  
A 3D icon that can be used to visualize the Station atom in 3D. Change this icon to switch between simple box, roll conveyor, belt conveyor or chain conveyor.
- *Delete dataset*  
Click on this button to remove a whole Station with all Ports from the Transfer Car system.
- *Rebuild transfer stations*  
Click this button to update parameters of the Transfer Stations with all Ports according to the parameters of the edit fields.
- *Rebuild transfer stations*  
Click this button to rebuild the Transfer Stations with all Ports according to the parameters of the edit fields.
- *View transfer station table*  
Click on this button to view the table with the detailed parameters of all Transfer Stations.
- *View transfer station port table*  
Click on this button to view the table with the detailed parameters of all Transfer Stations and the assigned Ports.

**Note:** Because of the high number of Station parameters some of them are defined on a separate Transfer Station GUI.

- *View dataset*  
Click on this button to edit detailed parameters of the Station on a separate Transfer Station GUI.

**Picture 13-5: The detailed Transfer Station dataset**

The following additional parameters can be changed on page General of the detailed Transfer Station GUI:

- *Description*  
A describing text.
- *Height of port [m]*  
The size on the z-axis.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Station atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

- *Trigger on exit*  
A rule that determines what kind of action needs to be executed when a Product atom exits the Station atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.
- *Send to*  
A rule that determines to which output channel a Product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.

The following additional parameters can be changed on page Transfer station of the detailed Transfer Station GUI:

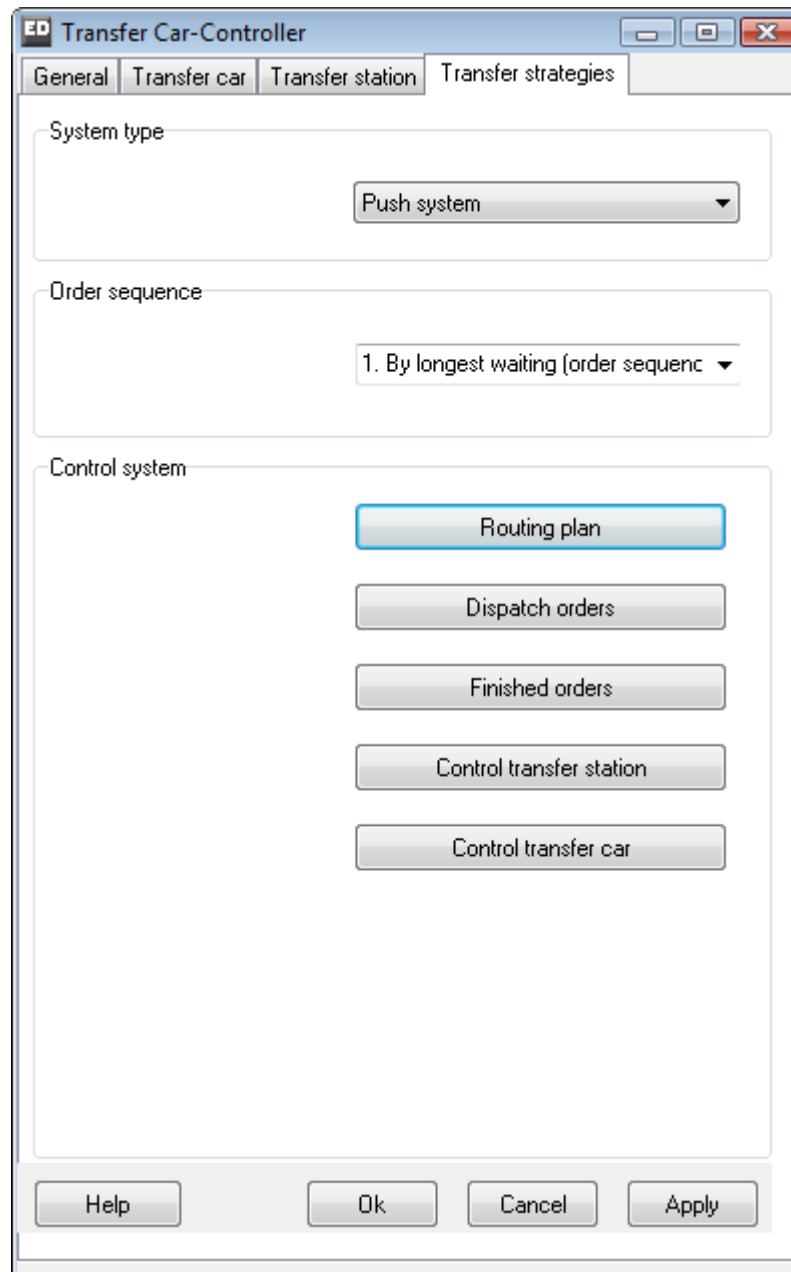
- *Station [ID]*  
An information about the index of the Station (read only).
- *Port [ID]*  
An information about the index of the Port (read only).
- *Direction at entry*  
Determines the offset location and the direction of the input movement for incoming Products. There are 7 strategies available:
  - 1: *Product enters from the top.*
  - 2: *Product enters from the bottom.*
  - 3: *Product enters from the left.*
  - 4: *Product enters from the right.*
  - 5: *Make entry direction dependent on income channel:*
  - 6: *Choose between several entry channels:*
  - 7: *By user: Enter your own 4DScript expression:*
- *Direction at exit*  
Determines the direction of the output movement for outgoing Products. There are 7 strategies available:
  - 1: *Product exits at the top.*
  - 2: *Product exits at the bottom.*
  - 3: *Product exits at the left.*
  - 4: *Product exits at the right.*
  - 5: *Make exit direction dependent on income channel:*
  - 6: *Choose between several exit channels:*
  - 7: *By user: Enter your own 4DScript expression:*

The following additional parameters can be changed on page Speed of the detailed Transfer Station GUI:

- *Convey*  
Speed [m/s], Acceleration [m/s<sup>2</sup>] and Deceleration [m/s<sup>2</sup>] of Transfer Stations when loading or unloading Products from or to external connections.

The following additional parameters can be changed on page Visualization of the detailed Transfer Station GUI:

- *Color*  
The color of the Transfer Station.

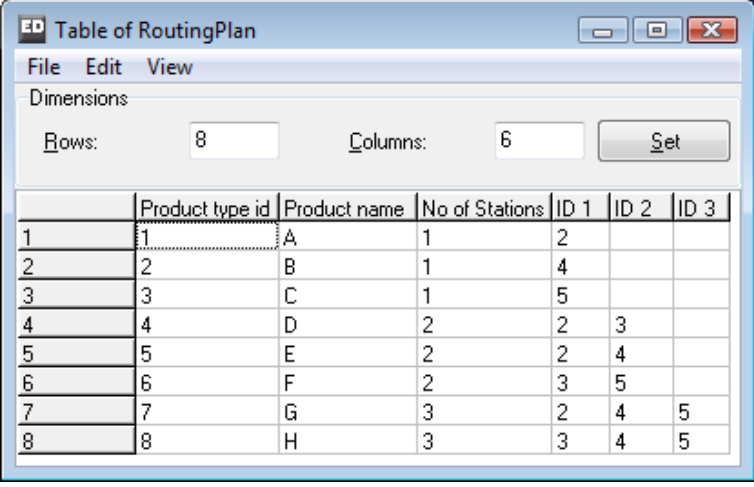


**Picture 13-6: The Transfer strategies**

- *System type*  
The Transfer Car system can be controlled as Push system or Pull system.

If you select Push system, the Controller evaluates incoming Product with a Label 'ProductTypeID' to identify the Station (row) inside the Routing plan

table. Be sure that the Routing plan has all information about the sequence of Stations for each product type.



	Product type id	Product name	No of Stations	ID 1	ID 2	ID 3
1	1	A	1	2		
2	2	B	1	4		
3	3	C	1	5		
4	4	D	2	2	3	
5	5	E	2	2	4	
6	6	F	2	3	5	
7	7	G	3	2	4	5
8	8	H	3	3	4	5

**Picture 13-7: The Routing Plan (push mode)**

If you select Pull system, the Controller is awaiting calls from the Place-Stations, before he is searching for Products ready for transportation. For more information see the 'Transfer Car Pull' Model in the 'Examples' directory of your Enterprise Dynamics program directory.

- *Order sequence*  
Define how the sequence of orders (transportation tasks) is executed by the Transfer Car. Choose a predefined strategy or define your own strategy.
- *Routing plan*  
Click this button to edit the product-specific sequence of Stations.
- *Dispatch orders*  
Click this button to view all open orders.
- *Finished orders*  
Click this button to view all finished orders.
- *Control transfer station*  
Click this button to view the table of the Transfer Station Controller.
- *Control transfer car*  
Click this button to view the table of the Transfer Car Controller.



## 14 THE RFID-GATE ATOM

---

The screenshot shows the 'RFID-Gate' configuration window with the 'General' tab selected. The window has a title bar with the 'ED' logo and standard window controls. Below the title bar are four tabs: 'General', 'Settings', 'Network', and 'Visualization'. The 'General' tab contains the following fields:

- Atom name:** A text box containing 'RFID-Gate9'.
- Physical settings:** A group box containing five text boxes:
  - Length [m]:** 1
  - Width [m]:** 3
  - Height [m]:** 2
  - Floor level [m]:** 0
  - Rotation [°]:** 0
- Triggers:** A group box containing two dropdown menus:
  - Trigger on entry:** Set to '4DS 0'.
  - Trigger on exit:** Set to '4DS 0'.

At the bottom of the window are four buttons: 'Help', 'Ok', 'Cancel', and 'Apply'.

**Picture 14-1: The RFID-Gate atom**

This atom can be connected to a network (Network Nodes) and work as a station, where Tags can be manipulated. Tags are treated as Labels. The RFID-Gate atom consists of 3 different working modes; it can read, write and read and write Labels. All 3 modes can be used simultaneously. The user can specify several Labels and operation in the GUI table. When an Advanced Transporter atom passes the RFID-Gate, the values of the Labels are written to the result table. The table can be found in the GUI.

**Important:** Before an RFID-Gate can be connected to a Network, you have to chose whether the path through the Gate is bidirectional or unidirectional.

- *Atom Name*  
The name of the atom.
- *Length [m]*  
The length of the atom in meters.
- *Width [m]*  
The width of the atom in meters.
- *Height [m]*  
The height of the atom in meters.
- *Floor level [m]*  
Helpful if the level of transport is not equal to the floor level.
- *Rotation [°]*  
The atom can be rotated into transport direction to match the flow of the connected Network Nodes.
- *Trigger on entry*  
Determines what kind of action needs to be executed when a Advanced Transporter atom enters the RFID-Gate. There are a number of rules predefined (see also Trigger on exit), but you can also create your own rule via a 4DScript expression.
- *Trigger on exit*  
Determines what kind of action needs to be executed when a Advanced Transporter atom exits the RFID-Gate. There are a number of rules predefined (see also Trigger on entry), but you can also create your own rule via a 4DScript expression.

ED RFID-Gate

General Settings Network Visualization

Table

Number of rows: 3

Labels

Name: ID

Reference: Product

Mode: Read

Value: 4DS 0

<| < > >|

	Tag name	Reference	Mode	Value
1	ID	Product	r-	0
2	Group	Product	r-	0
3	Timestamp	Transporter	rw	round(Time)

Show results Update

Help Ok Cancel Apply

**Picture 14-2: The RFID-Gate label settings**

- *Number of rows*  
Define the number of operations that needs to be performed in the RFID-Gate. Each operation has an individual row to define the Label of the specific operation.
- *Name*  
The name of the Label (Tag) that you want to define.
- *Reference*  
The operation can refer to the Advanced Transporter or to the transported Product. If the Advanced Transporter carries more than one Product, the desired

operation is performed on all Products inside the Transporter. If the reference is the Transporter, the Label will be read from or written to the Transporter. If the reference is the Product, the Label will be read from or written to all Products carried by the Transporter.

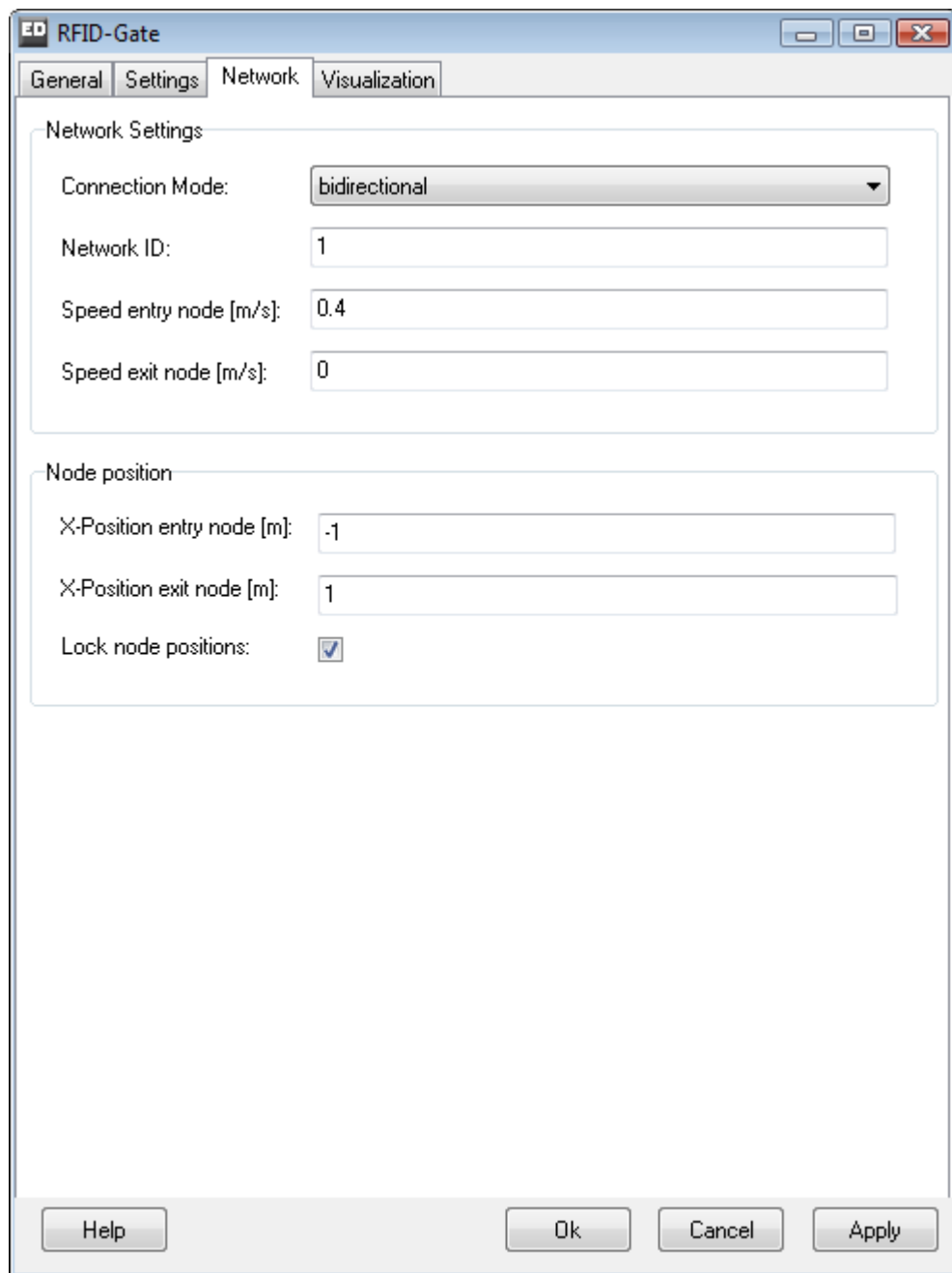
- *Mode*  
The RFID-Gate can work in different modes. There are 3 modes available:  
1: *Read mode only*,  
2: *Write mode only*,  
3: *Read and write mode*  
Means that the Labels are read out and then rewritten.
- *Value*  
If the Label has to be written, you can specify the desired value of the Label here. Expressions are allowed.

**Important:** The content of the above field should appear immediately in the GUI table. If this is not the case, choose the 'Reference' or the 'Mode' again.

- *'Forward' and 'Backward' Buttons*  
Click this buttons to switch between the table rows. The marker will go up or down according to the selected action.
- *'Show Results' Button*  
If a Label is read out, its content is written to the result table. Click this button to open the table.

*'Update' Button*

When the user adds, deletes or edits an operation, this button has to be clicked to apply the changes to the results table. This button cleans the results table from old result values (previous runs) as well.



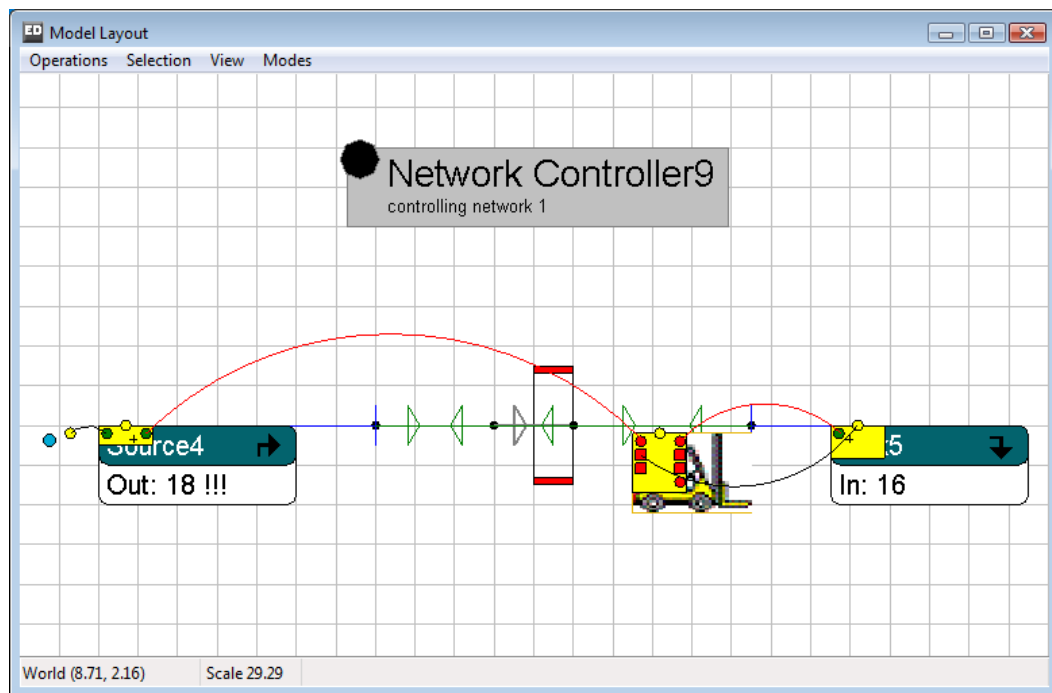
**Picture 14-3: The RFID-Gate network settings**

- *Connection Mode*  
Choose whether the RFID-Gate can be passed in two directions (bidirectional) or one way only (unidirectional).

**Note:** The RFID-Gate will perform its operations if the Advanced Transporter enters through the entry node. No operation will be performed if the Advanced Transporter enters the RFID-Gate from the exit node.

- *Network ID*  
This is the ID of the Network where the RFID-Gate is connected to.

- *Speed entry node [m/s]*  
This is the speed limit for the Advanced Transporter if it enters through the entry node.
- *Speed exit node [m/s]*  
This is the speed limit for the Advanced Transporter if it enters through the exit node.
- *X-Position entry node [m/s]*  
This is the position of the entry node on the x-axis. This is measured relative to the RFID-Gate.
- *X-Position exit node [m/s]*  
This is the position of the exit node on the x-axis. This is measured relative to the RFID-Gate.



**Picture 14-4: The RFID-Gate 2D Model Layout**

## 15 THE ADVANCED VERTICAL ARTICULATED ROBOT ATOM

---

Advanced Vertical Articulated Robot - Advanced Vertical Art...

General | Size | Load | Speed | Routines | Tasks | Visualization

Atom name: Advanced Vertical Articulated Robot1

Coordinate system

Offset (x) of coordinate system [m]: 0

Offset (y) of coordinate system [m]: 0

Offset (z) of coordinate system [m]: 0

Tool default position

Start position (x) of tool [m]: 1

Start position (y) of tool [m]: 1

Start position (z) of tool [m]: 1

Triggers

Trigger on entry: 4DS 0

Trigger on exit: 4DS 0

Trigger on reset: 4DS 0

Help Ok Cancel Apply

**Picture 15-1: The Advanced Vertical Articulated Robot atom**

An Vertical Articulated Robot is a handling tool for any kind of pick-, place- and motion-tasks. The sphere of action is limited by the length of robot arms. A layout of at least 5 axis are necessary to reach any destination within the sphere of action. Motions are defined based on a world coordinate system. The destinations of the Tool are defined with xyz-coordinates. Any kind of action like loading, unloading, moving, delays are defined within Routines. Those Routines are stored with global references, so they can be used by other Vertical Articulated Robots as well. Every kind of command has its specific parameters like load time or speed to define the behavior of the robot. There are two ways to define routine motion commands. First one is to move the tool of the robot by hand and then get the coordinates in order to teach the command into the routine. Second one is the other way around when the robot is assigned to drive to user edited

coordinates. If the destination has been reached successfully, the coordinates can be integrated into an motion command. Routines can be assigned to tasks. Routine execution is triggered if there is a task available. Strategies can be defined to decide which routine has to be executed, for what kind of task. If a Routine has been started it executes all commands in sequence before it becomes idle again. Due to high level of danger Advanced Vertical Articulated Robot systems are driven without human interference.

Important: There are two ways to use the Advanced Vertical Articulated Robot. For any kind of pick and place tasks, the robot channels have to be connected to previous and following atoms to make it part of the material flow. Motion without material handling do not need channel connections, but the Routines need to be started by external call.

- *Atom Name*  
The name of the atom.
- *Offset (x/y/z) of coordinate system [m]*  
All motions of Advanced Vertical Articulated Robot are determined based on the origin of its coordinate system. Per default it is located in the rotation center of the robot socket at z-level zero. Edit offsets to shift that point to a more comfortable location, before starting routine definition.

**Note:** It can be helpful to lock the position of the robot to protect it from unintended dislocation.

- *Start position (x/y/z) of tool [m]*  
Default position of the tool center relating to the coordinate system. The position is set on reset.

**Note:** Choose a start position that can be reached according to robot geometry and sizes.

- *Trigger on entry*  
Determines what kind of action needs to be executed when a Product atom enters the Advanced Vertical Articulated Robot atom. There are a number of rules predefined (see also Trigger on exit), but you can also create your own rule via a 4DScript expression.

**Note:** The trigger on entry is executed from the Tool sub-atom, not from the Advanced Vertical Articulated Robot atom itself.

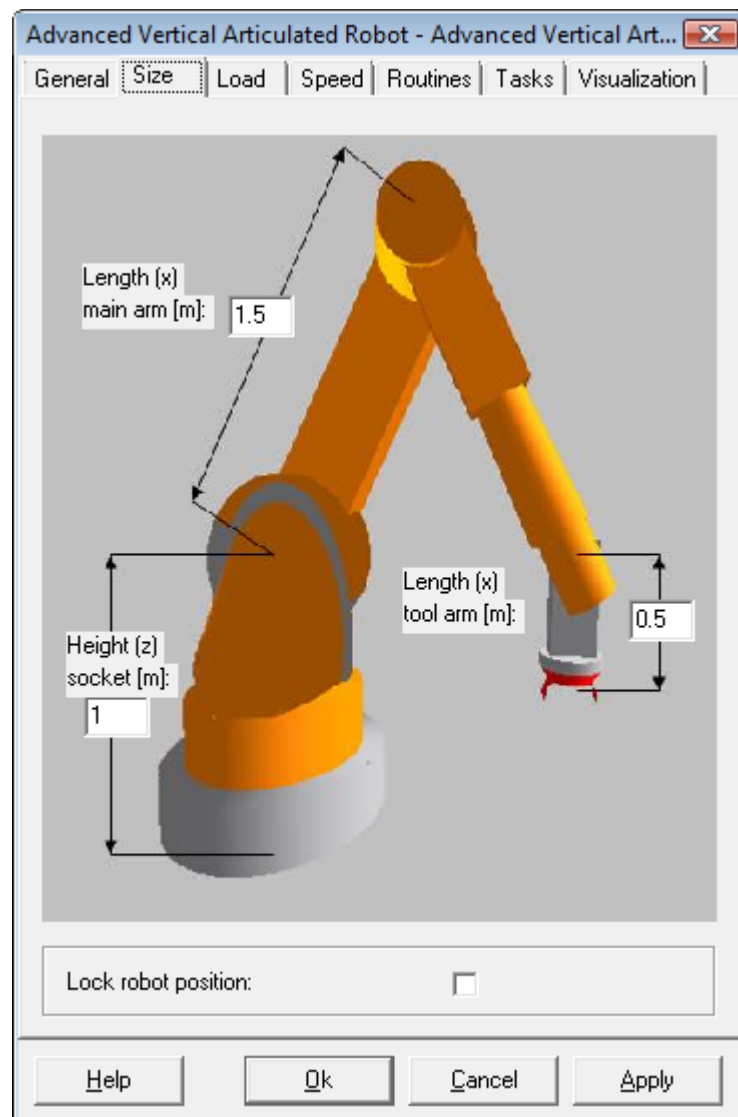
- *Trigger on exit*  
Determines what kind of action needs to be executed when a Product atom exits the Advanced Vertical Articulated Robot atom. There are a number of rules predefined (see also Trigger on entry), but you can also create your own rule via a 4DScript expression.

**Note:** The trigger on exit is executed from the Tool sub-atom, not from the Advanced Vertical Articulated Robot atom itself.



- *Trigger on reset*  
Determines what kind of action needs to be executed when the Advanced Vertical Articulated Robot atom is reset. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

**Note:** The trigger on reset is executed from the Tool sub-atom, not from the Advanced Vertical Articulated Robot atom itself.

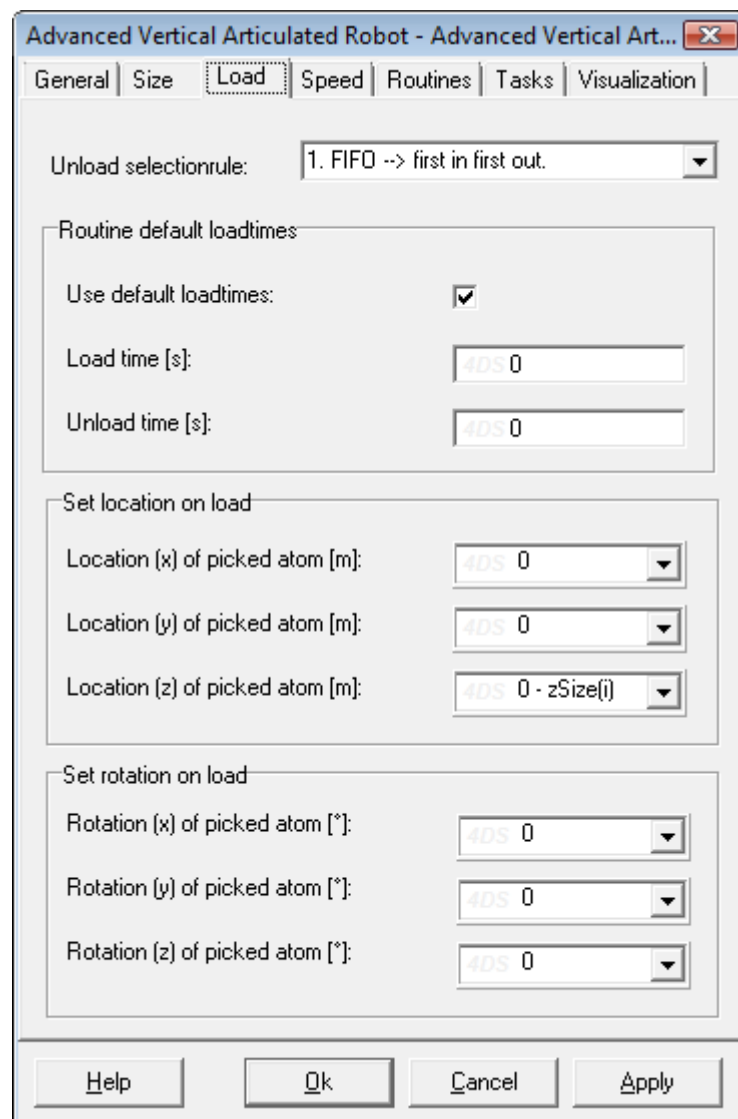


**Picture 15-2: The Advanced Vertical Articulated Robot size parameters**

The layout of robots is defined by number/kind of articulated joints and the length of arms. The Advanced Vertical Articulated Robot atom has 5 axis. Starting with a horizontally rotating socket there are two vertical main arms of equal length, followed by another vertical Tool arm that finally can be rotated around its vertical centre.

- *Height (z) of socket [m]*  
The z-location of the first joint of the main arm.

- *Length (x) of main arm [m]*  
Define the main sphere of action.
- *Length (x) of tool arm [m]*  
Determines the sphere of action in detail.
- *Lock robot position*  
If this box is checked the position of the robot is locked to protect it from unintended dislocation.



**Picture 15-3: The Advanced Vertical Articulated Robot load parameters**

- *Unload selectionrule*  
This option lets unload the picked atoms in a specified sequence. The internal queue of the Advanced Vertical Articulated Robot is sorted in the selected

sequence before unloading. Default is first in first out (FIFO). There are 6 strategies available.

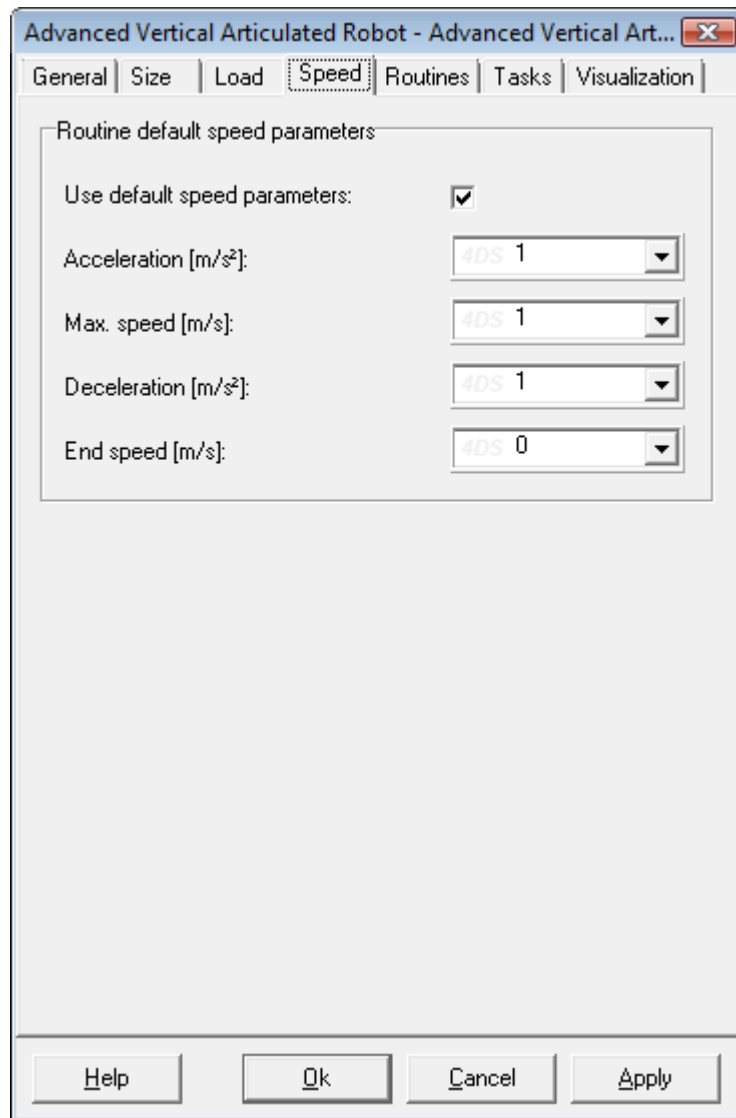
- 1: *FIFO --> first in first out*
- 2: *LIFO --> last in first out*
- 3: *Label minimum (picked atoms)*
- 4: *Label maximum (picked atoms)*
- 5: *Icon index minimum (picked atoms)*
- 6: *Icon index maximum (picked atoms)*

**Note:** Picked atoms are stored in the sub-atom Tool of Advanced Vertical Articulated Robot.

- *Use default load times*  
If this box is checked the default load and unload times are directly assigned to Routine edit fields in order to reduce the number of edit parameters.

**Note:** Use this parameter for simplified and fast Routine definition. Uncheck for detailed un- and load times.

- *Load time [s]*  
The default time (s) that is needed to load something to Advanced Vertical Articulated Robot. Assigned to the Routine load command when checkbox use default load times is checked.
- *Unload time [s]*  
The default time (s) that is needed to unload something from Advanced Vertical Articulated Robot. Assigned to the Routine unload command when checkbox use default load times is checked.
- *Location (x/y/z) of picked atom [m]*  
For picked atoms you can define location offsets for each individual x, y, and z.
- *Rotation (x/y/z) of picked atom [m]*  
For picked atoms you can define rotation offsets around each axis individual x, y, and z.



**Picture 15-4: The Advanced Vertical Articulated Robot speed parameters**

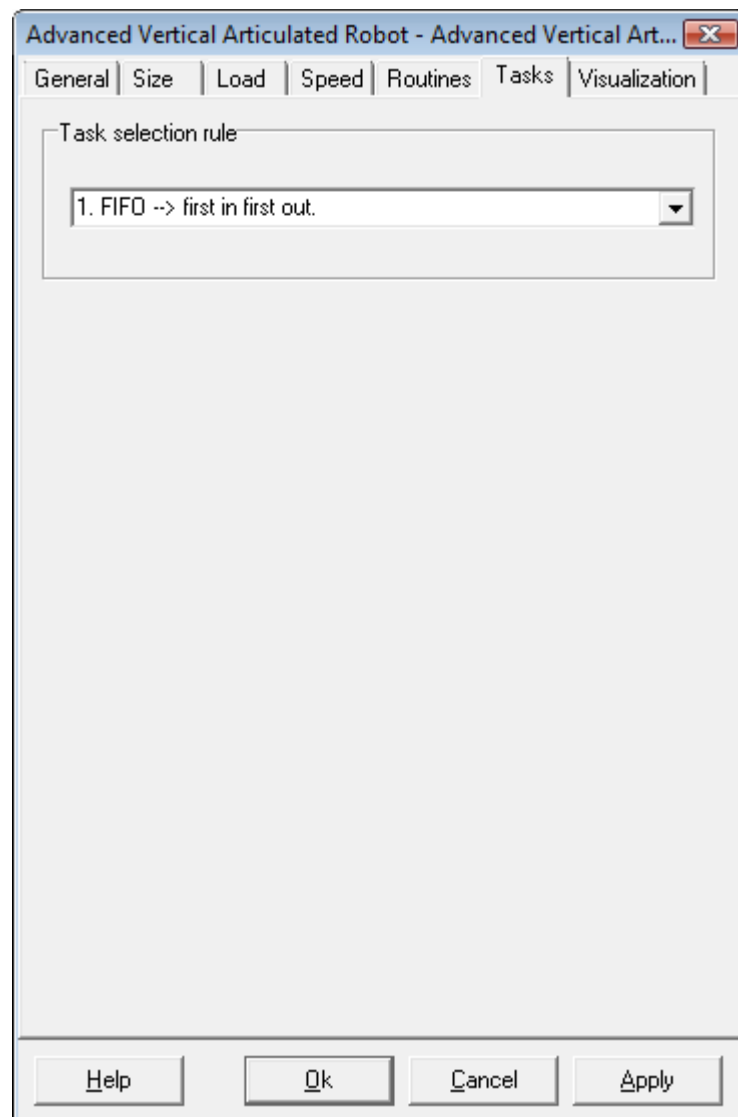
- *Use default speed parameters*  
If this box is checked the default speed parameters are directly assigned to routine edit fields in order to reduce the number of edit parameters.

**Note:** Use this parameter for simplified and fast Routine definition. Uncheck for detailed speed definitions.

- *Acceleration [m/s<sup>2</sup>]*  
Acceleration of Tool center in relation to origin of coordinate system.
- *Max. speed [m/s]*  
Maximum speed of Tool center in relation to origin of coordinate system.
- *Deceleration [m/s<sup>2</sup>]*  
Deceleration of Tool center in relation to origin of coordinate system.

- *End speed [m/s]*  
End speed of Tool center in relation to origin of coordinate system.

**Note:** The Advanced Vertical Articulated Robot uses the 'MovingTo' command to define the motions between the origin of the coordinate system and the center of the Tool. In 'MovingTo' the acceleration and deceleration are leading. The end speed will be tried to reach, but when this is impossible the end speed is the last speed taking acceleration and deceleration into account.



**Picture 15-5: The Advanced Vertical Articulated Robot task parameters**

Tasks are calls to the Advanced Vertical Articulated Robot to execute a specified Routine. There are two ways to create a Task. Pick and place Task are created every time an atom connected to an input channel of the robot causes an IcReady. Motion Tasks are created if there is an external call created with the 'Advanced\_Vertical\_Articulated\_Robot\_create\_motion\_task' command. If the robot is busy, the Tasks are stored until the robot is idle again. The Tasks selection rule determines what Tasks has to be done next.

- *Task selection rule*

This option lets execute the Tasks in a specified sequence. The internal queue of the Advanced Vertical Articulated Robot is sorted in the selected sequence before choosing the next Tasks to execute. Default is first in first out (FIFO). There are 10 strategies available:

1: *FIFO*

First in first out

2: *LIFO*

Last in first out

3: *Content minimum*

Atoms in the container with the smallest content first

4: *Content maximum*

Atoms in the container with the largest content first

5: *IC minimum*

Lowest input channel of robot first

6: *IC Maximum*

Highest input channel of robot first

7: *Label minimum (atom)*

The atom with the lowest value on label first

8: *Label maximum (atom)*

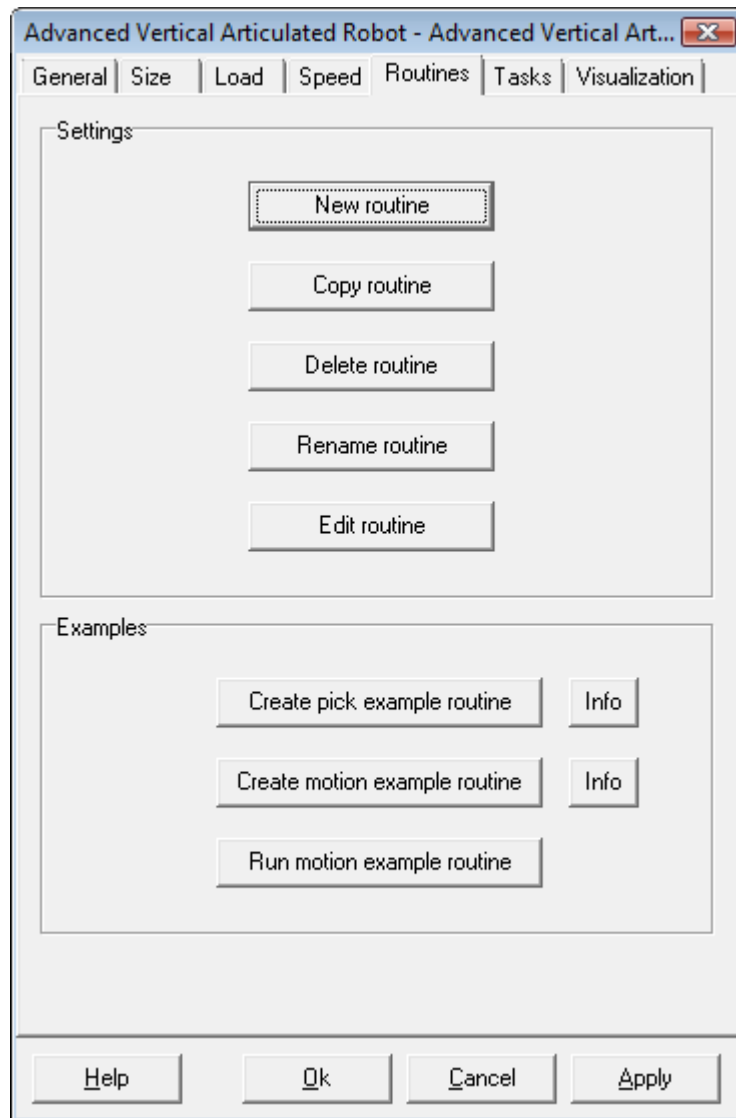
The atom with the highest value on label first

9: *Label minimum (container)*

Atoms in the container with the lowest value on label first

10: *Label maximum (container)*

Atoms in the container with the highest value on label first



**Picture 15-6: The Advanced Vertical Articulated Robot routine parameters**

Routines are sub atoms of the Advanced Vertical Articulated Robot. Storing all information about repeatable sequences into tables, the behavior becomes reproducible in order to execute frequently necessary actions like motions, loading, unloading or waiting.

- *New Routine*  
Click this button to create a new Routine. You will be asked to enter a Routine name. If the Routine has been created successfully, you are asked to edit commands to the Routine table by using the edit commands GUI.
- *Copy Routine*  
Click this button to copy a existing Routine with all its commands. Select the sample Routine via atom selector. If selection was successful, you will be asked to enter a different Routine name. If the Routine has been created successfully, you are asked to edit Commands to the Routine table by using the edit routines GUI.

- *Delete Routine*  
Click this button to delete a existing Routine with all its table-stored Commands.

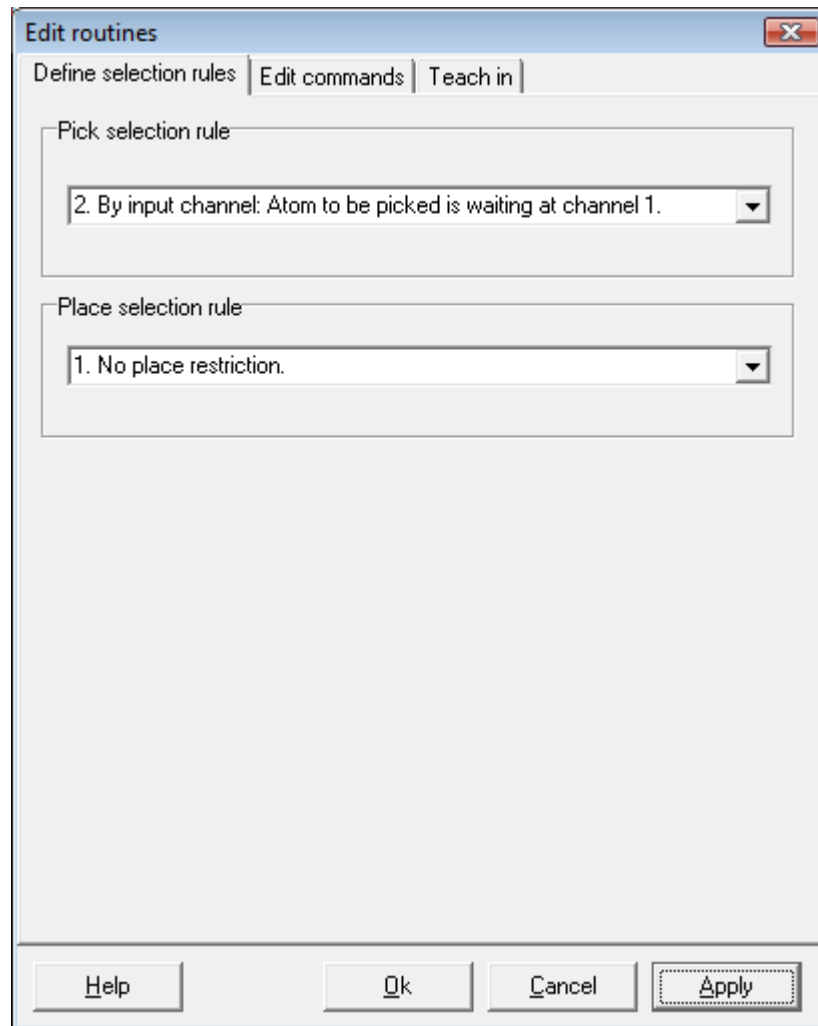
**Note:** The global reference to the Routine will be unregistered too. Ensure, that the Routine is not needed by any other Advanced Vertical Articulated Robot.

- *Rename Routine*  
Click this button to rename an existing Routine. You will be asked to enter a new Routine name.
- *Create pick example Routine*  
Click this button to create a Pick Routine example with the name 'Routine\_pick\_example'. If the Routine has been created successfully, you are asked to edit Commands to the Routine table by using the edit commands GUI.

**Warning!:** Pick example Routine only works if input channel 1 of tool is connected to pick atom and output channel 1 of Tool is connected to place atom.

- *Rename Routine*  
Click this button to rename an existing Routine. You will be asked to enter a new Routine name.
- *Create motion example Routine*  
Click this button to create a Motion Routine example with the name 'Routine\_motion\_example'. If the Routine has been created successfully, you are asked to edit Commands to the Routine table by using the edit commands GUI.
- *Run motion example Routine*  
Click this button to run the 'Routine\_motion\_example' Routine.
- *Edit Routine*  
Click this button to select the Routine you want to edit via atom selector. If selection was successfully, you are able to edit Commands to the Routine table by using the Advanced Vertical Articulated Robot Edit Routines GUI.





**Picture 15-7: The selection rule definition at the routine editor**

Any kind of action like loading, unloading, moving, delays are defined within Routines. The Advanced Vertical Articulated Robot Edit Routines GUI is the tool to apply, insert, copy, delete and edit Routine Commands. It is also possible to define selection rules for the Routines, so the Robot is able to select the Routine that fits best to the current Task. The Teach In functionality allows to create motions in an easy way.

**Note:** Some of the parameters used for the Edit Routines GUI are defined within the Advanced Vertical Articulated Robot GUI.

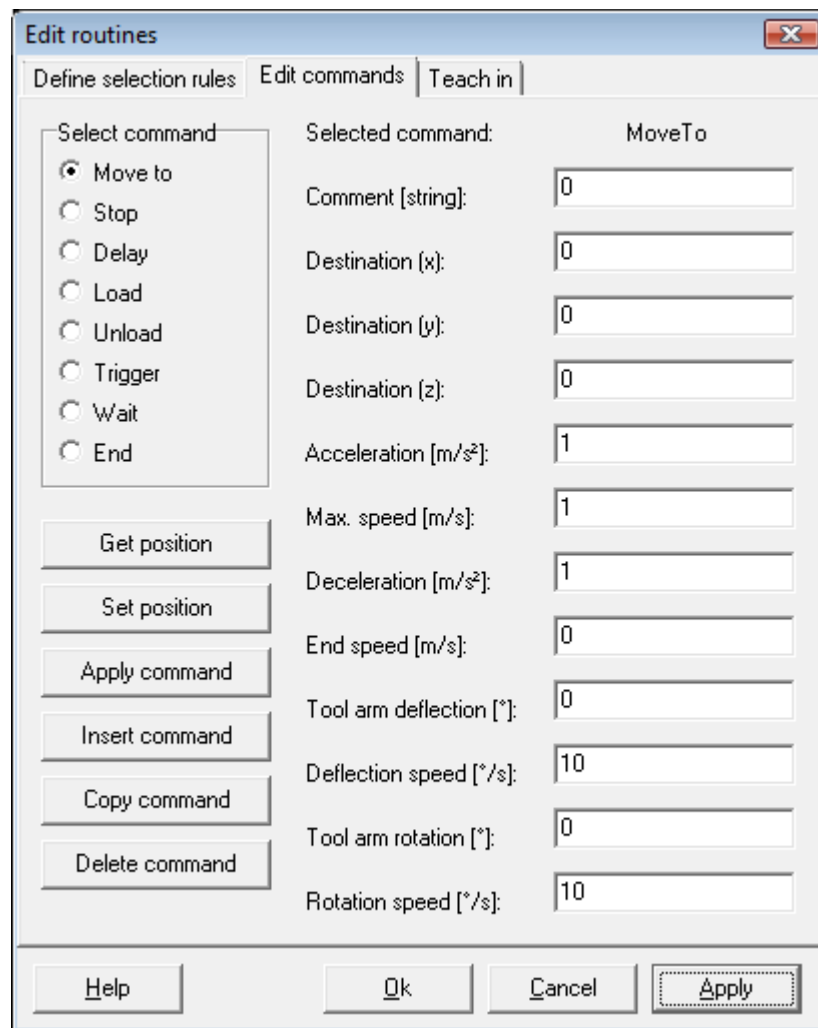
- *Pick selection rule*  
Determines the Pick condition to select the Routine. The Routine is selected, if the selected statement is true. There are 16 strategies available.
  - 1: *No pick restriction or motion routine*
  - 2: *By input channel*
  - 3: *By pick atom name*
  - 4: *By pick icon name*
  - 5: *By pick icon number*
  - 6: *By pick label value (direct)*

- 7: *By pick label value (conditional)*
- 8: *By pick label text*
- 9: *By pick location (direct)*
- 10: *By pick location (conditional)*
- 11: *By pick size (direct)*
- 12: *By pick size (conditional)*
- 13: *By pick content (direct)*
- 14: *By pick content (conditional)*
- 15: *Conditional statement*
- 16: *By user*

**Note:** Choose a Pick selection rule if there is more than one Routine available and the selection criteria can be found via input channel connection. The Pick selection rule can be combined with a Place selection rule to define advanced strategies.

- *Place selection rule*  
Determines the Place condition to select the Routine. The Routine is selected, if the selected statement is true. There are 11 strategies available.
  - 1: *No place restriction or motion routine*
  - 2: *By output channel*
  - 3: *By output channel fail safe*
  - 4: *By place location (direct)*
  - 5: *By place location (conditional)*
  - 6: *By place size (direct)*
  - 7: *By place size (conditional)*
  - 8: *By place content (direct)*
  - 9: *By place content (conditional)*
  - 10: *Conditional statement*
  - 11: *By user*

**Note:** Choose a Place selection rule if there is more than one Routine available and the selection criteria can be found via output channel connection. The Place selection rule can be combined with a Pick selection rule to define advanced strategies.



**Picture 15-8: The Advanced Vertical Articulated Robot command editor**

- *Select command*  
Select between the predefined kinds of Commands to define robot behavior. There are 8 predefined Commands available:
  - 1: *Move to*  
Defines any kind of robot motion
  - 2: *Stop*  
Stops any kind of robot motion
  - 3: *Delay*  
Makes the robot waiting for a specific time
  - 4: *Load*  
Executes picking of incoming products
  - 5: *Unload*  
Executes placing of outgoing products
  - 6: *Trigger*  
Allows the execution of 4DScript code
  - 7: *Wait*  
Let the robot wait for an external trigger to continue
  - 8: *End*  
Finishes a routine

Choose a Command to edit the necessary parameters. The number and kind of parameters depends on the type of selected Command.

Move to command:

- *Comment*  
Comment the Commands to make the Routines more readable.
- *Destination (x/y/z)*  
Tool center destination according to the origin of the coordinate system.
- *Acceleration [m/s<sup>2</sup>]*  
The tool centre acceleration to reach the destination.
- *Max. speed [m/s]*  
The tool centre maximum speed to reach the destination.
- *Deceleration [m/s<sup>2</sup>]*  
The tool centre deceleration to reach the destination.
- *End speed [m/s]*  
The Tool centre end speed to reach the destination.
- *Tool arm deflection [°]*  
Absolute deflection of the tool arm at the destination.
- *Deflection speed [°/s]*  
The deflection speed to reach the absolute Tool arm deflection.
- *Tool arm rotation [°]*  
Absolute rotation of the Tool arm at the destination.
- *Rotation speed [°/s]*  
The rotation speed to reach the absolute Tool arm rotation.

**Note:** You are able to edit detailed speed parameters, if the default speed parameters checkboxes is selected at the Advanced Vertical Articulated Robot GUI.

**Note:** The Advanced Vertical Articulated Robot uses the 'MovingTo' Command to define the motions between the origin of the coordinate system and the centre of the Tool. In 'MovingTo' the acceleration and deceleration are leading. The end speed will be tried to reach, but when this is impossible the end speed is the last speed taking acceleration and deceleration into account.

**Note:** You can get the current position of the Tool into the edit fields by click on the Get position button.

**Note:** You can set the position of the Tool according to the edit fields by click on the Set position button.

Stop command:

- *Comment*  
Comment the Commands to make the Routines more readable.

Delay command:

- *Comment*  
Comment the Commands to make the Routines more readable.
- *Time [s]*  
The delay before the Routines is continued with the next command (row).

Load command:

- *Comment*  
Comment the Commands to make the Routines more readable.
- *Time [s]*  
The load time of the atom to be picked.
- *Channel [index]*  
The input channel of the robot, where the atom is waiting.

**Note:** You are able to edit detailed load parameters, if the default load parameters checkboxes is selected at the Advanced Vertical Articulated Robot GUI.

Unload command:

- *Comment*  
Comment the Commands to make the Routines more readable.
- *Time [s]*  
The unload time of the atom to be placed.
- *Channel [index]*  
The output channel of the robot, where the atom has to be sent to.

**Note:** You are able to edit detailed unload parameters, if the default load parameters checkboxes is selected at the Advanced Vertical Articulated Robot GUI.

Trigger command:

- *Comment*  
Comment the Commands to make the Routines more readable.

- *Trigger [4DScript]*  
The code that has to be executed before the Routine is continued with the next Command. Select between the predefined Commands defined in the kernel function 'getTriggerEnterExit' or enter your own 4DScript code.

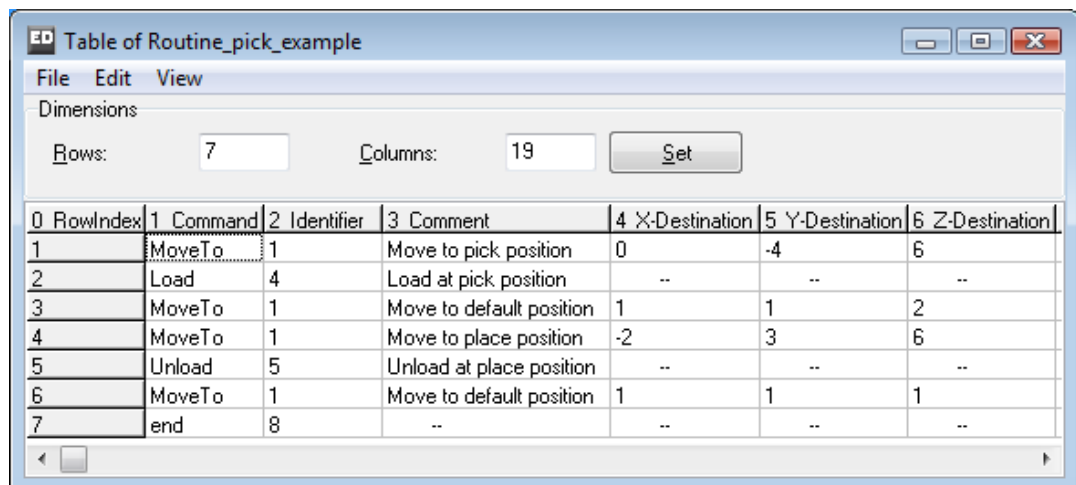
Wait command:

- *Comment*  
Comment the Commands to make the Routines more readable.
- *Awaited message [string]*  
The text message that has to arrive in the OnMessage event handler of the Advanced Vertical Articulated Robot before the Routine is continued with the next Command.

End command:

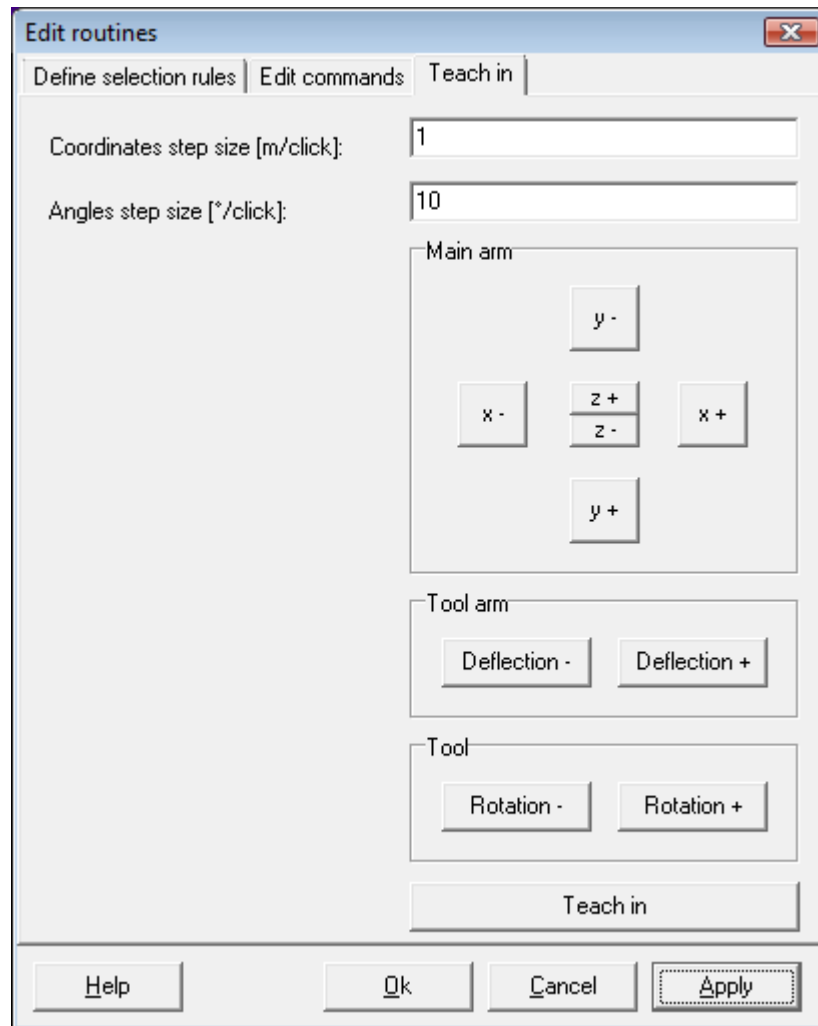
If you select the End Command no parameters can be changed, but another row is added to the Routine in order to set a flag, that that there will be no further Commands. The Robot is set to status Idle and it searches for other Tasks available.

There are four buttons to apply, insert, copy and delete Commands. On click you are asked to edit the row index of the Command that you want to apply, insert, copy or delete.



0 RowIndex	1 Command	2 Identifier	3 Comment	4 X-Destination	5 Y-Destination	6 Z-Destination
1	MoveTo	1	Move to pick position	0	-4	6
2	Load	4	Load at pick position	--	--	--
3	MoveTo	1	Move to default position	1	1	2
4	MoveTo	1	Move to place position	-2	3	6
5	Unload	5	Unload at place position	--	--	--
6	MoveTo	1	Move to default position	1	1	1
7	end	8	--	--	--	--

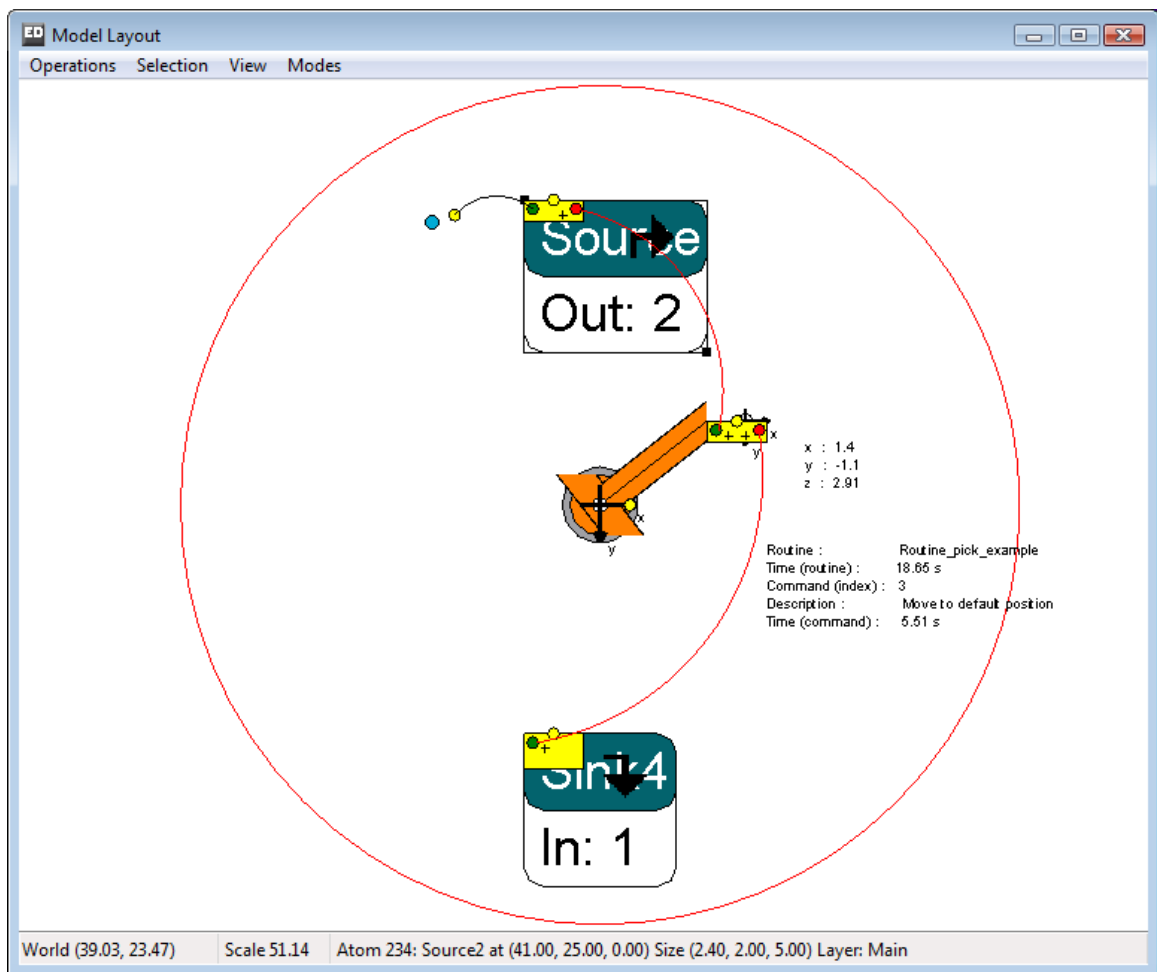
**Picture 15-9: Routine example (pick and place)**



**Picture 15-10: Routine teach panel (move to command)**

The Teach In functionality allows to create motion commands in an easy way.

- *Coordinates step size [m/click]*  
On click of main arm buttons the tool position will move according to the coordinates step size.
- *Angles step size [°/click]*  
On click of tool arm buttons the tool deflection will rotate according to the angles step size. On click of tool buttons the tool will rotate according to the angles step size.
- *Teach in button*  
On click all motion parameters are updated into the edit fields of the Edit Command page and the destination is stored to the active routine as Move to Command.



Picture 15-11: 2D Layout of a running pick example



## 16 THE ADVANCED ASRS ATOM

---

Advanced ASRS - Advanced ASRS1

General | Storage | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

Atom name: Advanced ASRS1

Alias

Create alias: ☒

Use atom name: ☒

Alias name: untitled

Triggers

Trigger on reset: IDS 0

Help Rebuild Ok Cancel Apply

**Picture 16-1: The Advanced ASRS atom**

An ASRS is a rail bound single track vehicle to store and retrieval goods of warehouse racks. Commonly used layout contains one locations for Inbound, one storage and retrieval machine (SRM) with up to 2 High Rise Racks on the sides and one location for Outbound. There are parameters to edit size, position and layout of all elements.

The SRM is made of 3 atoms. The SRM atom itself uses the rail track to move along aisle (x-axis) to the assigned column of the Racks. The Hoist atom is lifting up to the allocated row of the rack (z-axis). The Shuttle atom reaches into the racks to store or retrieve the goods (y-axis). Each axis has its own parameters to control acceleration, maximum speed and deceleration. Detailed load times can be defined for every transfer point inside the system. Due to high level of automation ASRS systems are driven without human interference, but controlled by strategies for Inbound, movement and Outbound of goods.

**Important:** The Advanced ASRS layout is based on rows and columns of warehouse racks coordinate system. Use rows and columns to define the dimension of Racks. Also use coordinates to define Inbound and Outbound position.

**Note:** This atom is part of ED Logistics. However ED Logistics also has a single Warehouse atom to model non automated storage systems.

- *Atom Name*  
The name of the atom.

**Note:** There are 2 ways to access the ASRS's values from another atom. The first possibility is to connect an atom to the ASRS's central channel.

Suppose you connected output channel 1 of an atom XYZ to the ASRS, then you can refer to ASRS from XYZ as follows: Out(1, c).

Since ASRS's are referenced from many atoms this is not a comfortable way to get or set the required data. You can have the ASRS automatically create 4DScript functions (aliases) to be able to access the data directly without any channel connections.

Example:

If you name the ASRS MyASRS, and you check the box Create Alias 3 4DScript functions will be created: MyASRS, SetMyASRS, and RefMyASRS. You can now use these functions to access the table from another atom without connecting channels.

MyASRS is used to obtain a cell value. SetMyASRS is used to set a table value, and RefMyASRS is used to refer to the table atom itself.

- *Trigger on reset*  
A rule that determines what kind of action needs to be executed when the ASRS atom is reset. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

**Advanced ASRS - Advanced ASRS1**

General | **Storage** | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

**High Rise Racks**

Location

☐ 1. Both sides of SRM

☒ 2. Left side of SRM only

☐ 3. Right side of SRM only

Z location [m]:

Rotation [°]:

**Dimensions**

Number of rows:

Number of columns:

**Sizes**

Height (z) of bin locations [m]:

Width (x) of bin locations [m]:

Depth (y) of bin locations [m]:

Width (y) of SRM aisle [m]:

**Restrictions**

Edit -1 for non available bin locations:

**Picture 16-2: The Advanced ASRS storage parameters**

- *Location*  
The layout of the High Rise Racks. Select between left or right High Rise Rack or define High Rise Racks on both sides of the SRM atom.
- *Z location [m]*  
The z-location of the ASRS.
- *Rotation [°]*  
At default the ASRS is positioned from left to right in your screen. However you

can also rotate the atom to display the flow of atoms over the ASRS matches reality.

#### Dimensions:

- *Nr of rows*  
Number of bin locations in vertical direction (z) in each High Rise Rack.
- *Nr of columns*  
Number of bin locations in horizontal direction (x) in each High Rise Rack.

#### Sizes:

- *Height (z) of bin locations (m)*  
The z-size of all bin locations inside each High Rise Rack. Determines (together with nr of rows) the total height of High Rise Racks.
- *Width (x) of bin locations (m)*  
The x-size of all bin locations inside each High Rise Rack. Determines (together with nr of columns) the total length of High Rise Racks.
- *Depth (y) of bin locations (m)*  
The y-size of all bin locations inside each High Rise Rack. Determines the total depth of High Rise Racks.
- *Width (y) of SRM aisle (m)*  
The y-size of the aisle between the High Rise Racks. Determines the space for SRM movement.
- *Restrictions*  
Due to constructional limitations (pillars, passages, etc.) it is frequently not possible to use all bin locations of the High Rise Racks. Edit -1 into the cells of the High Rise Rack's table to disable place strategies to assign goods to those bin locations.

**Note:** Capacity of High Rise Racks is calculated without disabled bin locations. Place strategies consider the bowdlerized capacity when calculation the fill level of a High Rise Rack.

Advanced ASRS - Advanced ASRS1

General | Storage | **Inbound** | Outbound | Motion | Load | Pick | Place | Visualization

Location

☒ 1. Left side of aisle

☐ 2. Right side of aisle

Position

Row:

Column:

Settings

Input strategy:

Triggers

Trigger on entry:

Trigger on exit:

Help Rebuild Ok Cancel Apply

**Picture 16-3: The Advanced ASRS inbound parameters**

- *Location*  
The location of the Inbound atom. Select between left or right side of the aisle.

Position:

- *Row*  
Locations in vertical direction (z) based on the coordinate system of the High Rise Racks.

- *Column*  
Locations in horizontal direction (x) based on the coordinate system of the High Rise Racks.
- *Input strategy*  
A rule that determines how Product atoms may be let into the Inbound atom of the ASRS. There are a number of rules predefined (see also Input strategy), but you can also create your own rule via a 4DScript expression.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Inbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.
- *Trigger on exit*  
A rule that determines what kind of action needs to be executed when a Product atom exits the Inbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

Advanced ASRS - Advanced ASRS1

General | Storage | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

Location

☐ 1. Left side of aisle

☒ 2. Right side of aisle

Position

Row:

Column:

Settings

Send to:

Triggers

Trigger on entry:

Trigger on exit:

Help Rebuild Ok Cancel Apply

**Picture 16-4: The Advanced ASRS outbound parameters**

- *Location*  
The location of the Outbound atom. Select between left or right side of the aisle.

Position:

- *Row*  
Locations in vertical direction (z) based on the coordinate system of the High Rise Racks.

- *Column*  
Locations in horizontal direction (x) based on the coordinate system of the High Rise Racks.
- *Send to*  
A rule that determines to which output channel a Product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Outbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

#### *Trigger on exit*

A rule that determines what kind of action needs to be executed when a Product atom exits the Outbound atom. There are a number of rules predefined (see also Trigger on



entry and exit), but you can also create your own rule via a 4DScript

Advanced ASRS - Advanced ASRS1

General | Storage | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

Location

☐ 1. Left side of aisle

☒ 2. Right side of aisle

Position

Row:

Column:

Settings

Send to:

Triggers

Trigger on entry:

Trigger on exit:

Help Rebuild Ok Cancel Apply

expressio

**Picture 16-4: The Advanced ASRS outbound parameters**

- *Location*  
The location of the Outbound atom. Select between left or right side of the aisle.

Position:

- *Row*  
Locations in vertical direction (z) based on the coordinate system of the High Rise Racks.

- *Column*  
Locations in horizontal direction (x) based on the coordinate system of the High Rise Racks.
- *Send to*  
A rule that determines to which output channel a Product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Outbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.

*Trigger on exit*

A rule that determines what kind of action needs to be executed when a Product atom exits the Outbound atom. There are a number of rules predefined (see also Trigger on

entry and exit), but you can also create your own rule via a 4DScript

expression

Advanced ASRS - Advanced ASRS1

General | Storage | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

Location

☐ 1. Left side of aisle

☒ 2. Right side of aisle

Position

Row: 1

Column: 0

Settings

Send to: 2. An open channel (First channel first): see...

Triggers

Trigger on entry: 4DS 0

Trigger on exit: 4DS 0

Help Rebuild Ok Cancel Apply

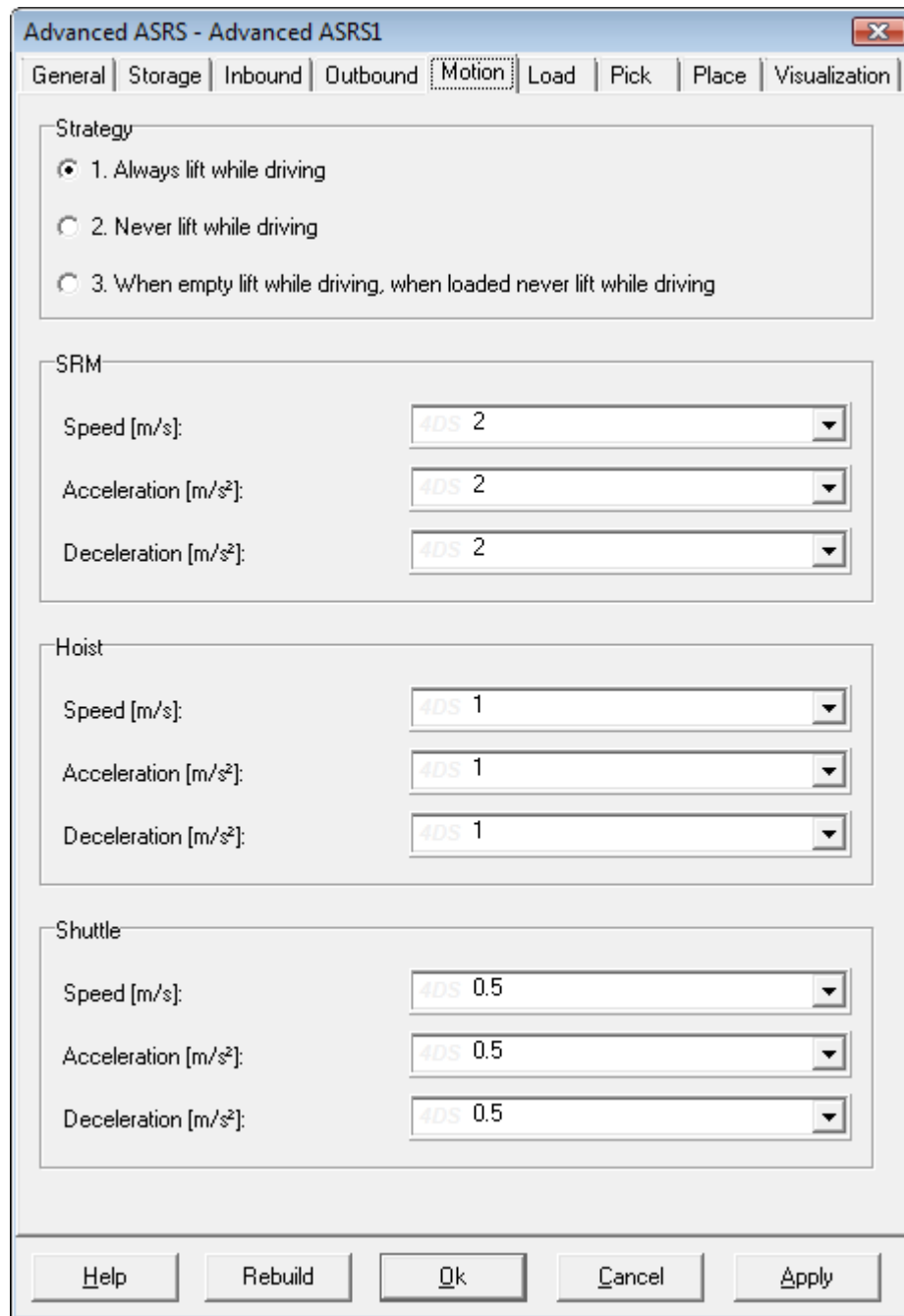
Picture 16-4: The Advanced ASRS outbound parameters

- *Location*  
The location of the Outbound atom. Select between left or right side of the aisle.

Position:

- *Row*  
Locations in vertical direction (z) based on the coordinate system of the High Rise Racks.

- *Column*  
Locations in horizontal direction (x) based on the coordinate system of the High Rise Racks.
- *Send to*  
A rule that determines to which output channel a Product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a Product atom enters the Outbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.
- *Trigger on exit*  
A rule that determines what kind of action needs to be executed when a Product atom exits the Outbound atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule via a 4DScript expression.



**Picture 16-5: The Advanced ASRS motion parameters**

- *Motion strategy*  
A rule that determines how the SRM atom is coordinates the movements of lifting and driving. There are 3 strategies available:  
1: *Always lift while driving*  
2: *Never lift while driving*  
3: *When empty lift while driving, when loaded never lift while driving*
- *SRM*  
The speed, the acceleration and the deceleration can be defined for x-direction.

- *Hoist*  
The speed, the acceleration and the deceleration can be defined for z-direction.
- *Shuttle*  
The speed, the acceleration and the deceleration can be defined for y-direction.

**Advanced ASRS - Advanced ASRS1**

General | Storage | Inbound | Outbound | Motion | Load | Pick | Place | Visualization

**Strategy**

**Order**

- ☒ 1. Inbound and Outbound alternately (if both is possible)
- ☐ 2. Inbound preferred (if both is possible)
- ☐ 3. Outbound preferred (if both is possible)
- ☐ 4. Inbound only (never roll out)
- ☐ 5. Outbound only (never roll in)
- ☐ 6. Conditional statement defined by label
- ☐ 7. Outbound preferred (external calls only)

**Labels**

Condition: n/a

**Time**

Load time from inbound [s]:

Unload time to rack [s]:

Load time from rack [s]:

Unload time to outbound [s]:

Help Rebuild Ok Cancel Apply

**Picture 16-6: The Advanced ASRS load parameters**

- *Order strategy*  
A rule that determines the interaction between Inbound and Outbound of products. There are 6 strategies available:

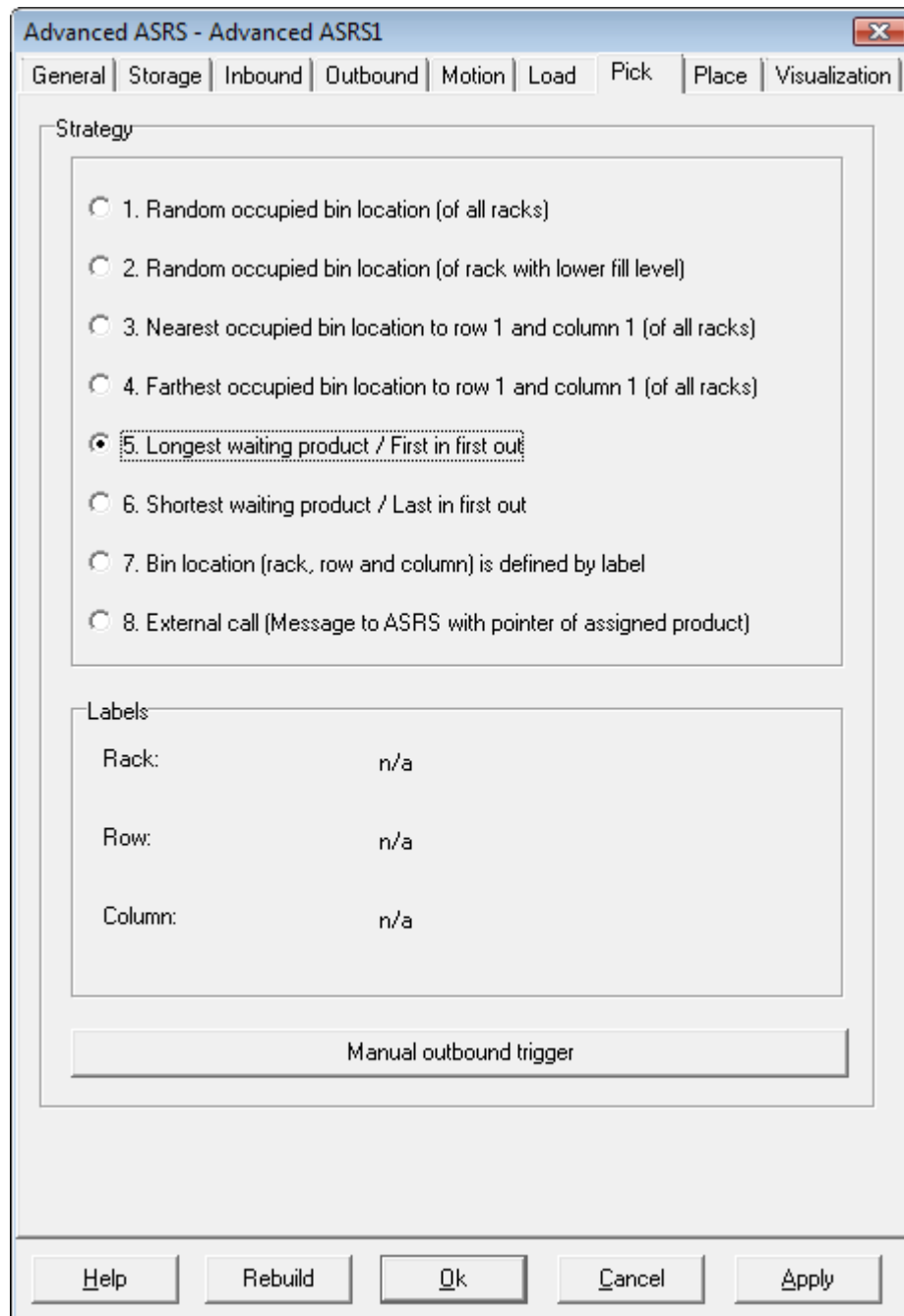
- 1: *Inbound and Outbound alternately (if both is possible)*
- 2: *Inbound preferred (if both is possible)*
- 3: *Outbound preferred (if both is possible)*
- 4: *Inbound only (never roll out)*
- 5: *Outbound only (never roll in)*
- 6: *Conditional statement defined by label*

- *Condition*

Label to be executed when order strategy nr 6 is the active strategy. There are a number of rules predefined, but you can also create your own rule via a 4DScript expression.

Endurance:

- *Load Time*  
The time (s) that is needed to load something from Inbound to SRM.
- *Unload Time*  
The time (s) that is needed to unload something from SRM to High Rise Racks.
- *Load Time*  
The time (s) that is needed to load something from High Rise Racks to SRM.
- *Unload Time*  
The time (s) that is needed to unload something from SRM to Outbound.



**Picture 16-7: The Advanced ASRS pick parameters**

- Pick strategy**  
 A rule that determines the Outbound action of ASRS by selecting the Product (and its location) for the next roll out movement. There are 8 strategies available:
  - 1: *Random occupied bin location (of all High Rise Racks)*
  - 2: *Random occupied bin location (of High Rise Racks with lower fill level)*
  - 3: *Nearest occupied bin location to row 1 and column 1 (of all High Rise Racks)*
  - 4: *Farthest occupied bin location to row 1 and column 1 (of all High Rise Racks)*



- 5: *Longest waiting product / First in first out*
- 6: *Shortest waiting product / Last in first out*
- 7: *Bin location (High Rise Rack, row and column) is defined by label*
- 8: *External call (Message to ASRS with pointer of assigned product)*

Labels:

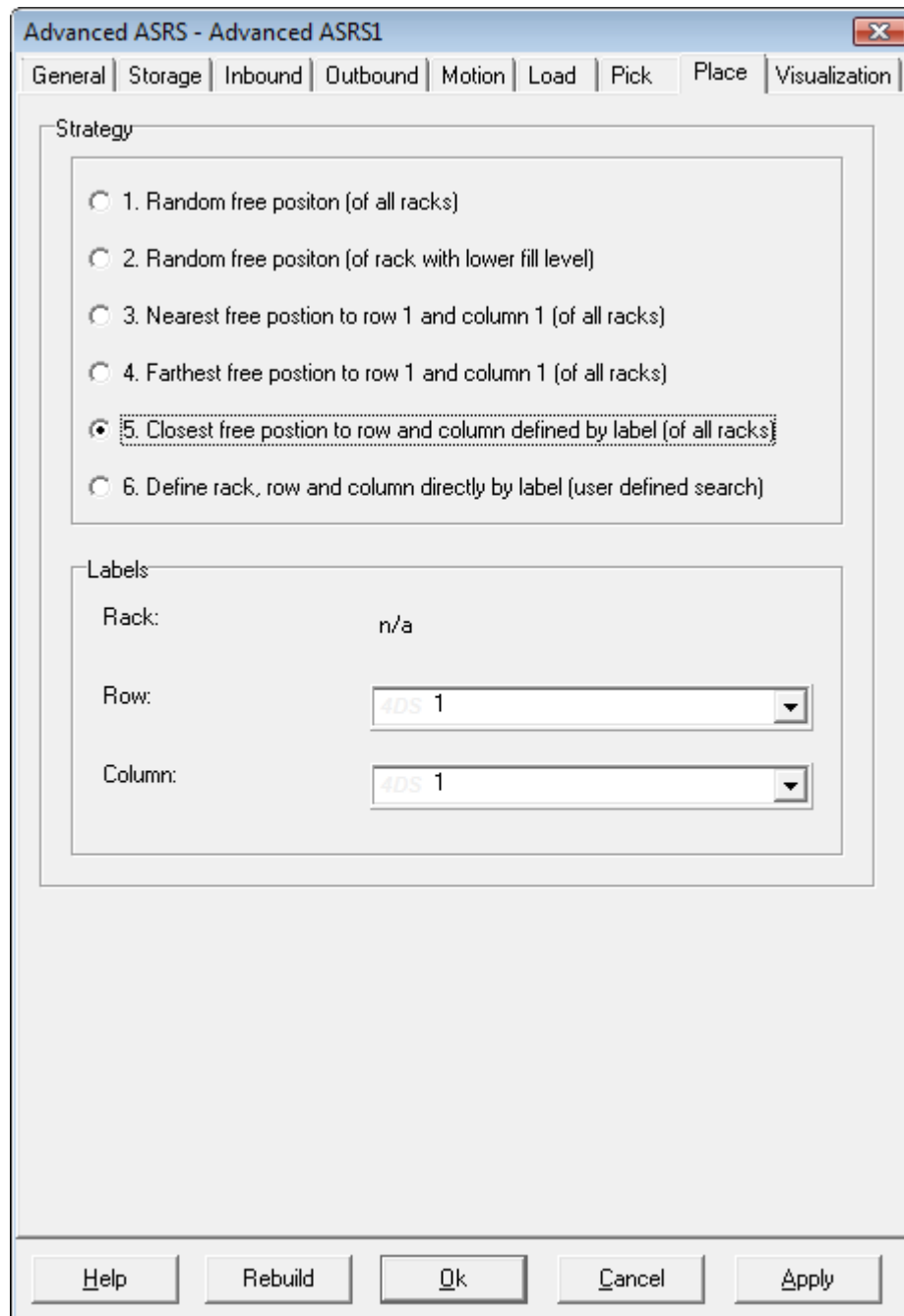
- *Rack*  
Label to be executed when order strategy nr 7 is the active strategy. There are a number of rules predefined to select the index of the High Rise Rack, but you can also create your own rule via a 4DScript expression.

**Note:** The result of the rule to select a rack has always to be an index between 1 and 2. The left High Rise Rack is always assigned by index 1. The right High Rise Rack is always assigned by index 2. Be sure that the resulting index refers to an existing High Rise Rack.

- *Row*  
Label to be executed when order strategy nr 7 or 8 is the active strategy. There are a number of rules predefined to select the index of the row, but you can also create your own rule via a 4DScript expression.
- *Column*  
Label to be executed when order strategy nr 7 or 8 is the active strategy. There are a number of rules predefined to select the index of the column, but you can also create your own rule via a 4DScript expression.

**Note:** The result of the rule to select a row or column has always to be  $> 0$  and within the dimensions of your High Rise Racks. Be sure that the resulting index of row and column refers to a non assigned/occupied bin location. Occupied bin locations of the High Rise Rack tables always contain the pointer of the assigned product.

- *Manual outbound trigger*  
Often picking commands are coming from external controller and not from the ASRS itself. When pick strategy nr 8 is the active strategy, ASRS picking is triggered by external messages containing the pointer of the product that has to be rolled out. The button is doing the same as an external controller and sends a message to the ASRS with the pointer of the next product to roll out. Helpful to test ASRS functionality before external controller has been implemented. Furthermore it contains a helpful syntax example for getting started with external product search.



**Picture 16-8: The Advanced ASRS place parameters**

- *Place strategy*  
A rule that determines the Inbound action of ASRS by selecting the bin location (High Rise Rack, row and column) for incoming products. There are 6 strategies available:
  - 1: *Random free position (of all High Rise Racks)*
  - 2: *Random free position inside (of High Rise Rack with lower fill level)*
  - 3: *Nearest free position to row 1 and column 1 (of all High Rise Racks)*
  - 4: *Farthest free position to row 1 and column 1 (of all High Rise Racks)*
  - 5: *Closest free position to row and column defined by label (of all High Rise Racks)*

6: *Define High Rise Rack, row and column directly by label (user defined search)*

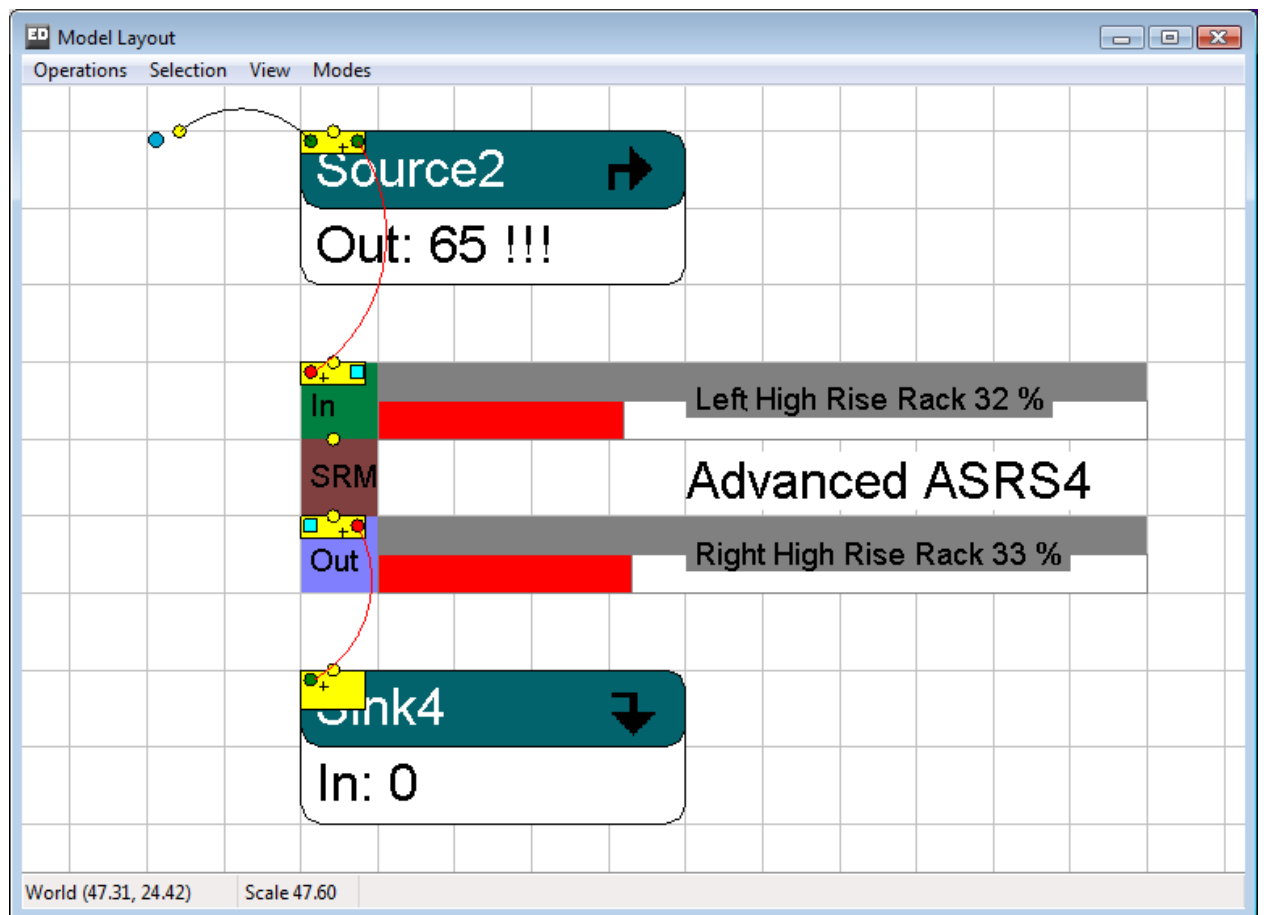
Labels:

- *Rack*  
Label to be executed when order strategy nr 5 is the active strategy. There are a number of rules predefined to select the index of the High Rise Rack, but you can also create your own rule via a 4DScript expression.

**Note:** The result of the rule to select a High Rise Rack has always to be an index between 1 and 2. The left High Rise Rack is always assigned by index 1. The right High Rise Rack is always assigned by index 2. Be sure that the resulting index refers to an existing High Rise Rack.

- *Row*  
Label to be executed when order strategy nr 5 or 6 is the active strategy. There are a number of rules predefined to select the index of the row, but you can also create your own rule via a 4DScript expression.
- *Column*  
Label to be executed when order strategy nr 5 or 6 is the active strategy. There are a number of rules predefined to select the index of the column, but you can also create your own rule via a 4DScript expression.

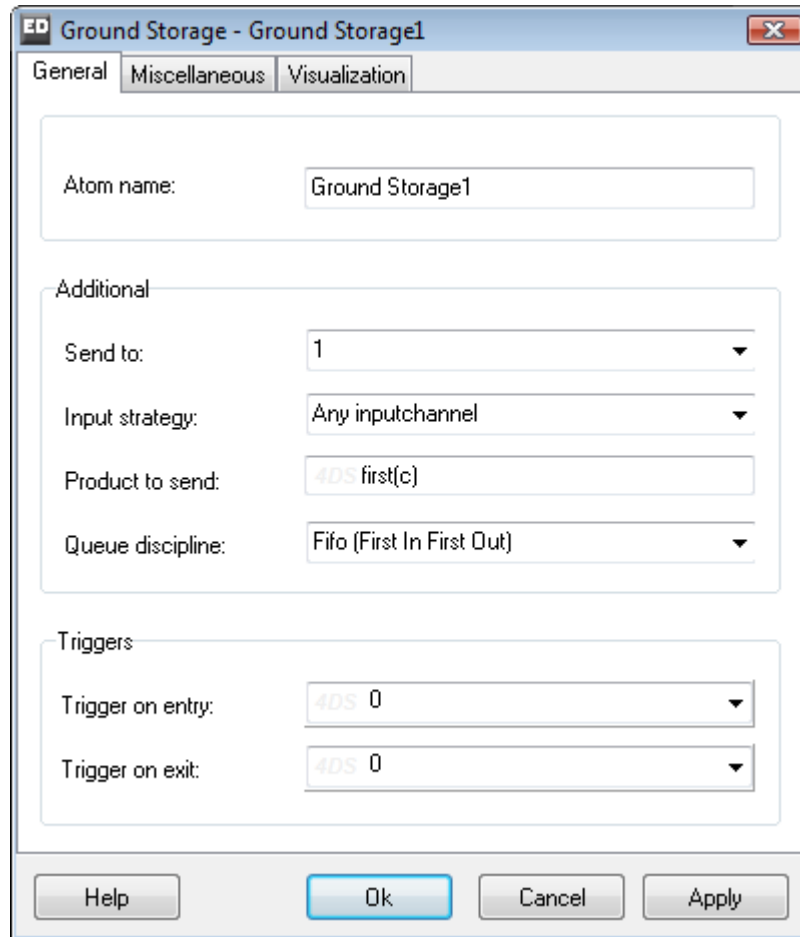
**Note:** The result of the rule to select a row or column has always to be  $> 0$  and within the dimensions of your High Rise Racks. Be sure that the resulting index of row and column refers to a non assigned/occupied and available (not -1) bin location. Non occupied bin location of the High Rise Rack tables always has the value 0.



Picture 16-9: The Advanced ASRS 2D Model Layout

## 17 THE GROUND STORAGE ATOM

---

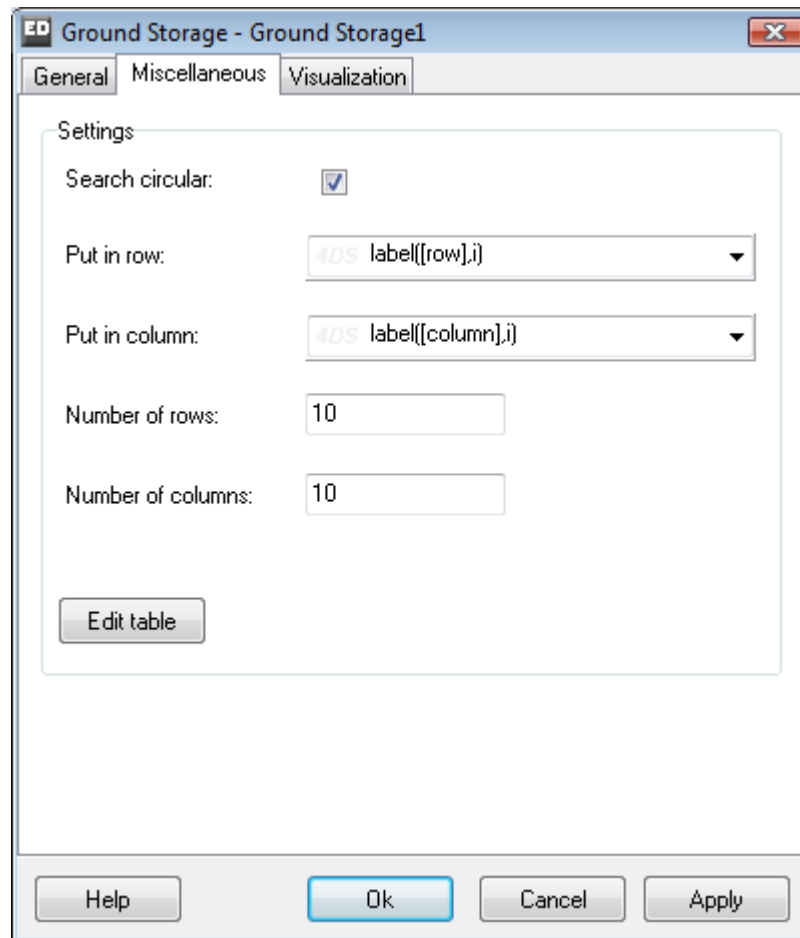


Picture 17-1: The Ground Storage atom

The Ground Storage atom stores product atoms that enter at a specific location. If the Ground Storage is full its input is closed. Each storing position (a storing position is treated as a cell in the internal table) can contain one product. Specific storing positions can be deactivated to simulate e.g. columns in a building. Just click on the button 'Edit Table' and write '-1' in the cell which you want to deactivate. Storing positions are represented as cells in that table. If the Ground Storage has output channels, products are sent out. You can specify which product to send out.

- *Atom Name*  
The name of the atom.
- *Send to*  
A rule that determines to which output channel a product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.

- *Input strategy*  
A rule that determines to which output channel a product atom needs to be sent. There are a number of rules predefined (see also Send to), but you can also create your own rule via a 4DScript expression.
- *Product to send*  
When the Ground Storage is ready to send a new product, this code will be evaluated to see which product should be send. The sequence of the products depends on the Queue Discipline.
- *Queue Discipline*  
When a product enters, this code will specify the way products are ranked in the Ground Storage. This is not related to the place (row and column) of the product in the warehouse.
- *Trigger on entry*  
A rule that determines what kind of action needs to be executed when a product atom enters the Ground Storage atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule by writing a 4DScript expression.
- *Trigger on exit*  
A rule that determines what kind of action needs to be executed when a product atom exits the Ground Storage atom. There are a number of rules predefined (see also Trigger on entry and exit), but you can also create your own rule using 4DScript.



**Picture 17-2: The Ground Storage bin location parameters**

- *Search circular*

The collision check can be turned on and off here. When a product enters the atom and the check is on, it is checked if the desired storing place is free. If it is full, it is checked in increasing circles around the desired storing place if a free one can be found. If the check is off and the desired storing place is full, the next free storing place is taken. If you want to have a nicer visualization, you can make the collision check before the product enters the Ground Storage atom. You have to call the function 'GroundStorage\_\_SearchForAFreeStoragePlace(AtomByName([Ground Storage1], Model), i)' on the atom that is connected to the 'Ground Storage'. The function expects the labels 'row' and 'column' on the product (i). They need to be set before you call the function.

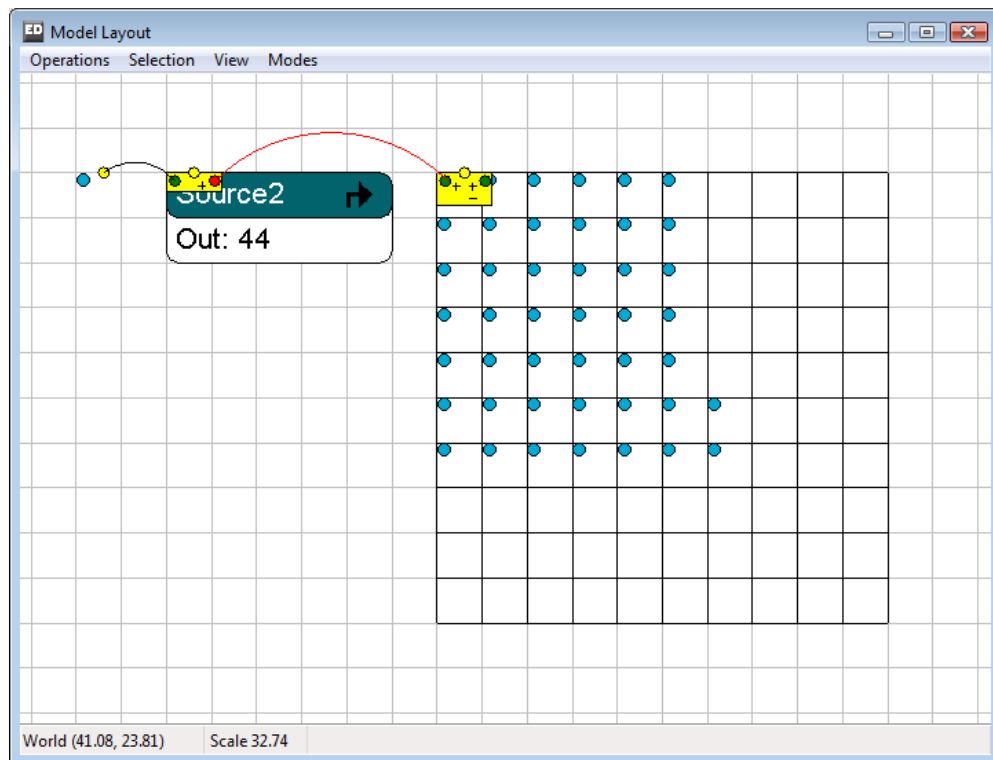
- *Important*

If you use the above function, checking or un-checking the Control 'Search circular' has no effect. This is because the collision check happens in the function and the function is called before the product enters the 'Ground Storage'.

- *Put in row*

It can be defined in which row an incoming product should be placed with a value or an expression here. If this row is completely full, the next row is chosen. If the value is 0, the search for an empty storing place (cell) starts always with row 1.

- *Put in column*  
It can be defined in which column an incoming product should be placed. Values or expressions are allowed here. If this column is completely full, the next column is chosen. If the value is 0, the search for an empty storing place (cell) starts always with column 1.
- *Number of rows*  
This is the number of locations in vertical direction in your Ground Storage.
- *Number of columns*  
This is the number of locations in horizontal direction in your Ground Storage.
- *Edit table*  
This will open a window with a table in which you can view the content of the Ground Storage. The ID numbers of the atoms in the Ground Storage are displayed in the table. You can deactivate storing places by writing -1 in the cells.



**Picture 17-3: The Ground Storage 2D Model Layout**



## Bijlage 3      A first start in 4DScript

---

4DScript is the programming language of Enterprise Dynamics. Everything that is executed in Enterprise Dynamics is or can be done via 4DScript.

This document provides for beginners the structure of 4DScript and a list of often used commands, illustrated with examples.

We start with the syntax rules together with the mathematical and logical operators. Second the concept of referencing is illustrated with examples. The third part consists of the most common commands in ED

### 1.      The basics of 4DScript

The basic syntax of 4DScript is simple and is valid at any position where you can write 4DScript:

- 1      the language contains a number of words (commands);
- 2      these 4DScript commands can have (up to 24) parameters;
- 3      the parameters are placed between parenthesis ( );
- 4      parameters can be *values*, *strings* or *expressions* (other 4DScript words)
- 5      parameters are always separated by commas;
- 6      if a parameter should be interpreted as a string, the string parameters are always placed between square brackets [ ]. If the parameter should be executed as 4DScript code, the parameter is written in a normal fashion;
- 7      comments can be placed between squiggly brackets { }.

These rules apply everywhere and are always valid. However, for some statements these rules do not result in a very readable code. In order to make parts of the code easier to understand for others, the 4DScript syntax makes an exception in some cases. This applies in particular to mathematical and logical symbols.

#### Mathematical symbols

Consider the following valid statements:

$+(12,7)$	results in 19	A more natural way: 12+7
$+/(100,10),6)$	results in 16	100/10+6    or 6+100/10

All the special mathematical operators are evaluated in the following order:

*	=	multiplication
/	=	division
+	=	addition
-	=	subtraction

## Logical symbols

For logical operators (like >, <, =) it is also possible to disregard the rule that the 4DScript command needs to be written first and then the parameters

Consider the following valid statement:

>(10,6)      can also be written in the 'normal' fashion as:      10>6  
<=(23,12)    can also be written as:      23<=12

This example can be extended to more 4DScript logical symbols:

=	=	equal
>	=	larger than
<	=	smaller than
>=	=	larger than or equal to
<=	=	smaller than or equal to
<>	=	not equal

### **and**

syntax:      and(e1,e2)

Returns 1 if e1 and e2 are true (1), returns 0 if not. If e1 is not true, then e2 is not evaluated. Instead of writing and(e1,e2) the user can also write e1 **and** e2.

See also **or**

So, both commands are used as usual...

### **max**

Syntax:      max(e1,e2)

Returns the maximum of e1 and e2. See also **min**

We end this chapter with a few commands on time and time conversion. Remember that ED 'thinks' in seconds, but it is easy to convert time:

1.      **time**      returns the current model time in seconds
2.      **mins(e1)**    returns e1 (minutes) in seconds: multiplies e1 with 60.
3.      **hr(e1)**      returns e1 (hours) in seconds: multiplies e1 with 3600.

## 2. Referencing

A very important subject in ED is referencing: if we are talking about the 'next' atom, what is our viewpoint? It gets more complicated if there are atoms contained in other atoms like products in a queue.

We have seen the letters c and i in different statements, now we will explain them:

'c' refers to the current atom where the statement is written on

'i' refers to the involved atom, the one entering or leaving the current atom

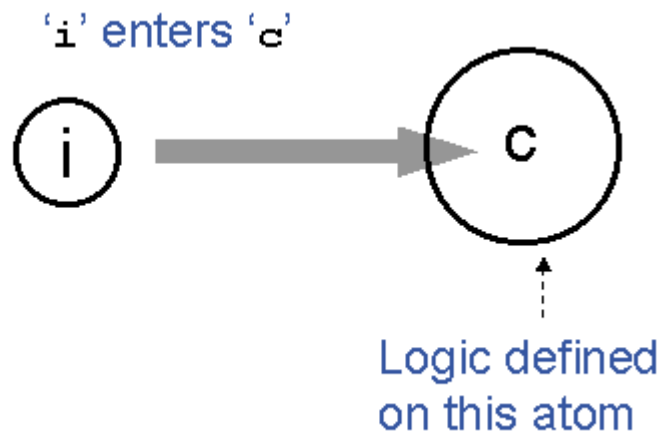


Figure 1

To keep it simple: the 'i' is only used in statements written on Trigger on Entry or Trigger on Exit!

The key to understanding the concept of referencing is the next example:

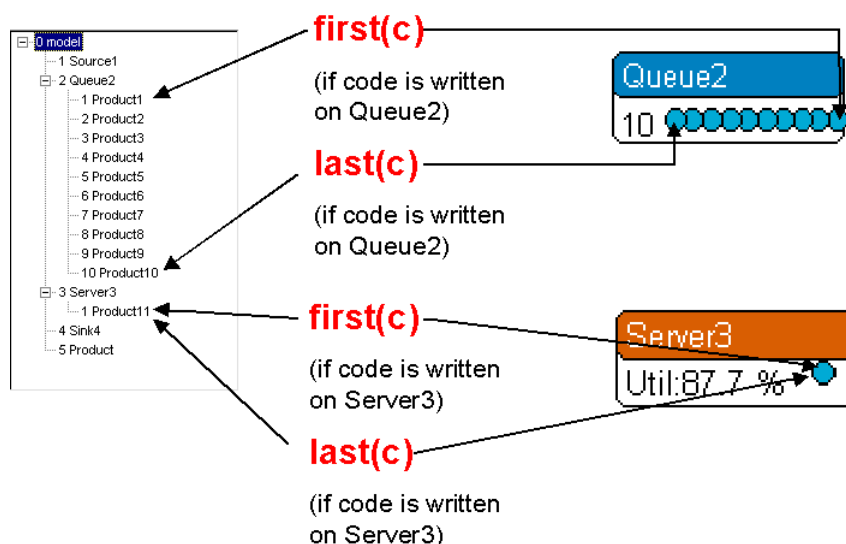


Figure 2

On the left of figure 2 you see the model tree of a simple queuing system, frozen at one moment in time:



Figure 3

On the same level we find the Source, the Queue, the Server and the Sink and the Product-atom placed before the Source (number 5 in the model tree, because it is mostly added after you reset your model the first after making the lay-out).

Queue2 contains 10 products, named Product1 to Product 10. They form a second level, compared to the Product, Source, Queue, Server and Sink. Of course Product1 and, lets say, Product5 are on the same level. Server3 contains 1 product, named Product11.

The highest level is the model itself. In figure 1 you can see this in the model tree! Can you predict how the model tree changes if you reset this model?

If you understand the hierarchy in this model, you will understand referencing...

First a few commands which are often used for referencing:

- first(e1)** refers to the first atom inside atom e1
- last(e1)** refers to the last atom inside atom e1
- next(e1)** refers to the next atom on the same level as atom e1
- prev(e1)** refers to the atom in front on the same level atom e1

So first(c) refers to the first atom inside the current. If the current is a Queue it could be the first product in the queue. Now look at figure 1 again and the examples on *first* and *last*. So, *first* and *last* are always looked at one level deeper from the atom where the statement is written on!

The commands next and previous are explained below:

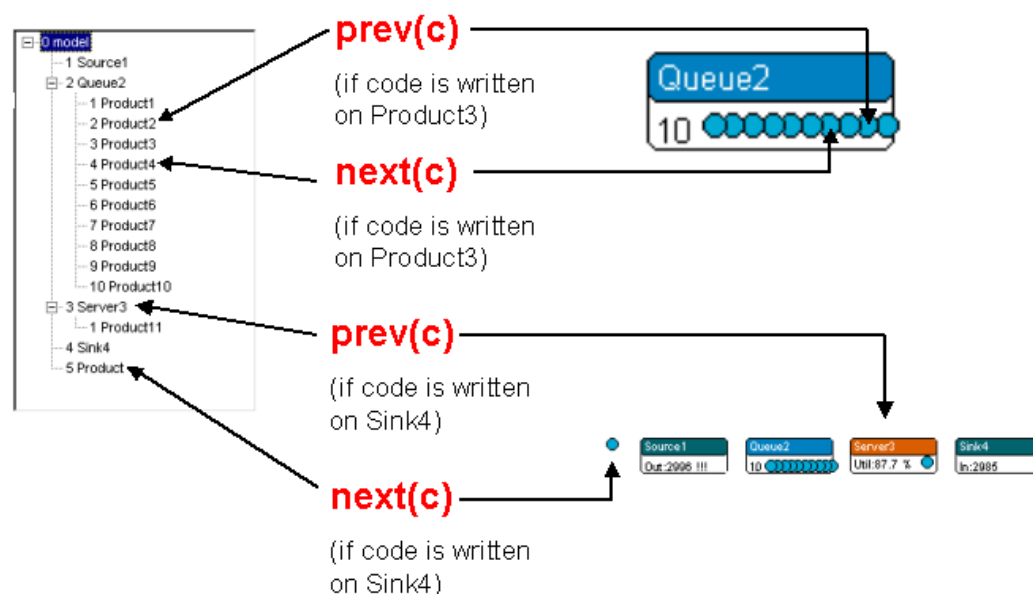


Figure 4

Now look at figure 3 and the examples on *prev* and *next*.

So, *prev* and *next* are always looked at the same level from the atom where the statement is written on!

There are a few commands often used for flowcontrol and related to channels:

**out(e1,e2)** refers to the atom connected to output channel e1 of atom e2

**in(e1,e2)** refers to the atom connected to input channel e1 of atom e2

**openoutput(e1)** opens the general output of atom e1.

**closeoutput(e1)** closes the general output of atom e1. If closed, the channels cannot be ready, regardless of individual channel settings.

In a similar way you can define **openinput** en **closeinput**.

Take a look at figure 5 regarding the use of *in* and *out*:

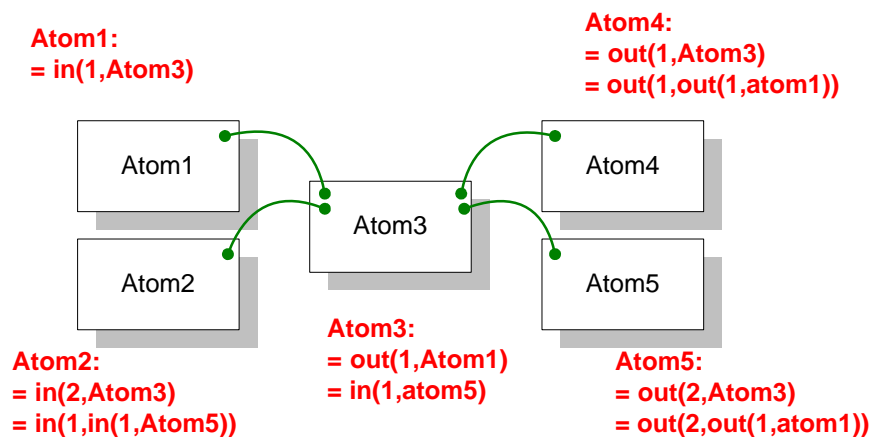


Figure 5

Example 1:

So refers *in(1,c)* (written on Atom3, the point of view) to Atom1, but the same statement written on Atom4 refers to Atom3!

Example 2:

*content(in(3,c))* returns the content (number of atoms) contained in the atom which is connected to input channel 3 of the current atom.

Example 3:

*closeoutput(in(2,c))* closes the output of the atom connected to the second inputchannel of the current

Notice the way commands are nested...

### 3. Important commands

In every programming language there are important commands, which perform as some kind of building blocks.

1. For conditional statements: **if**

Syntax: `if(e1,e2 {,e3})`

Executes e2 if e1 is true (1); otherwise executes e3 (if specified). Returns result of e2 or e3.

Example 1:

`if( time>3600, msg[more than an hour], msg[less than an hour])`

Translated: if time>3600 seconds then give message `more than an hour' else give message `less than an hour'.

Example 2:

`if(content(c)>10, closeinput(in(1,c))`

Translated: if the content of the current atom is more than 10 then stop the input of the atom connected with the first incoming channel else do nothing

Don't worry about the used commands like msg, content or closeinput. We'll explain them later!

2. For executing more than one statement: **do**

Syntax: `do(e1,e2,...,e25)`

Executes e1, e2, etc. in order of sequence. Returns result of last expression. You can use up to 25 parameters. If more parameters are needed, several do loops can be nested.

Example:

`do( set(color(i),colorred),  
set(icon(i),2),  
setlabel([temperature],uniform[20,40],i)`

Three things are done at the involved atom (mostly a product): the color is set on red, the icon is changed to iconnumber two and a label with the name *temperature* is stamped on the product, it's value is chosen randomly between 20 and 40.

3. Labels

The use of labels is the way to store local (temporary) variables by users. They are mostly attached to products and can represent a weight, a customernumber, a productiontime etcetera. They are defined with the command *setlabel* and reproduced with *label*

a. **setlabel(e1,e2,e3)**

Defines a label on atom e3 where e1 is the name and e2 is the new content of the label. Labels do not have to be created, at the moment they are referenced they exist. The *sddb* command does the same.

Example 1:

`setlabel([Weight],10.24,i)`

First select an atom and create (and give a value to) a label with the command `setlabel`. Suppose the name of the label is `v1` and the value is 100:

Example 2:

```
setlabel([v1],100,animatom)
```

`label([v1],animatom)` now returns the specified value (100) of the selected atom.

#### **b. label(e1,e2,{e3})**

Returns the contents of label named `e1` defined on atom `e2`. Labels do not have to be created, at the moment they are referenced they exist. The label name `e1` is always a string and is case sensitive.

The result is a string or a value, depending on the contents and on `e3`: if `e3` is not specified or `e3=0` then the result is a value if the contents can be converted to a value, otherwise the result is a string. If `e3=1`, the result is always a value. If the contents cannot be converted to a value the value is 0. If `e3=2`, the result is always a string. If you use long names and many labels you loose speed.

To avoid mistakes: only use 'small' letters, because labels are CASE-SENSITIVE!

4. We conclude with a few other important commands:

#### **age**

syntax:        `age(e1)`

Returns the age of atom `e1`. The age is the difference between current time and creation time. When a run is reset, the creation time of all atoms becomes 0.

See also **maxage**, **minage**, **avgage**

#### **content**

syntax:        `content(e1)`

Returns the contents (the number of atoms contained at the highest level) of atom `e1`.

See also **maxcontent**, **avgcontent**

#### **avgstay**

Syntax:        `avgstay(e1)`

Returns the average staying time of atoms in atom `e1` over the total runtime.

#### **input**

Syntax:        `input(e1)`

Returns the input of atom `e1`. The input is defined as the number of atoms that have entered atom `e1` until now. The opposite is the output of the atom.

See also **output**