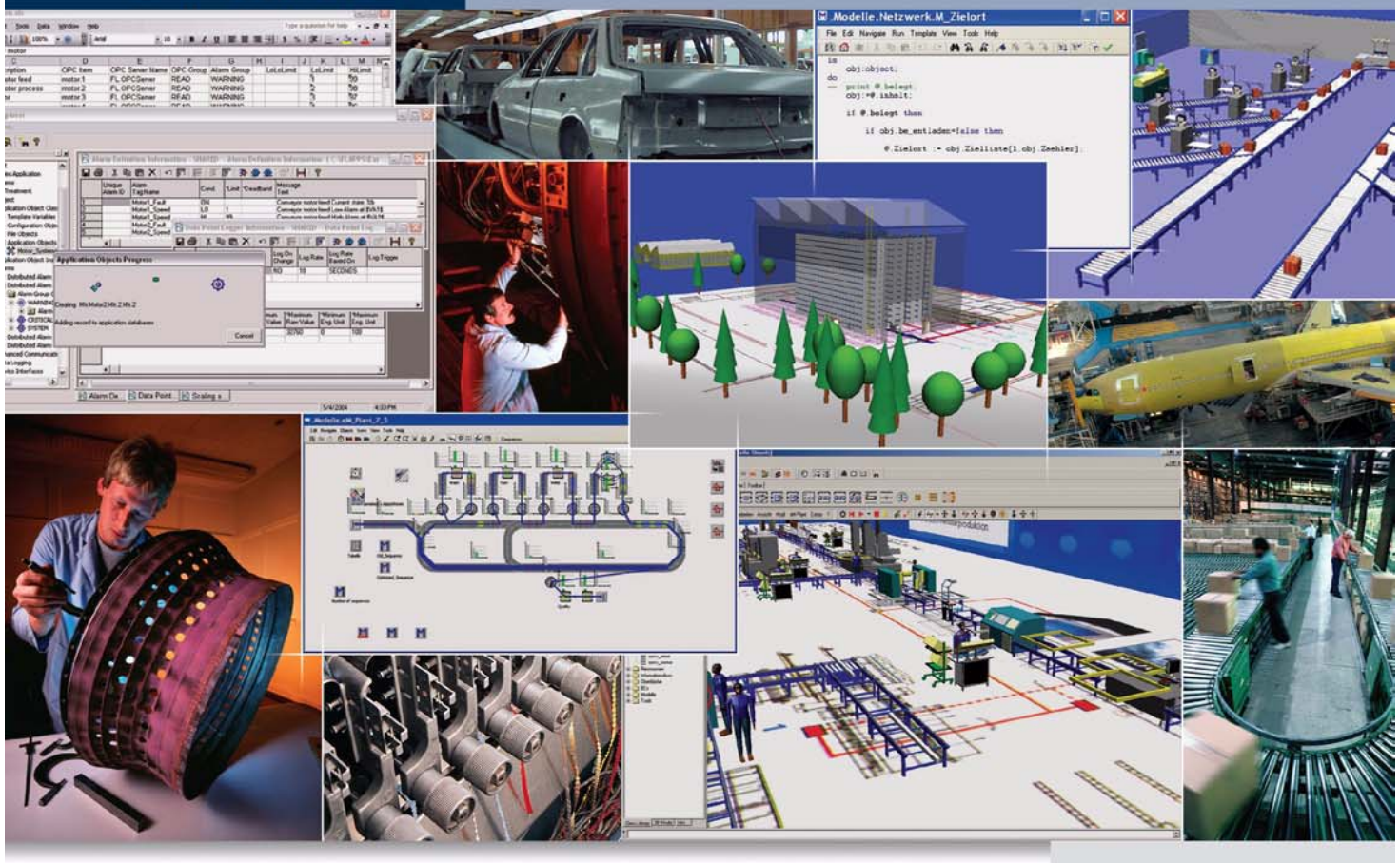
Plant Simulation Assembly library

Reference manual

Siemens PLM Software

www.siemens.com/plm



TECNOMATIX

SIEMENS

Proprietary and Restricted Rights Notice

© 2008 Siemens Product Lifecycle Management Software II (DE) GmbH. All Rights Reserved.

This documentation is proprietary to Siemens Product Lifecycle Management Software II (DE) GmbH.

This document contains proprietary information and is protected by copyright. No part of this document may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form or by any means, without the prior explicit written consent of Siemens Product Lifecycle Management Software II (DE) GmbH.

Siemens and the Siemens logo are registered trademarks of Siemens AG.

Tecnomatix and the Tecnomatix logo are registered trademarks of Siemens Product Lifecycle Management Software Inc.

All other product names or brand names are trademarks or registered trademarks of their respective owners.

Information in this document is subject to change without notice.

TECNOMATIX
Plant Simulation

SIEMENS

Plant Simulation

Assembly Library

Version 9.0

December 2008

Table of Contents

Introduction **1**

Material Flow Objects **7**

SourceN (Source Normal) 7

Sink 12

Usage-oriented Source (SourceUse) 15

SingleprocAttr 17

DistributorType 18

OverflowBuffer 19

Assembly 21

AssemblyType 22

Dismantle 25

Blackbox 28

AssemblyAttr 29

Controls **37**

Sequence 37

SectionCtrl 40

ProtectiveCircuit 41

Cycle 41

AssemblyLine 43

Moving Objects **51**

Entity 51

Carbody/Dolly 51

Skid 51

Rework **53**

RetouchArea 53

SingleprocRetouch 58

Def_Retouch 60

Skids **63**

Load 63

Unload 65

Transfer 68

Tools **71**

RandMgr 71

Distr 72
DistrReference 73

Evaluations 75

Statistics (Stats) 75
PartsStats 79
StateChart 84
Histogram 85
HistogramArea 89
NumMUPlotter 92
TpHPlotter 93
Checkpoint 95
SequenceAnalyzer 98

Just in Time (JIT) 103

SourceJIT 103
Just in Time Control (JIT) 103

Introduction

Plant Simulation Assembly is a library for modeling, simulating and evaluating assembly areas.

In version 9 of Plant Simulation the library was divided into several libraries which can be managed by the library manager. Each of these libraries can be used separately in your simulation model and can be added at every time. Also the documentation was divided into several separate documentations according to the library objects.

The assembly library consists of the following libraries.

- *Assembly*
- *Personnel*
- *Finite State Machine*
- *Kanban*
- *Transport*
- *Assembly Line*

All of these libraries are integrated into the Assembly License and can be used together or in any combination of the libraries.

The Assembly library

This library contains all objects for

- Material flow
- Controls
- Rework
- Skids
- Evaluation
- Random numbers
- JIT

These objects will be described in more detail in the following chapters.

The Personnel library

The following objects are part of the library

ServicePool

ServiceStation

Use these objects to simulate staff members. This librarty has its own documentation.

The Finite State Machine

The following objects are in this library

- FSM Control
- State

- Signal_Connector
- Actor Control
- Sensor Control

Use these objects to create state machines which will react on events of the material flow and will chenge its state according to the event signal. This library has its own documentation.

The Transport library

The following objects are part of the Transport library

- Logistic
- Shelf
- Transporter

Use these objects for modelling inbound logistics. The transport of the parts can be simulated just by increasing and decreasing the numbers. It is also possible to simulate the transport of the parts to get information about the utilization. This library has its own documentation.

The Kanban library

In earlier versions this library was a part of the assembly library. The objects of this library are now replaced by the objects of the Kanban library in the core system. The objects are described in the Plant Simulation reference manual.

The Assembly Line library

The followng objects are part of the assembly line library

AssyWork

AssyTrack

AssyLIneChart

AssemblyExplorer

ProductionplanManager

Using these objects enables you to simulate assembly processes where the parts are moving along a track. This library has its own documentation.

Starting Plant Simulation Assembly

First start Plant Simulation. Use the library manager to load the libraries you will need for creating your simulation model. It is possible to add other libraries to your class library at any point in time using the library manager.

Updating Existing Models

Simulation models using older versions of the assembly library can be updated using the library manager. Select the library you want to update and select the version you want to use. See also the Plant Simulation Reference Manual for this.

Where to Get Help

We do not describe the built -in objects, such as the *AttributeExplorer* and the *ExperimentManager*, in this manual. You find information about them in the Plant Simulation Reference Manual or in the Plant Simulation Step-by-Step Help. The libraries *Personnel* and *FSM* have their own documentation.

How to Get in Touch with Us

You will always find the current version of the assembly library on our customer page on the Plant Simulation Homepage:

www.emplant.de/support/plant/

Select Customer Support and enter your password. Select AOLs and then select assembly. You always find the most current version at this location. If you have a maintenance contract, you can download it from here.

The Library Modules

MaterialFlow

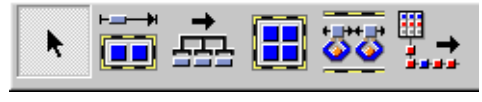
You will graphically and interactively create the production structure with the material flow objects of the application object library Plant Simulation Assembly. When modeling you can use the built-in Plant Simulation objects as well as the objects from this folders. You can also connect these material flow objects with *Workplaces* and with *Workers*. Use the *ShiftCalendar* to define the availability of workers and machines.

This folder contains these objects:

- *Processing* and *Assembly* according to attributes, *Blackbox*
- *SourceNormal*, *usage-oriented Source*
- *Sink*
- *Assembly* and *Dismantle* according to attributes
- *Assembly* on an assembly line with corresponding evaluations
- *Distributor* according to attributes

We will describe the objects in detail in the chapters below.

Controls



You can use the object *Sequence* to define variants of the parts to be produced, while the objects *Cycle* and *ProtectiveCircuit* influence the course of production.

- *SectionCtrl*
- *Cycle* and *ProtectiveCircuit*
- AssemblyLine
- *Sequence*

Rework

You can use the objects from this folder to model quality control and rework stations. It contains these objects:

- Objects for modeling rework stations
- Quality control or discharge station respectively

- Definition of error symptoms and frequencies

Skids

This module contains all objects for loading and unloading loading equipment, such as skids or AGVS. The library contains these objects:

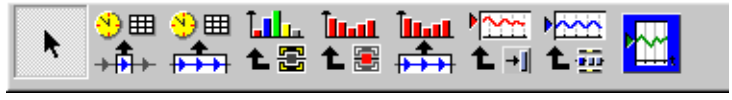
- *Load* for loading any type of loading equipment.
- *Unload* for unloading any type of loading equipment.
- *Transfer* for transferring products from one transport circle to another transport circle.

Just in Time (JIT)

You can control the creation of parts depending on the selected stations with the *Just in Time* module. You can do this with the objects:

- *SourceJIT* for creating the parts controlled by the object *JIT*.
- *JIT* for controlling the parts to be created.

Evaluation



You can show the results of the simulation runs, such as the utilization of resources, the fill levels of buffers, etc. graphically in state charts and histograms. You can also write the results to a table.

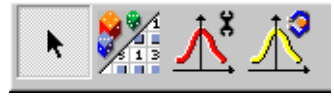
You can do this with the objects:

- *Stats*
- *Checkpoint*
- *Parts statistics*
- *Histogram*, *StateChart*
- *TpHPlotter*
- *SinkEvaluation*

PartsAndOrders/MyObjects

The folder PartsAndOrders contains pre-defined MUs, which you can either use as we defined them or use them as a template for your own, project-specific MUs. You can place these project-specific MUs into the folder MyObjects.

RandomNumbers



For setting-up your models and for executing simulation runs, we provide these objects:

- Discrete empirical distribution
- Managing Random Number Streams (*RandMgr*)

1

Material Flow Objects

When using these objects, keep in mind that you cannot change the structure of the model or individual parameters of the objects while your model is running. You have to restart the simulation run each time you make a change.

Many of the material flow objects are modeled in a *Frame*. The processing stations within these *Frames* always have the name workplace.

All material flow objects offer the built-in distributions which Plant Simulation provides. The Plant Simulation Help and What's This help describe how to enter the parameters of the individual distributions.

SourceN (Source Normal)

This type of *Source* can create parts in different ways:

- It produces parts of a single type as long as it is not blocked.
- It produces the number of parts, such as *Skids* or *Containers*, of a single type. After it finished producing the amount of *Skids*, the *SourceN* switches to passive mode.
- It produces parts of a sequence type as long as it is not blocked. The type is determined by a sequence element. The sequence element itself is determined by the associated *Sequence* object. The *SourceN* is not intended to model failures and pauses. We recommend to handle these events in a station which is located after the *SourceN* in the sequence of stations.
- It produces parts of a VariantGenerator as long as it is not blocked.

Name, Label

These text boxes contain the name or the label of the object respectively.

Source Type

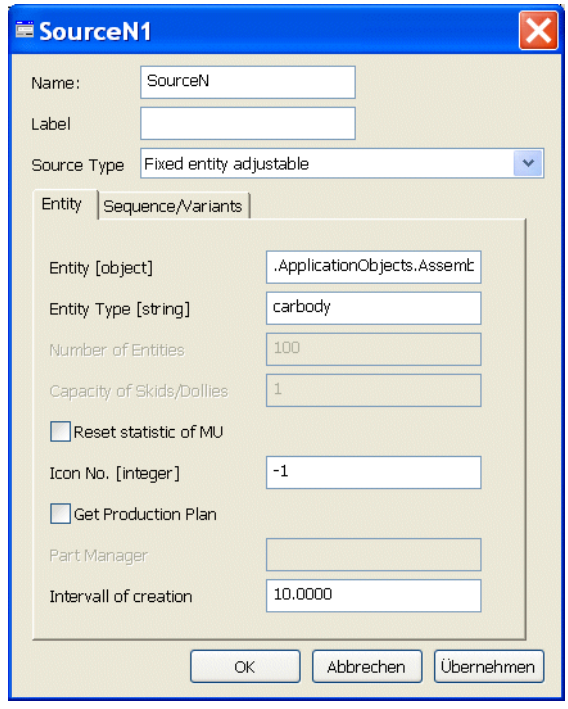
Select how the *SourceN* produces parts from this drop-down list. Depending on what you select, Plant Simulation Assembly activates or deactivates the associated dialog items.

You can select:

- Fixed Entity adjustable: The *SourceN* produces constantly a certain part type
- Number adjustable: The Source produces the number of parts you specify. Once the parts are produced, the *SourceN* stops producing parts.
- Entity is defined by Sequence: A *Sequence* object defines the parts to be produced.
- Entity is defined by VariantGenerator: A VariantGenerator defines the parts to be produced.

Below we will describe the parameters for the different settings.

Setting “Fixed Entity adjustable”



Entity

Enter the MU class of which you would like to create instances. You can also drag an MU onto the object *SourceN* and drop it there. In general you should only use MUs from the folder *PartsAndOrders* as we already defined all required attributes for the MUs stored there.

Part type

Enter the part type of the part into this text box. The part type *Entitytype* is the MU attribute which is, for example, accessed during a type-dependent *assembly*. This part type is also use as the name of the produced MU.

Reset statistic of MU

If this checkbox will be set, the statistic values of the MU will be reset as soon as the MU leaves the *SourceN* object. This will have effect on the lifetime of the MU.

Note: This will result in modified values for the value of statAvgLifetime of the MU.

Icon No

The produced MU can have several icons. Enter the number of the icon to be displayed after the *SourceN* created the MU. When you enter -1, the MU uses the default icon.

Get Production Plan

Use this check box to set if the production plan manager, the *Part Manager*, is to read the production plan for the produced parts. The production plan which is attached to the part as a customized attributes is only used by the objects *AssemblyLine*, (*Assy_Work*) and *AssyAttr*. You will define the format of the production plan in the *Part Manager*.

Part Manager

Enter the path to the associated *PartManager* into this text box. The *PartManager* assigns the produced parts their production plan, provided you selected the check box Get Production Plan.

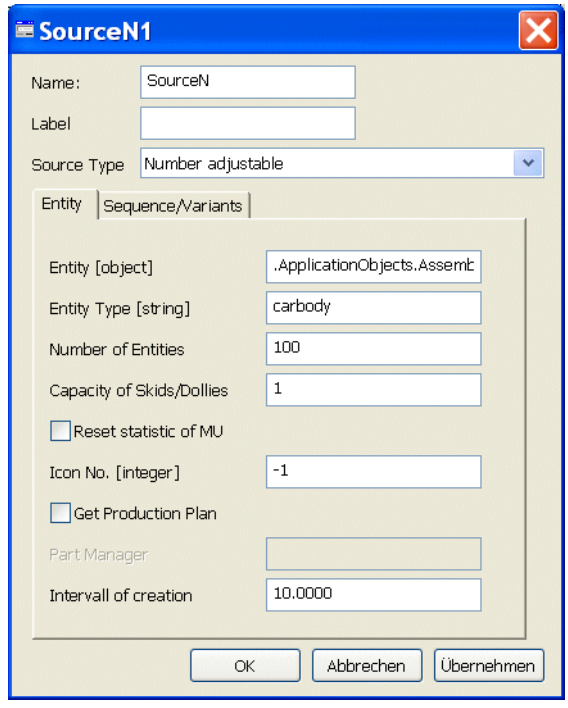
Note: This entry is only used in connection with the production line and not anywhere else.

Interval of creation

Enter the interval for creating the MUs to be produced into this text box. The *SourceN* attempts to move MUs to the succeeding stations in this time interval. The *SourceN* can only keep this interval when it is not blocked by the succeeding station.

Setting “Number adjustable”

You will, as a rule, use this setting to create a certain number of *Skids*.



Entity

Enter the MU class of which you would like to create instances.

Part type

Enter the part type of the part into this text box. The part type *Entitytype* is the MU attribute which is, for example, accessed during a type-dependent *assembly*.

Number of Entities

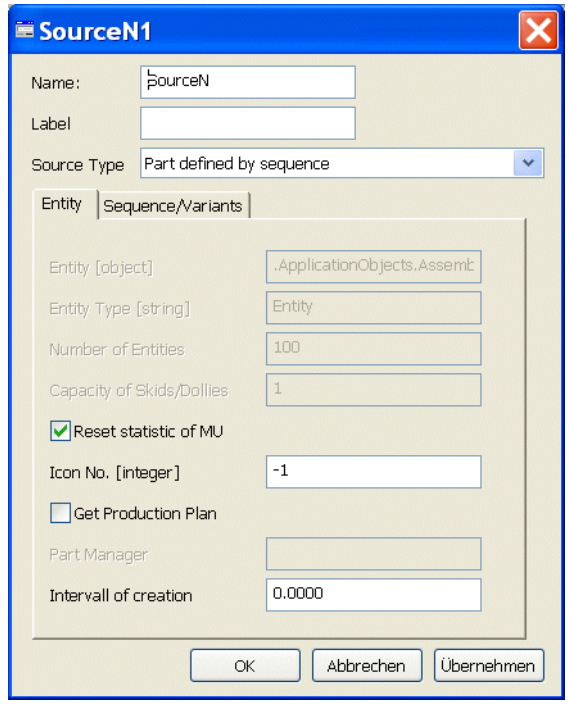
Enter the number of the parts to be produced into this text box. After the parts have been produced, the *SourceN* becomes passive.

Capacity of Skids/Dollies

Enter the capacity of the *Skids* here. If you did not define a *Skid* or a *Transporter* as an MU class in the text box Entity, this value will not be used.

Setting "Part defined by Sequence"

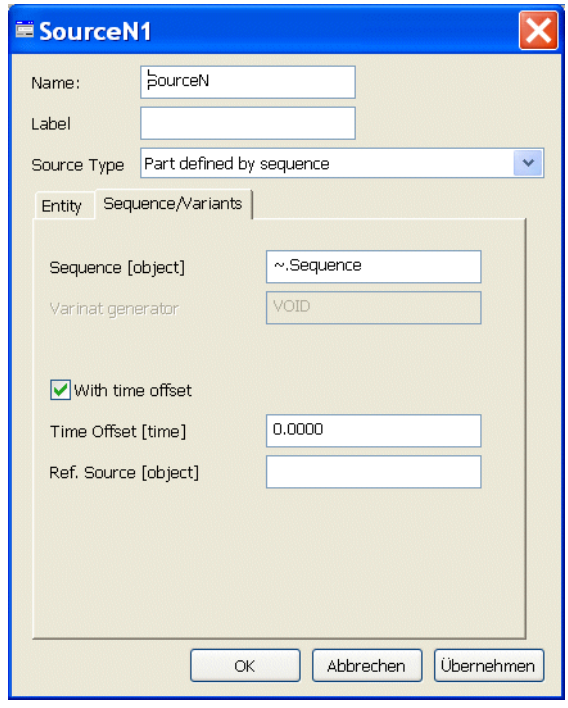
When you select this setting, the *SourceN* will be connected with an object of type *Sequence*, which provides it with a sequence of different parts, which you can define.



Sequence [object]

Enter the *Sequence* object into this text box which controls the *SourceN*.

You can either enter the name of the *Sequence* object into the text box or you can drag it over the icon of the *SourceN* and drop it there.



With Time Offset

Use this check box to synchronize the points in time at which two sequential objects of type *SourceN* produce parts. If, for example, you want the pre-assembly line to start working 2 hours before the main assembly line, you will enter the main assembly *SourceN* as the Reference Source into the pre-assembly *SourceN* and enter an Offset of -2:00:00.

Time-Offset [time]

Enter the time offset as compared to the *Reference Source* into this text box. Positive values mean, that the *SourceN* produces parts at a later point in time than the *Reference Source*.

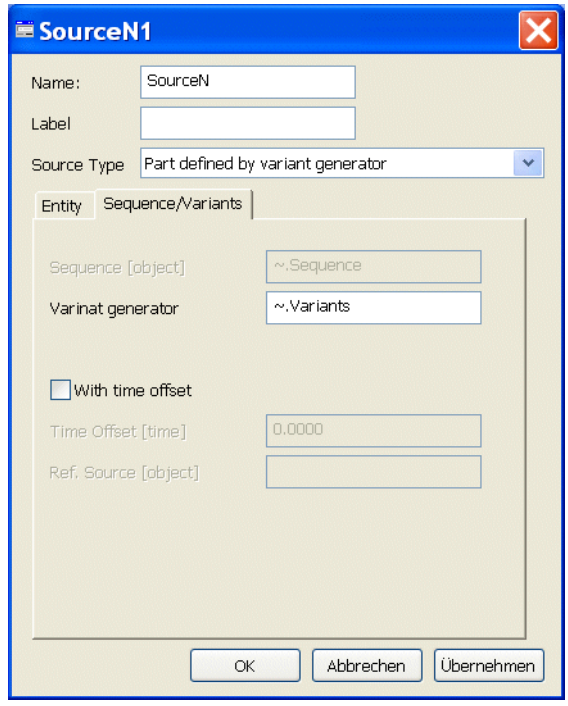
Ref. Source [object]

Enter the *Source* object from which the *SourceN* gets the points in time for creating MUs into this text box.

You can either enter the name of the Reference Source object into the text box or you can drag it over the icon of the *SourceN* and drop it there.

Setting “Part defined by variant generator”

When you select this setting, the *SourceN* will be connected with an object of type *VariantGenerator*, which provides it with a sequence of different parts, which you can define.



Variant generator

Enter the *VariantGenerator* object into this text box which controls the *SourceN*.

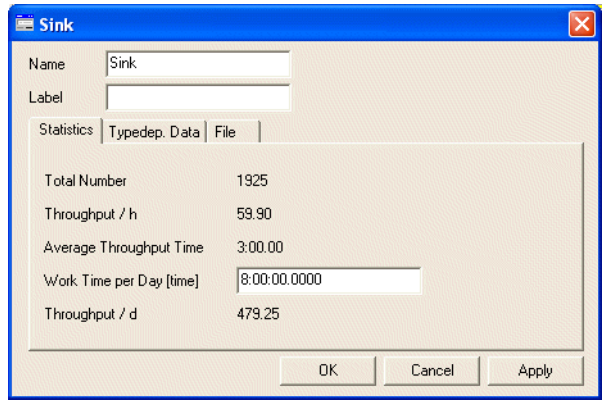
You can either enter the name of the *VariantGenerator* object into the text box or you can drag it over the icon of the *SourceN* and drop it there.

Attributes of the Object

objType: SourceN

Sink

This object immediately destroys incoming parts. The *Sink* calculates the statistics data using the arrival times and the part type. The parts are destroyed immediately, without any processing time. The *Sink* is not intended for modeling failures or pauses. These should be handled in a station before the sink.



Name, Label

These text boxes contain the name and the label respectively of the object.

Tab Statistics

Total number

Is the number of parts which arrived at the *Sink*.

Throughput/h

Mean throughput of parts per hour which arrived at the *Sink*.

Average Throughput Time

Average throughput time of arriving parts over all part types.

Work Time per Day

Enter the working time per day into this text box. This value is only used for calculating the value Throughput/d and does not otherwise influence the simulation model.

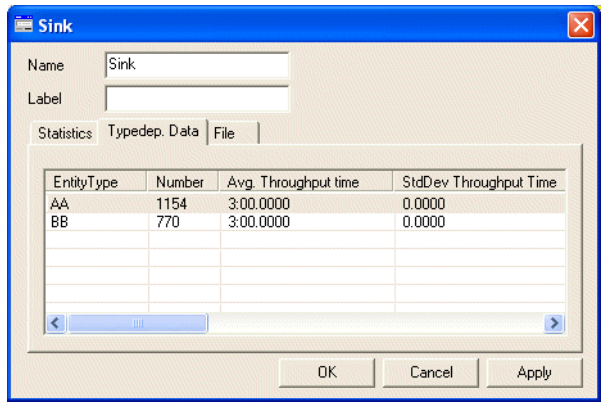
Throughput/d

Throughput of parts per work day. For calculating it the program uses the time you entered for Work Time per Day. The throughput per hour is multiplied with the Working time per day.

Note: All displayed values are reset at a statistics start.

When you click Apply, the program updates the values in the dialog window.

Tab Type-dependent Data



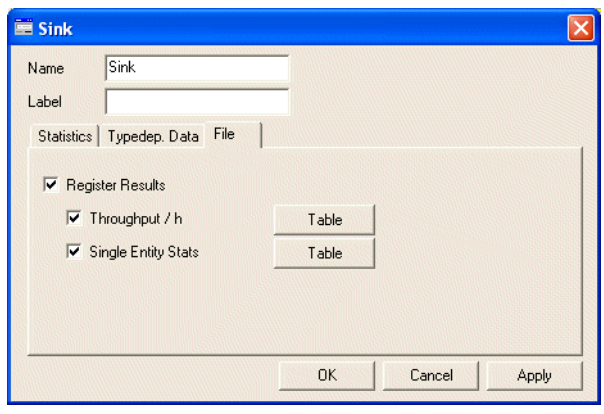
This tab shows the statistics values for each part type. The overview provides these values:

- EntityType: Type, i.e. name of the part.
- Number: The amount of parts of this type.
- Avg. Throughput Time: Average throughput time of parts of this type.
- StdDev Throughput Time: Standard deviation of the throughput time.
- min Throughput Time: Minimal value of the throughput time.
- max Throughput Time: Maximal value of the throughput time.
- Avg throughput / h: Mean throughput per hour.
- StdDev Throughput / h: Standard deviation of the mean throughput.
- min Throughput / h: Minimal value of the throughput.
- max Throughput / h: Maximal value of the throughput.

Tab File

On this tab you can select to save the individual values in tables or not.

Be aware that these tables might be of considerable size.



Register Results

Use this check box to set if you want to save the results or if you do not want to save them.

Throughput/h

Select this check box to set if you want to save the throughput for each individual hour in a table or not.

Table: Throughput/h

This table records the throughput per hour of parts.

Single Entity Stats

If you want to protocol the times and type data of the individual parts in a table, select the check box Single Entity Stats.

Note: Be aware that recording the part statistics requires large amounts of RAM.

Table: Single Entity Stats

The table shows this data of the destroyed parts:

- Part type
- Starting point in time of the part
- Finishing point in time of the part

Attributes of the Object

objType: Sink

typestat: This table contains all type dependent statistical values.

	string 1	integer 2	time 3	time 4	time 5	time 6	real 7
string	Entity Type	Number	Avg. Throughput time	Std. dev. throughput time	Min. throughput time	Max. throughput time	Avg. throughput per hour
1	carbody	100	20.0000	0.0000	20.0000	20.0000	0
2							

	real 7	real 8	real 9	real 10
string	Avg. throughput per hour	Std. dev. throughput per h	Min. throughput per hour	Max. throughput per hour
1	0	0	0	0
2				

Table TypeStat

The meaning of the values was defined at Tab Type-dependent Data, described on page 14

Methods of the Object

Usage:

```
<path>.statistic(<table>);
<path>.statistic(<table_path>);
```

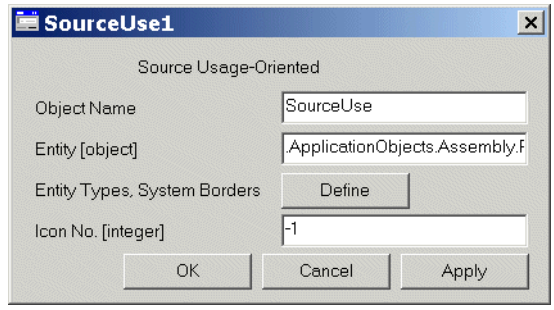
This method returns the values from the tab TypeStat and writes them into a table. You can pass the name of a table or an object of type table. Plant Simulation automatically formats the table.

We described the contents of the individual columns above.

Usage-oriented Source (SourceUse)

The object *SourceUse*, which is usage-oriented, creates parts whose type depends on the occupation of the succeeding line. The succeeding lines are located between the *Source* and the respective system border, which you enter into column border

in the table for defining the part types and the system borders. The part type, which you entered into the column Type will be used as the identifier for the MU. This part will then be transferred to the respective succeeding line.



Object Name

This text box shows the name of the object.

Entity

The Source creates the object from the object library which you enter here.

Entity Types, System Borders

Enter the succeeding lines into the column successor into this table. The program automatically enters all objects connected with the *SourceUse* into this column.

	object 1	string 2	object 3	integer 4
string	successor	Type	border	no
1	~.s1	A	~.S5	0
2	~.S3	B	~.B2	0
3				

For each succeeding line which starts with the object, which you entered into the column successor, you have to enter the part type into the column type. Enter the respective system border for the succeeding lines into the column border. If you do not enter a succeeding line, the program uses the direct successor of the *SourceUse* as system border. The column no is used for internal purposes.

Take care to enter the respective object *workplace* when you enter the stations. When you enter a relative path, remember that the path starts with *location*, as the entry starts within the object *SourceUse*.

Icon No

Enter the number of the icon to be displayed after the Source created the MU. When you enter -1, the MU uses the default icon.

Course of Action

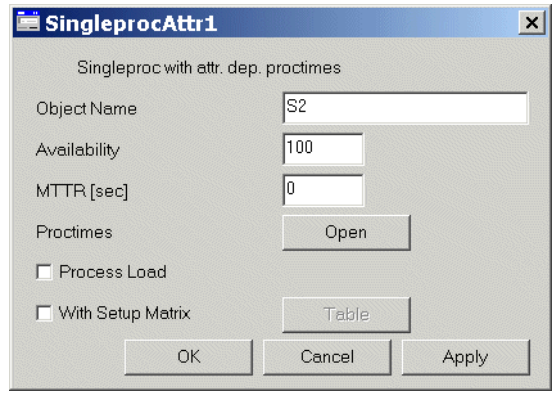
The source *SourceUse* alternatively creates parts for the defined lines. When doing so, the program makes sure that the same number of parts are located on each line. Provided a part exits the line at the system border, the SourceUse creates this part anew.

Attributes of the Object

objType: SourceUse

SingleprocAttr

The object *SingleprocAttr* processes parts depending on the attributes of these parts. You can define which attributes of the parts on the *SingleprocAttr* are accessed. In addition, you can also define a set-up matrix.



Object Name

This text box shows the name of the object.

Availability

Enter the availability of the station into this text box.

MTTR

Enter the mean time to repair into this text box.

Processing times (Proctimes)

You define the attribute-dependent processing times of the parts in a table.

	string 1	string 2	string 4
string	proctime distribut	proctime parameter	entityType
1		75	AA
2	normal	1:20.0000, 20.0000, 20.0000, 3:00.0000	BB
3	const	80	CC
4			

Here you define the processing times depending on the attributes of the parts. Enter the distribution function you want to use for the processing time into the column proctime distribution. Define the associated parameters in the column proctime parameters. The parameters of the individual distribution functions are described in the Plant Simulation Help. The column EntityType defines the part type for which this setting applies. You can also enter all as the `EntityType`. This way you can create a default setting and you only have to define the exceptions from this default. In addition to the part type you can define any number of additional attributes.

Note: When you do not enter anything into the cells in the first column (proctime distribution), the processing time is constant.

When entering the parameters for the distribution, you do not have to enter the random number stream.

Process Load

Use this check box to set if the part which is directly located on the *SingleprocAttr* will be processed or if the parts which are located on a workpiece carrier, which is located on the *SingleprocAttr*, will be processed.

The processing time is determined according to what you select here. If several parts are located on the workpiece carrier, the program calculates the sum of the individual processing times.

With Set-up Matrix

Use this check box to set, if the set-up time will be defined in a matrix. In this matrix you define the times for setting up from one part type to processing the succeeding part types.

	string 0	time 1	time 2	ti 3
string	old / new typ	limo	coupe	
1	limo	0.0000	2:30.0000	
2	coupe	1:40.0000	0.0000	
3				
4				

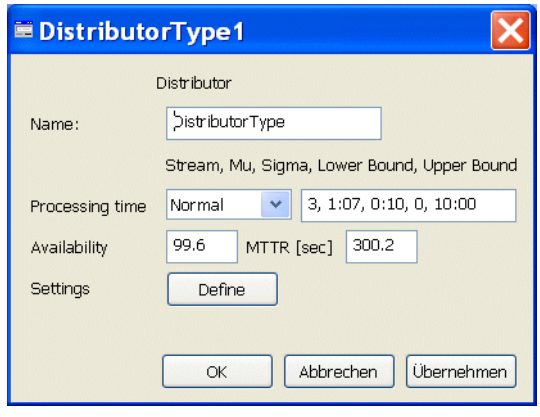
Note: The station will not set up for the first part, which the station processes, as no preceding part can be determined from the set-up matrix.

Attributes of the Object

objType: SingleprocAttr

DistributorType

This object distributes the incoming MUs according to the settings you selected for the attributes to its successors.



Name:

This text box shows the name of the object.

Processing Time

Select one of the distributions from the drop-down list and enter the parameters which this distribution requires. Plant Simulation shows which parameters the distributions require above the text box.

Settings

The successors of the distribution object will automatically be entered into the column successor, provided the object is already connected with the succeeding stations. Press button *Define* to open the table of the successors.

successor	EntityType
~.Sink1.Workplace	Teil_A
~.Sink2.Workplace	Teil_B
~.Sink3.Workplace	Teil_C

Enter the attributes which the *DistributorType* access into the column indexes 2 to n. Then, enter the values into the cells of the columns, which the part attributes may take. In the example above, we only entered the part attribute EntityType, which means that the *DistributorType* only accesses this attribute. In this case additional part attributes are not relevant for transferring the MUs.

If different part types are to be transferred to the same successor, you have to enter the successor into the column successor the respective number of times.

To define a default behavior enter `all` into the column EntityType. Then, the program first checks if a unique successor is defined for the attribute combinations of the MU. If this is not the case, the program searches for all in a second step to use the default transfer strategy. If it does not find the entry all either, the MU will be transferred according to the last entry in the table to prevent blocking of the model.

Attributes of the Object

objType: DistributorType

Methods of the object

Statistic

Usage:
```
<path>.statistic(<table>)
<path>.statistic(<table_path>)
```

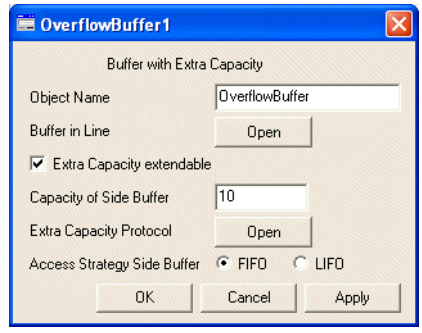
This method returns the values from the tab Statistics and writes them into a table. You can pass the name of a table or an object of type table. Plant Simulation automatically formats the table.

OverflowBuffer

This object is a buffer with an *OverflowBuffer* in parallel. In addition to the *Buffer* proper, which is located within the line, you can set if the *OverflowBuffer* has a fixed size as well as how this side buffer is accessed.

As soon as the *Buffer* within the line is all filled up and the part, which wants to move, cannot be transferred to the successor, this part will be moved to the *OverflowBuffer*.

Parts will be fed back from the overflow buffer into the line, as soon as the successor can accept it and the *Buffer* in line cannot deliver a part.



Object Name

This text box shows the name of the object.

Buffer in Line

You will parameterize the *Buffer*, which is located within the line, directly in the object *workplace*.

Extra Capacity Extendable

If you want the overflow buffer to extend its capacity, select this check box. Then the overflow buffer extends its capacity as soon as it is full.

Capacity of Side Buffer

If the capacity of the side buffer is not extendable, enter a fixed Capacity here. If the capacity of the side buffer is extendable, you can enter the initial capacity here.

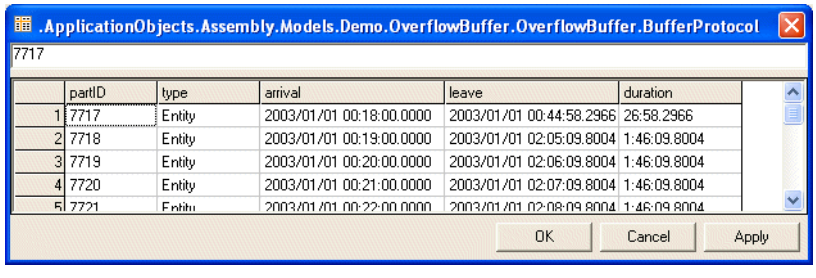
Access Strategy Side Buffer

Here you can set the release from stock behavior of the side buffer:

- First in First out: The part which has been stored the longest in the side buffer is moved first (*Queue*).
- Last in First out: The part which has been transferred to the side buffer last is moved first (*Stack*).

Extra Capacity Protocol

The program enters statistics data for all part which have been in the side buffer into a table.



	partID	type	arrival	leave	duration
1	7717	Entity	2003/01/01 00:18:00.0000	2003/01/01 00:44:58.2966	26:58.2966
2	7718	Entity	2003/01/01 00:19:00.0000	2003/01/01 02:05:09.8004	1:46:09.8004
3	7719	Entity	2003/01/01 00:20:00.0000	2003/01/01 02:06:09.8004	1:46:09.8004
4	7720	Entity	2003/01/01 00:21:00.0000	2003/01/01 02:07:09.8004	1:46:09.8004
5	7721	Entity	2003/01/01 00:22:00.0000	2003/01/01 02:08:09.8004	1:46:09.8004

It fills these columns at run time:

- partID: ID of the part.

- type: Part type.
- arrival: Arrival time of the part in the side buffer.
- leave: Exit time of the part from the side buffer.
- duration: Dwelling time of the part in the side buffer.

Attributes of the Object

objType: OverflowBuffer

Assembly

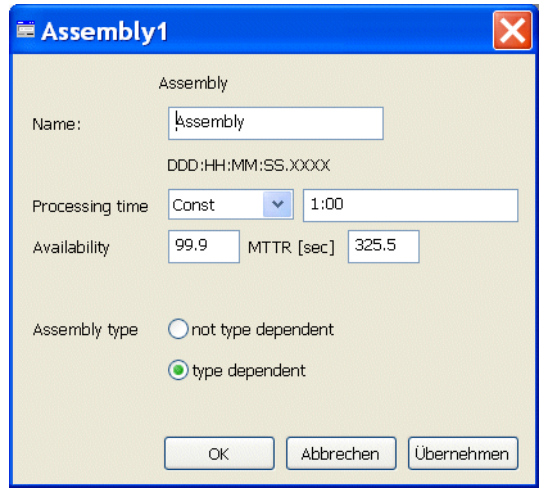
You can use this object to model the assembly of several parts. The assembly can depend on the part type or not depend on it.

The *Assembly* station has a main entrance at which the main part enters the station and several side entrances at which mounting parts enter. The main part will be transferred, the mounting parts will be destroyed. For this reason, the main entrance has to be connected. The station *workplace* represents the *Assembly* station proper, into which you enter the times. As long as the assembly time is running, the entrances are blocked and no new parts may enter the station. The side entrances will only be opened when a main part is located in the station.

When connecting the predecessor with the entrances of the assembly station, make sure that the main part enters through the Main entrance, which is connected with the *workplace*.

Note: It is imperative that the direct predecessors of the *Assembly* station are material flow objects, as the *Assembly* station directly influences the behavior of the objects preceding it. A *FlowControl* object may never be the predecessor of the Assembly station.

The same is true for an *Buffer* not being allowed as a predecessor of the *Assembly* station, as it does not allow accessing its contents. If you need to model a buffer, use the *PlaceBuffer*.



Name:

This text box shows the name of the object.

Processing Time

Select one of the distributions from the drop-down list and enter the parameters which this distribution requires. Plant Simulation shows which parameters the distributions require above the text box.

Availability, MTTR

Enter the availability and the mean time to repair into these text boxes.

Assembly Type

Select the respective radio button to set if the assembly depends on the type of the part or if it does not depend on it.

Not type-dependent

Each side entrance may only be connected with a single side line. As soon as a main part is located on the station one of each of the mounting parts is attached.

Type-dependent

The side lines which provide the different mounting parts are attached to **each** side entrance. When the main part of a certain type enters the station, the side lines only allow parts of a certain type to enter. For this the station checks the custom attribute *EntityType*, it does not check the name of the part.

Note: If one side line only is attached to a side entrance, the *Assembly* station does not run the type check and thus allows each and every part of the sideline to enter.

Main parts entering the assembly station are MUs of type *Limo* and *Coupe*. Both types of main parts each require a left door and a right door as mounting parts. When a part of type *Limo* enters the station, it requests a part of type *Limo* from *Side1*. It also requests a part of type *Limo* from *Side2*. When a part of type *Coupe* enters the station, it requests a part of type *Coupe* from *Side1*. It also requests a part of type *Coupe* from *Side2*.

Extend Entrances

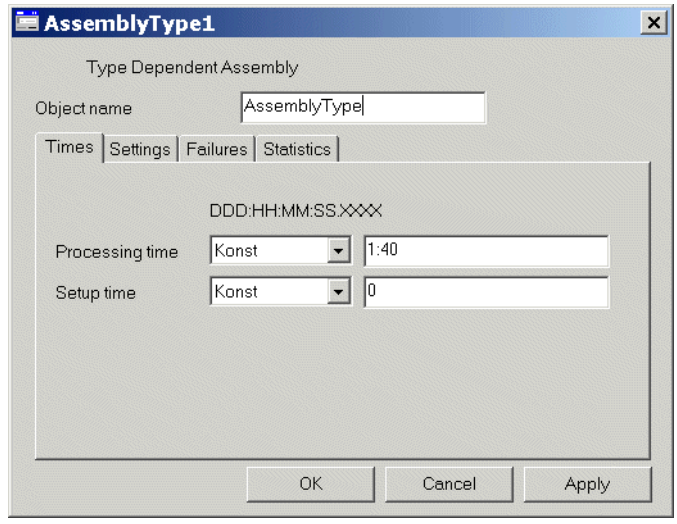
To add more side entrances, insert additional objects of type *SingleProc*. Set the processing time of the new stations to `0`. Now connect the new single processing stations.

Attributes of the Object

objType: Assembly

AssemblyType

The object AssemblyType is an assembly station based on the built-in object Montage. In the AssemblyType you can enter which and how many mounting parts are to be attached to a main part.



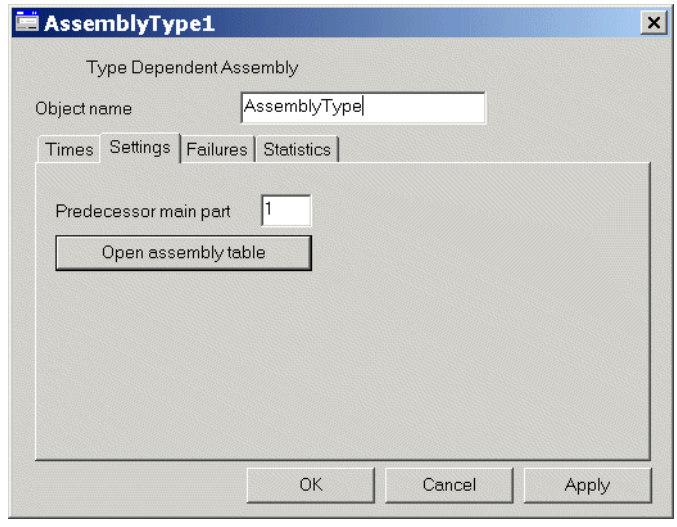
Object Name

This text box shows the name of the object.

Tab Times

Enter the assembly time and the set-up time.

Tab Settings



Predecessor main part

Enter the number of the predecessor which delivers the main part.

Open Assembly Table

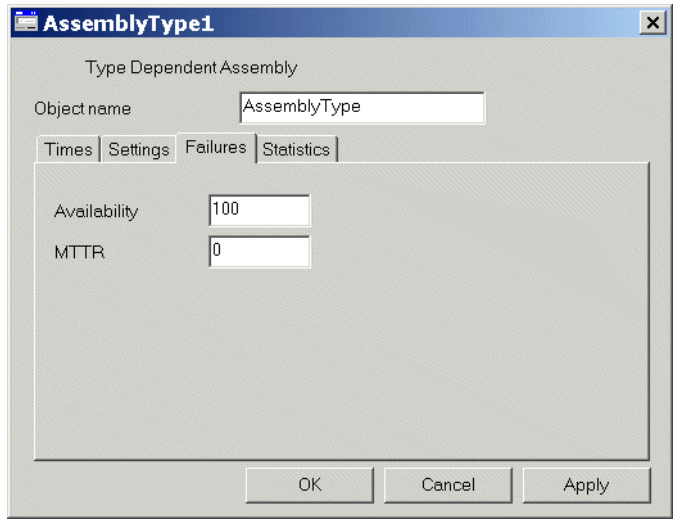
Click this button to open table in which you define the assembly process. Plant Simulation automatically sets the column headers 1..n of this table. Enter the allowed part types of the main part into column 0. Enter the number of parts from the respective predecessor which are required for the main part into the respective cells.

	string 0	integer 1	integer 2	integer 3	integer 4
string	EntityType	SingleProc1	SingleProc2	SingleProc3	
1	A	1	2	1	
2	B		1	2	
3	C	2	1	2	
4					

In our example the assembly of a part of type A required 1 part from predecessor Singleproc1, 2 parts from predecessor Singleproc2 and 1 part from predecessor Singleproc3.

Tab Failures

Define the failures of the object on this tab.



Availability

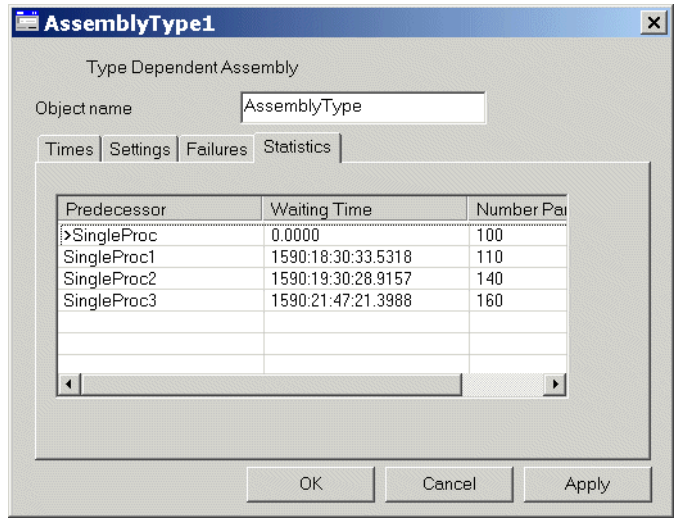
Enter the availability of the object into this text box.

MTTR

Enter the mean time to repair of the object into this text box.

Tab Statistics

This tab shows these values in a table:



Predecessor: is the name of the preceding object.

Waiting Time: is the waiting time for the required mounting parts from the predecessor.

Number Parts: is the number of parts from this predecessor.

Attributes of the Object

proctime: is the assembly time.

setupTime: is the set-up time

objType: AssemblyType

Methods of the Object

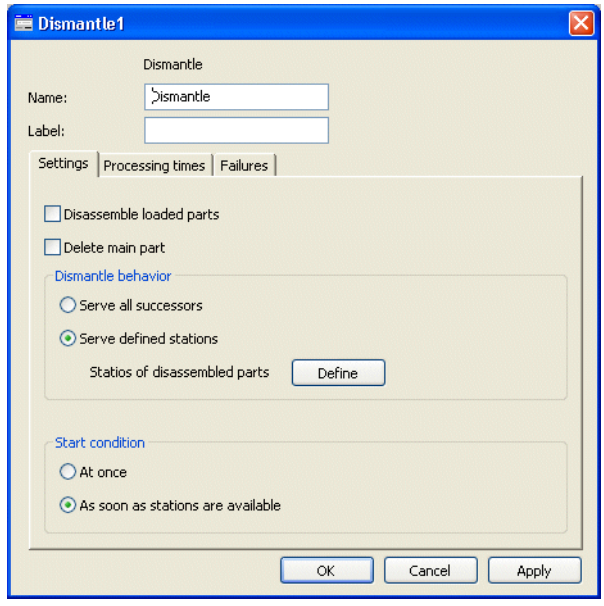
Statistic

Usage:
```
<path>.statistic(<table>)
<path>.statistic(<table_path>)
```

This method returns the values from the tab Statistics and writes them into a table. You can pass the name of a table or an object of type table. Plant Simulation automatically formats the table.

Dismantle

The *Dismantle* station creates several copies from a part and sends one each to an exit. The *Dismantle* station can only process additional parts when all parts have been transferred successfully.



Connect the *Dismantle* station with one predecessor and any number of successors.

Name:

This text box shows the name of the object.

Disassemble Loaded Parts

You can use this check box to set, if the parts are expected on a *Skid*. Accordingly the parts will be dismantled, i.e. the mounting parts will be created and transferred to the successor.

When you clear the check box, the *Dismantle* station expects the parts directly and treats them accordingly.

Dismantle Behavior

A part can be dismantled in two different ways. Both variants transfer the main part to the successor. The parts to be dismantled are created as follows.

Serve Each Successor

Here the program creates a single part on each succeeding material flow object.

Serve Defined Stations

For this setting you enter into a table which and how many parts are created on which material flow object.

	table 1	string 2
string	dismantle plan	EntityType
1	dp	limo
2	dp	coupe
3	dp	pickup

Enter this data into the columns:

- dismantle plan: This column contains the subtable with the dismantle plan.

- Column 2 to n: Enter the names of the attributes into the column indexes and the associated values of the attributes into the cells of the columns. The *Dismantle* station determines the dismantle plan according to the combination of attributes you selected.

Dismantle plan

	object 1	object 2	string 3	integer 4
string	station	entity	type	number
1	~.sp3	.ApplicationObjects.Assembly.PartsAnd	P_1_1	1
2	~.sp4	.ApplicationObjects.Assembly.PartsAnd	P_1_2	1

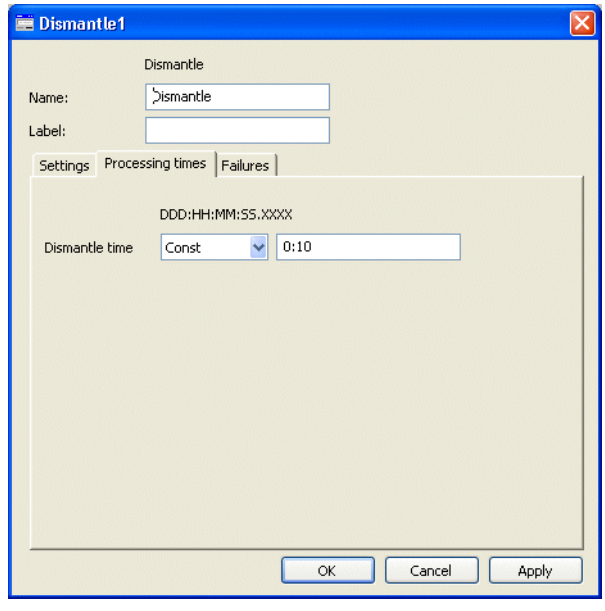
Enter this data into these columns:

- station: Enter the names of the material flow objects on which the MUs are to be created into the cells of this column.
- entity: Enter the MU class of which you would like to create instances.
- type: Enter the associated part type into the cells of this column.
- number: Enter the associated number into the cells of this column.

Start Condition

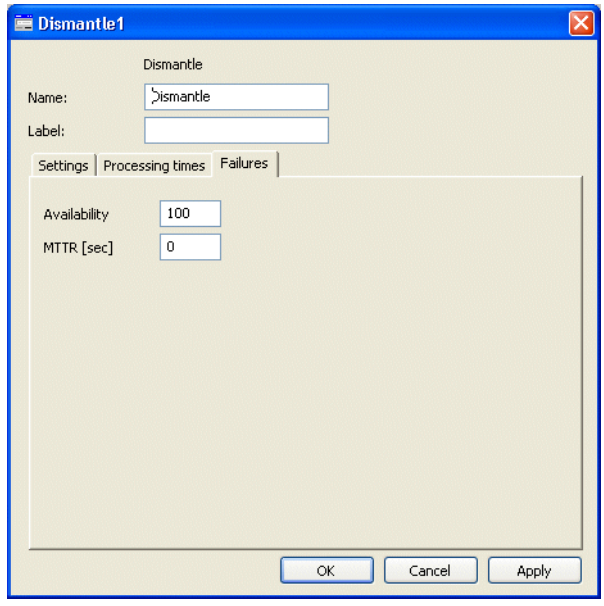
These radio buttons are active only, when you selected the radio button Serve defined stations.

- Immediately: the parts are created as soon as this is possible.
- As soon as the stations are available: the parts are created only when all stations are available and thus all parts can be created. Make sure that the succeeding stations have enough capacity to take in the parts, otherwise model might be blocked.



Dismantle Time

Select one of the distributions from the drop-down list and enter the parameters which this distribution requires. Plant Simulation shows which parameters the distributions require above the text box.



Availability, MTTR

Enter the availability and the mean time to repair into these text boxes.

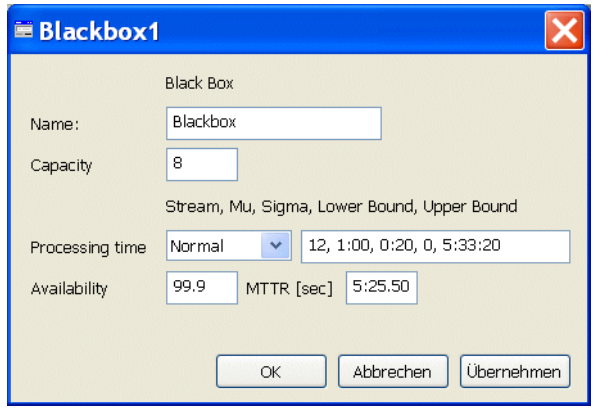
Attributes of the Object

objType: Dismantle

Blackbox

You can use the BlackBox to mix the sequence of MUs. Define a distribution for the dwelling time of the MUs. MUs can pass each other in the BlackBox. As soon as an MU enters the BlackBox, the program rolls the dice for setting the dwelling time according to the settings you chose. After the dwelling time has elapsed, the MU exit the BlackBox again.

When you define a constant dwelling time, the sequence of MUs remains the same.



Name:

This text box shows the name of the object.

Capacity

Enter the capacity of the *BlackBox* into this text box. The stations are arranged in parallel.

Processing Time

Select one of the distributions from the drop-down list and enter the parameters which this distribution requires. Plant Simulation shows which parameters the distributions require above the text box.

Availability, MTTR

Enter the availability and the mean time to repair into these text boxes.

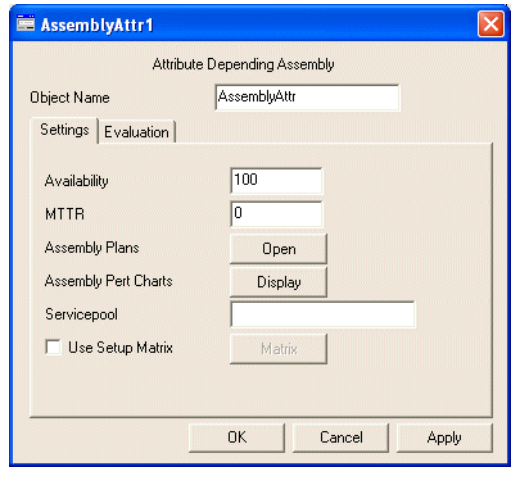
Attributes of the Object

objType: Blackbox

AssemblyAttr

This assembly station processes parts independent of their attributes, Each part can have any number of attributes. You can select in the object which assembly plan applies to which part type.

An assembly plan may consist of any number of sequential and/or parallel operations. For each operation you can specify the processing time as well as the parts to be used.



Object Name

This text box shows the name of the object.

Tab Settings

Availability, MTTR

Enter the availability and the mean failure time into these text boxes.

Assembly Plans

Click Open to open the table into which you enter the assembly plans. Here you define the assembly plans for the different part types and attribute combinations.

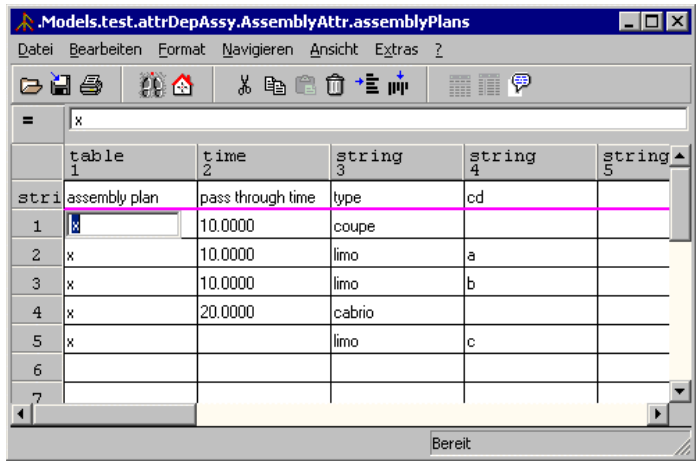


	table 1	time 2	string 3	string 4	string 5
stri	assembly plan	pass through time	type	cd	
1	x	10.0000	coupe		
2	x	10.0000	limo	a	
3	x	10.0000	limo	b	
4	x	20.0000	cabrio		
5	x		limo	c	
6					
7					

Enter the part attributes which the *AssemblyAttr* accesses into the column indexes 4 to n. Then, enter the values into the cells of the columns, which the part attributes may take. In our example above we entered the part attributes `type` and `cd`, which means that the *AssemblyAttr* only accesses these two attributes. In this case additional part attributes are not relevant for determining the assembly plan.

Enter the assembly plan for the various value combinations into column 1.

Enter an identifier of your choice into the column assembly plan to create a subtable. You can open this subtable by pressing F2.

Any parts for which you did not enter an assembly plan into the first column, will use up the time you enter into the column pass through time.

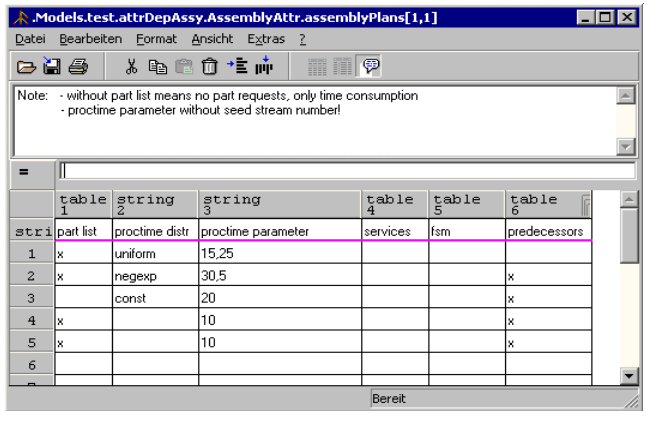


	table 1	string 2	string 3	table 4	table 5	table 6
stri	part list	proctime distr	proctime parameter	services	fsm	predecessors
1	x	uniform	15,25			
2	x	negexp	30,5			x
3		const	20			x
4	x		10			x
5	x		10			x
6						

The assembly plan can consist of several operations which are processed sequentially one after the other or in parallel to each other. Each row in this table defines an operation. Define the sequence of operations in the column predecessors. Enter which operations have to be finished before the operation itself starts into the subtable.

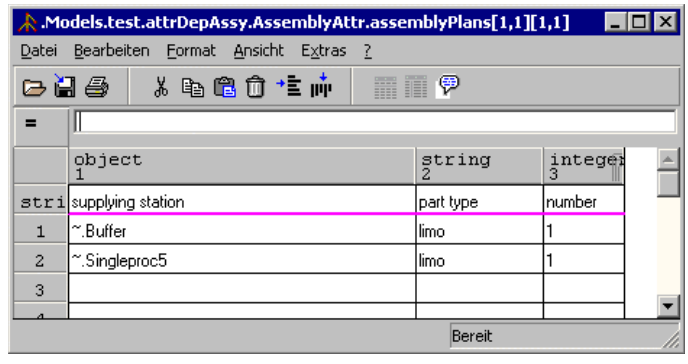
Enter the duration of an operation into the cells in columns 2 and 3. Enter the distribution into the column proctime distr. Enter the corresponding parameters, without the seed stream, into the column proctime parameter. When you do not enter anything into the column proctime distr, Plant Simulation Assembly uses the setting const.

An operation may require these elements:

- Parts which are attached.
- Services which are ordered from a *Servicepool*.
- *Final State Machines* which are requested.

Column part list

Enter the parts which are to be used by this operation into this subtable.

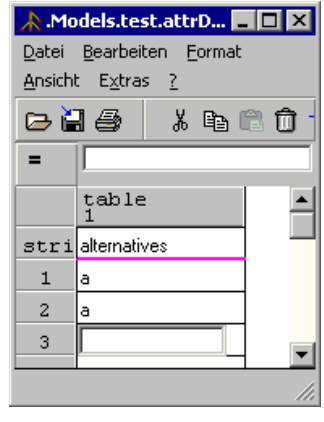


Enter this data into the columns:

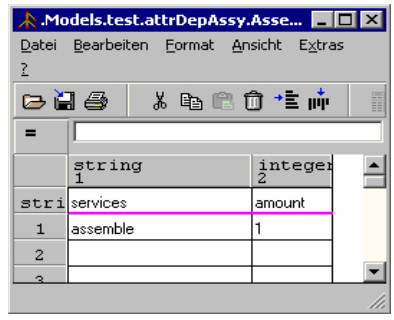
- supplying station: Enter the objects from which the object *AssemblyAttr* pulls the parts. In addition to the material flow objects, you can also enter the objects Logistics or Shelf. In both cases parts are attached which the tables in these objects manage.
- part type: Enter the part type which is to be used into the cells in this column.
- number: Enter the number of parts into the cells of this column.

Column services

Enter the services or resources used for attaching the parts into this subtable. Enter an identifier of your choice to create this subtable.



- alternatives: Enter an identifier of your choice into the cells of this column to create this subtable. Each cell in this table contains an alternative group.



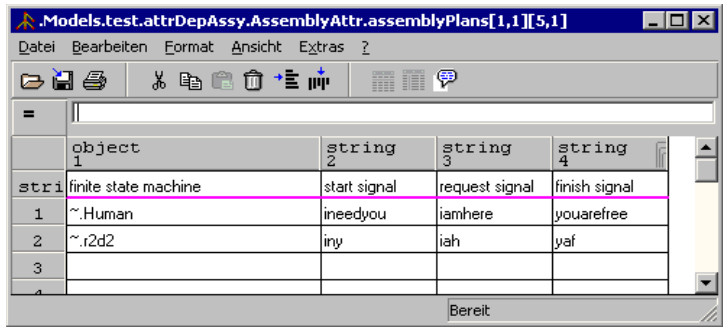
- services: Enter the names of the services into the cells of this column. The station then orders these services from the *ServicePool*.
- amount: Enter how often this service is requested into these cells.

Column fsm

Enter the *Finite State Machines* (*FSM*) which are required for executing the operation into these cells.

The station *AssemblyAttr* and the finite state machines communicate via signals. As soon as an *FSM* is required, you request it with a signal, which you can define. As soon as an *FSM* is available, it sends a signal saying so. After the operation is finished, the *AssemblyAttr* lets the *FSM* know this. Then the *FSM* can take on other tasks.

Create the subtable into which you enter the *FSMs* and the signals by entering an identifier of your choice into these cells.

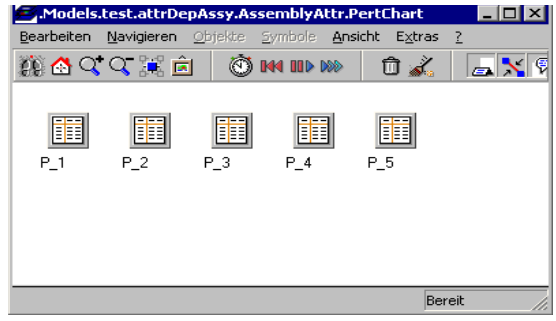


Enter this data into these columns:

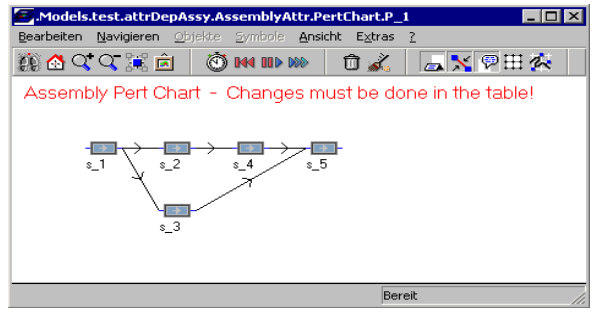
- finite state machine: Enter the paths to the *FSMs* into the cells of this column. You can also employ drag and drop.
- start signal: This is the signal which the *AssemblyAttr* sends to the *FSM* as soon as it needs it.
- request signal: This is the signal for which the *AssemblyAttr* waits. As long as it does not receive this signal, the operation cannot be started. Note that the *FSM* sends the signal.
- finish signal: This is the signal which the *AssemblyAttr* sends to the *FSM* as soon as the operation is finished.

Assembly Chart

Click Display to show the assembly plans you defined above. Note that you cannot change the plans in the chart.



The *AssemblyAttr* creates an object for each assembly plan which shows the assembly plan as a chart in a *Frame*. Double-click an object to open the *subframes*.



Each operation of the assembly plan is represented by an object. The sequence of operations is visualized by the connections between the objects.

Servicepool

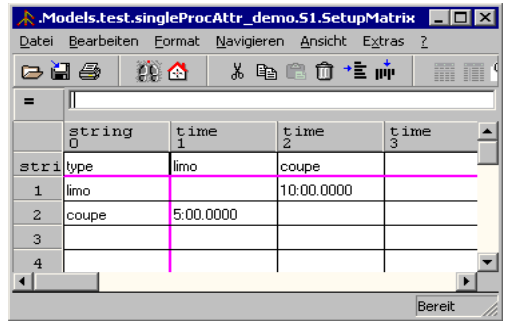
Enter the path to the associated *Servicepool* into this text box. This *Servicepool* assigns the services to the *AssemblyAttr*.

You can either enter the name of the *Servicepool* object into the text box or you can drag it over the icon of the *Servicepool* and drop it there.

The *Servicepool* is then required when services are requested in the assembly plans.

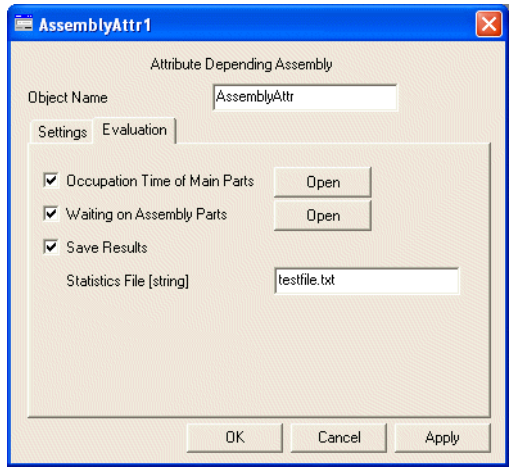
With Set-up Matrix

Use this check box to set, if the set-up time will be defined in a matrix.



Enter the predecessor types into the row index (column 0) and the successor types into the column index (row 0).

Note: The set-up times are determined depending on the part attribute *Entitytype*. So that the set-up will start at all, the parts have to have different names which is a standard functionality of Plant Simulation.



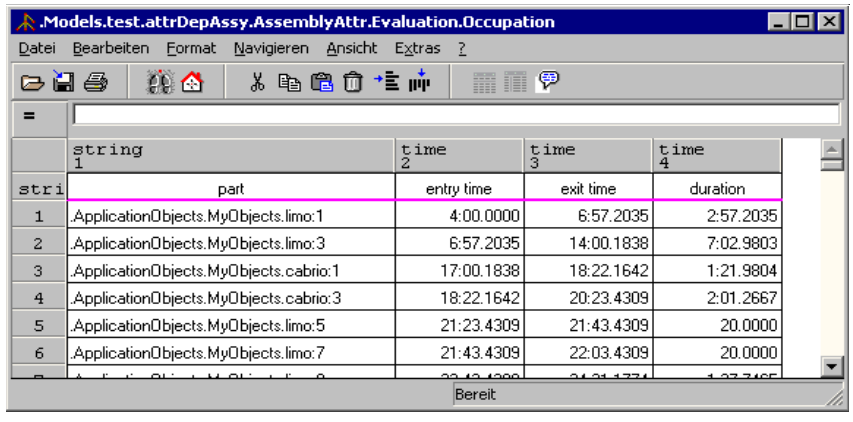
Tab Evaluation

In addition to the dwelling times of the main parts the *AssemblyAttr* can also record the waiting times for mounting parts.

dwelling time Main Parts

Select this check box to record the dwelling time of the main part on the *AssemblyAttr* station.

Note: Be aware that recording the dwelling times of the main parts requires large amounts of RAM.



	string 1	time 2	time 3	time 4
stri	part	entry time	exit time	duration
1	.ApplicationObjects.MyObjects.limo:1	4:00.0000	6:57.2035	2:57.2035
2	.ApplicationObjects.MyObjects.limo:3	6:57.2035	14:00.1838	7:02.9803
3	.ApplicationObjects.MyObjects.cabrio:1	17:00.1838	18:22.1642	1:21.9804
4	.ApplicationObjects.MyObjects.cabrio:3	18:22.1642	20:23.4309	2:01.2667
5	.ApplicationObjects.MyObjects.limo:5	21:23.4309	21:43.4309	20.0000
6	.ApplicationObjects.MyObjects.limo:7	21:43.4309	22:03.4309	20.0000

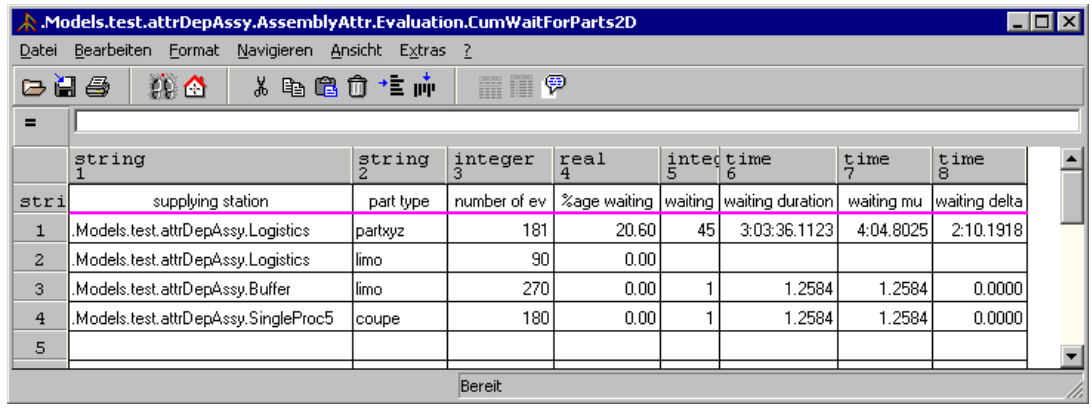
Enter this data into the columns:

- part: Shows the path of the part.
- entry time: Shows the entry time of the part into the *AssemblyAttr* station.
- exit time: Shows the exit time of the part from the *AssemblyAttr* station.
- duration: Shows the dwelling time of the part in the *AssemblyAttr* station.

Waiting Times for Mounting Parts

Select this check box to record the waiting times for mounting parts of the *AssemblyAttr* station.

Note: Be aware that recording the waiting times of the main parts requires large amounts of RAM.



	string 1	string 2	integer 3	real 4	integ 5	time 6	time 7	time 8
stri	supplying station	part type	number of ev	%age waiting	waiting	waiting duration	waiting mu	waiting delta
1	.Models.test.attrDepAssy.Logistics	partxyz	181	20.60	45	3:03:36.1123	4:04.8025	2:10.1918
2	.Models.test.attrDepAssy.Logistics	limo	90	0.00				
3	.Models.test.attrDepAssy.Buffer	limo	270	0.00	1	1.2584	1.2584	0.0000
4	.Models.test.attrDepAssy.SingleProc5	coupe	180	0.00	1	1.2584	1.2584	0.0000
5								

Enter this data into the columns:

- supplying station: Path of the station which supplied the mounting part.
- part type: Part type of the mounting part.
- number of ev: Number of times the mounting part was used.
- %age waiting: Percentage of the waiting time in relation to the statistics processing time.
- waiting: Number of times the object was waiting.
- waiting duration: Summed up waiting time.
- waiting mu: Mean waiting time.
- waiting delta: Standard deviation of the waiting time.

Save Results

Select this check box to save the results of the statistics at the end of the simulation experiment to disk. This makes sense when you want the *ExperimentManager* to execute several simulation runs and you do not want to overwrite the statistics of the individual simulation run.

Statistics File

Enter the name of the statistics file into this text box.

Attributes of the Object

objType: AssemblyAttr

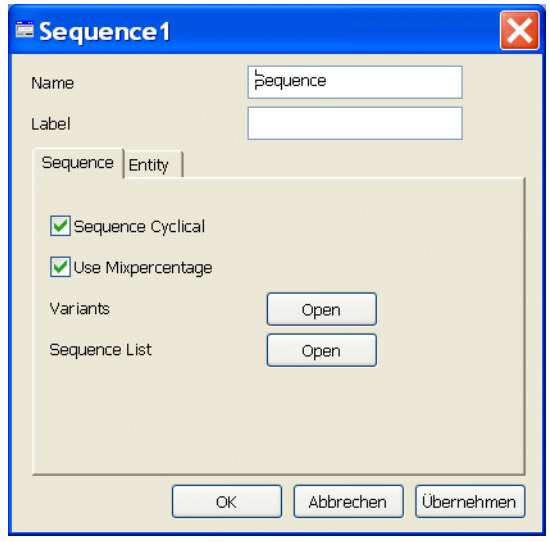
2

Controls

The application object library Assembly provides a number of objects to control the flow of materials. You can, among others, define protective circuits, synchronize material flows or define product sequences.

Sequence

The *Sequence* provides a sequence of variants which, for example, the *Source* objects access. The object *SourceN* produces parts according to the sequence defined in the *Sequence* object.



Name, Label

These text boxes contain the name or the label of the object respectively.

Tab Sequence

Sequence Cyclical

Select this check box to set if the sequence of the variant list will be created repeatedly. Clear it to create it once only.

Use Mixed Percentage

Select this check box to take the percentages of the respective variants into account when generating the sequence list. Clear it to create the sequence in a fixed cyclic sequence.

When you select the check box Use Mixed Percentage, the sequence of the sequence list to be generated in not defined in a fixed manner. The settings for the percentages of the variants (column mixpercentage), as well as the respective number, which you enter for creating a batch of a variant (column Batchsize), are taken into account when generating the sequence list.

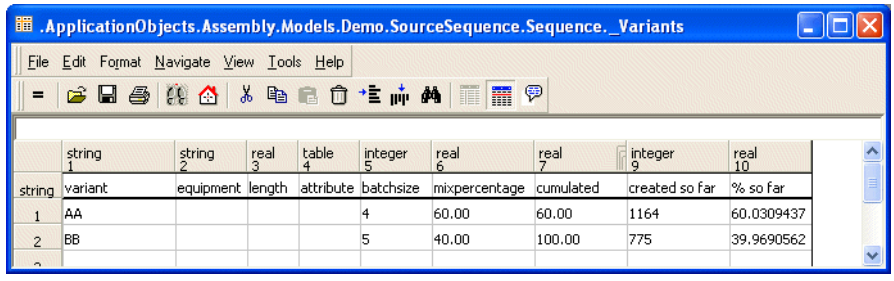
When you do not select Use mixed percentage the variant in the column Variant will be generated with the number you entered into the column Batchsize when generating the sequence list. The percentages you entered into the column Mix-percentage will be ignored.

When you clear the check box Sequence cyclical the program does not evaluate the column mixpercentage. The program processes the variant table once only and creates the defined MUs. When you select the check box Sequence cyclical you can, in addition, use the mixed percentage for defining the MU sequence.

When you clear the check box Use mixed percentage, the program creates the MUs, which you defined in the variant table, in the sequence defined in the table. It creates the number of MUs, which you entered into the cells of the column Batchsize. Once all MUs of the variant table have been created, the program starts processing the table again, starting in row 1.

When you select Use mixed percentage, Plant Simulation Assembly creates the MUs with the percentages you defined. It produces them in the batches which you entered into the cells of the column Batchsize.

Variants Table



	string 1	string 2	real 3	table 4	integer 5	real 6	real 7	integer 9	real 10
string	variant	equipment	length	attribute	batchsize	mixpercentage	cumulated	created so far	% so far
1	AA				4	60.00	60.00	1164	60.0309437
2	BB				5	40.00	100.00	775	39.9690562

- variant: Enter the names of the variants into the cells in this column.
- equipment: You can, but you do not have to, enter the part attribute equipment into the cells of this column.
- length: You can, but you do not have to, enter the length of the part into the cells of this column.
- attributes: In the cells of this column you can set additional type-specific attributes in a subtable. Create the subtable by entering an identifier of your choice into the cell.
- batchsize: Enter the number of parts of a variant which will be entered into the sequence list. The value has to be greater than 0.
- mixpercentage: When you want to fill the sequence list using a frequency distribution, enter the respective percentages into the cells of the column mixpercentage. Enter the how many variants of the same type are to be produced in sequence into the cells of the column batchsize.

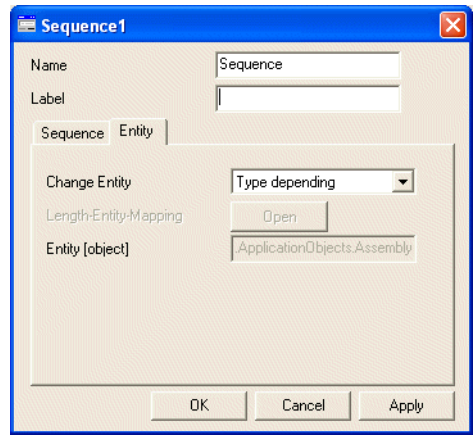
Note: After the initialization (init) of the simulation model you can check if the column cumulated has been filled in correctly. The value in last row has to be 100. The table Sequence List contains the first variants of the created sequence. Note that you cannot edit this table, it just serves for checking the settings you selected.

- created so far: The cells in this column record how many MUs of this variant have been created so far.
- % so far: The cells in this column calculate the percentage of total amount of created MUs. After a certain burn in phase this value should about match the value in the column Mixpercentage.

Sequence List

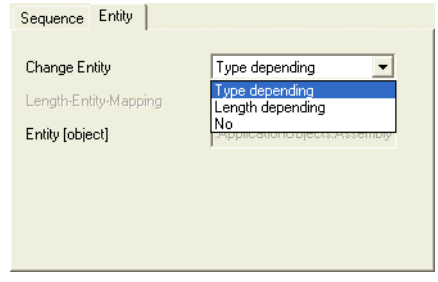
The Sequence List shows the sequence of the part types which the object *Sequence* produced. Note that the program only shows this information. You cannot edit the values.

Tab Entity

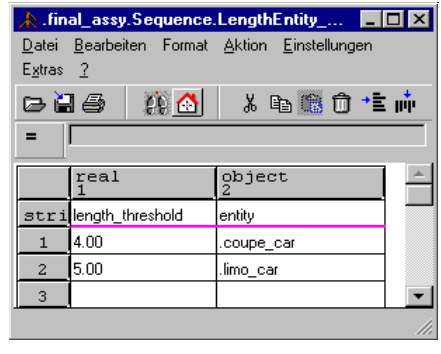


Change Entity

Select from this drop-down list if the connected sequential *Sources* will create instances of different MU classes or if they will always produce instances of the same MU class.



- Type-dependent: The names of the MU classes determine the variants in the variant list. Note that for each variant an MU class with the same name has to be present in the class library of the AOL. You can define the icons of the MUs in the sequential *Sources*. For this setting you can assign variant-specific icons to various MU classes. You can, for example, use variant-specific icon operational and waiting for each variant. If the program does not find the corresponding MUs in the folder *MyObjects*, it will automatically create the MUs in this folder.
- Length-dependent: The lengths of the variants determine the MU classes. Click Open and enter the MU classes into the table LengthEntity_Mapping:



Enter the threshold values into the cells in the column length_threshold up to which the MU classes, which you enter into the column entity, will be used.

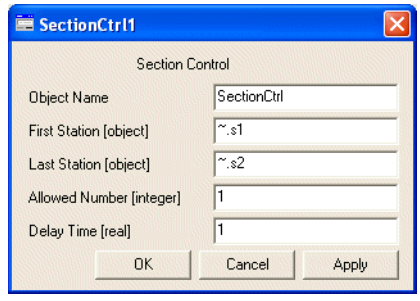
- No: Enter the name of the MU class into the text box Entity. The *Sources* always create instances of the same MU class for the variants.

Entity

Enter the MU class of which you would like the sequential *Sources* to create instances.

SectionCtrl

You can use this object to make sure that only a certain amount of MUs is located in the area which you define.



Object Name

This text box shows the name of the object.

First Station

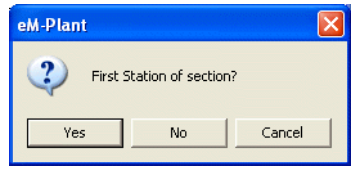
Enter the first station of the section into this text box. Depending on the number of MUs which you want to allow in this section the entrance of this station is temporarily closed.

Last Station

Enter the last station of the section into this text box. Depending on the number of MUs which you want to allow in this section the first station is opened again, when the MU exits the last station.

Note: When you enter a relative path, remember that the path starts with *location*, as the entry starts within the object *SectionCtrl*. The program automatically adds *.workplace* to the path of the object if this is required.

You can also drag the stations onto the *SectionCtrl* and drop them there. This action opens this dialog:



Click Yes to enter this station as the First station. Click No to enter this station as the last station.

Allowed Number

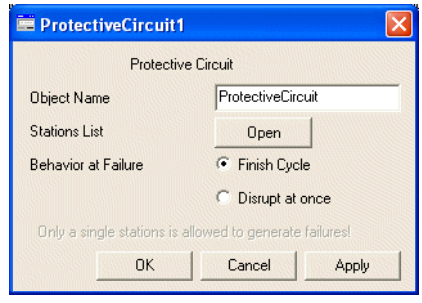
Enter the maximum number of MUs which may be located within this section.

Delay Time

Enter the Delay time for opening the entrance of the first station. Enter the time in seconds.

ProtectiveCircuit

This object allows you to combine a number of stations into a protective circuit.



Object Name

This text box shows the name of the object.

Stations List

Enter the station which you would like to combine into a *ProtectiveCircuit* into the Stations List.

Note: The program checks the entries of the table after you close it. It transforms absolute paths into relative paths and, if required, adds *.workplace* to the name of the station. This enables dragging stations over the table and dropping them there. You can also drag the stations over the icon of the *ProtectiveCircuit* and drop it there.

The stations do not necessarily have to follow each other in the sequence of stations.

Behavior on Failures

With these radio buttons you can set how failures are transmitted.

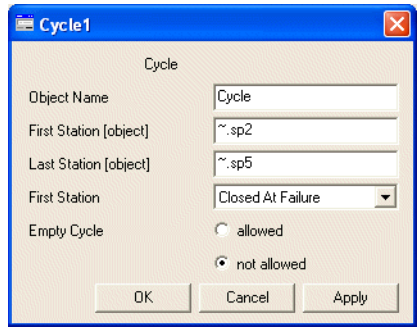
- Finish Cycle: The ProtectiveCircuit closes the entrance of the succeeding station, so that each station has to finish its cycle. Closing the entrance also affects the objects which follow the *ProtectiveCircuit* in the sequence of stations. When the last station is connected with its successors with *Connectors*, their entrance will be closed when a failure occurs. When you enter an exit control (front activated) into the last station, instead of connecting it with a *Connector*, the program adds a waituntil instruction to the method. You can define failures for each individual station which is part of the *ProtectiveCircuit*.
- Disrupt at once: All stations which are part of the *ProtectiveCircuit* will be disrupted at once. Here the failure of a single station controls the disruption, i.e. you may only define failure parameters for a single station which is part of the *ProtectiveCircuit*.

Cycle

You can use the object *Cycle* to synchronize the transfer of parts. A *Cycle* object can synchronize a group of material flow objects. The section to be synchronized may not contain any objects of type *Buffer*.

You can insert any number of *Cycle* objects into your simulation model.

Note that all station to be synchronized have to be connected with *Connectors*.



Object Name

This text box shows the name of the object.

First Station

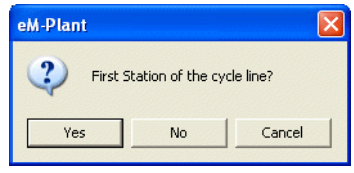
Enter the first station of the line to be synchronized into this text box.

Last Station

Enter the last station of the line to be synchronized into this text box.

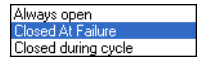
Note: When you enter a relative path, remember that the path starts with *location*, as the entry starts within the object *Cycle*. The program automatically adds *.workplace* to the path of the object.

You can also drag both stations onto the object *Cycle* and drop them there. This action opens this dialog:



Click Yes to enter this station as the First station. Click No to enter this station as the last station.

Behavior of the First Station



You can select from this drop-down list how the first station behaves during the cycle.

- Always open: During the cycle parts may be transferred to the first station This implicates though that all stations of the synchronized line have to wait until the first station has processed a part.
- Closed at failure: During the cycle parts may be transferred to the first station as long as no station of the synchronized line is failed or paused. The entrance of the first station will be closed as soon as a station is failed or paused. In general the behavior described above applies.
- Closed during cycle: During the cycle no parts may be transferred to the first station While synchronizing the entrance of the first station will be opened and any part ready to exit the predecessor will be transferred into the first station.

Empty Cycle

Use these radio buttons to set if empty cycles are permitted or not permitted. You can only select this setting when the first station is not closed during the cycle. When the first station is closed during the cycle, empty cycles are always permitted.

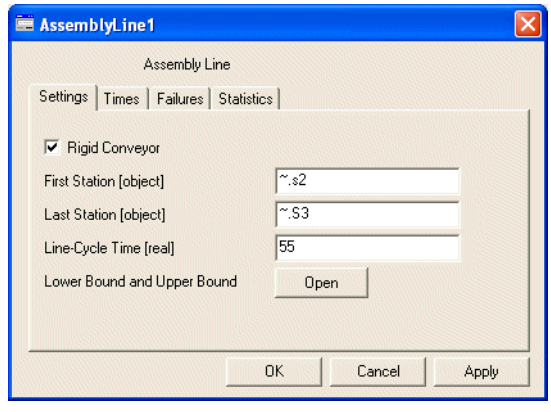
Note: Note that you cannot use the object *Cycle* with these objects:

ParallelProc, *Assembly* station, *DismantleStation*, *Buffer*, *PlaceBuffer* and *Line* as well as the *Assembly* objects *Blackbox*, *DistributorType*, *Dismantle*, *SourceN* and *Sink*.

AssemblyLine

You can combine several work stations to an assembly line with the object *AssemblyLine*. We attached special importance to modeling the behavior of the *Workers* on this line. We assume that a *Worker* may not only work at the station he is assigned to but also at the preceding station and at the succeeding station. You can define the portion of this preparatory work and subsequent work in percent of the cycle time for each station (and thus in percent of the length of the station) in the table Lower Bound and Upper Bound on the tab Settings.

Tab Settings



Rigid Conveyor

Select this check box to define the conveyor as a rigid one. Clear it to make it an accumulating conveyor.

First Station

Enter the first station of the *AssemblyLine* into this text box.

Last Station

Enter the first station of the *AssemblyLine* into this text box.

Note: When you enter a relative path, remember that the path starts with *location*, as the entry starts within the object *AssemblyLine*. The program automatically adds *.workplace* to the path of the object.

You can also drag both stations onto the object *AssemblyLine* and drop them there. This action opens this dialog:

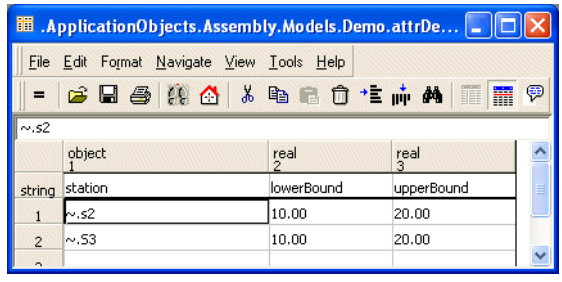
Click Yes to enter this station as the First station. Click No to enter this station as the last station.

Line Cycle Time

Enter the cycle time, i.e. the speed of the conveyor, into this text box.

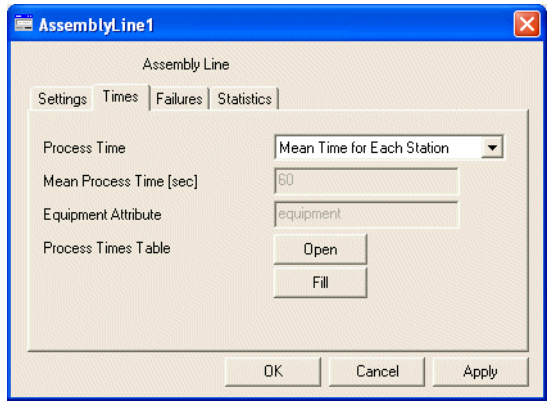
Lower Bound and Upper Bound

Click Open to open the table Stations. Enter the stations which constitute the conveyor section. You can enter lower bound and the upper bound for each individual station.



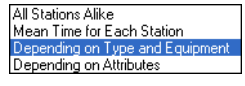
Note that the program automatically enters the first station and the last station into this table. Enter the preparatory work factor, which is a percentage based on the processing time into the cells of the column LowerBound. Enter the subsequent work factor into the column UpperBound.

Tab Times

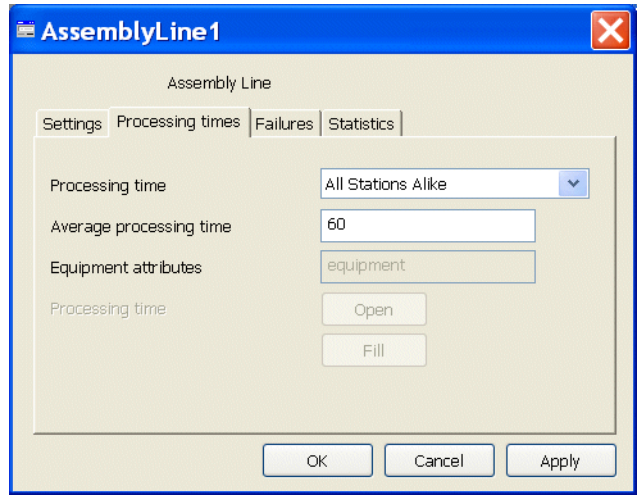


Processing Time

You can define the processing times in four levels of detail:



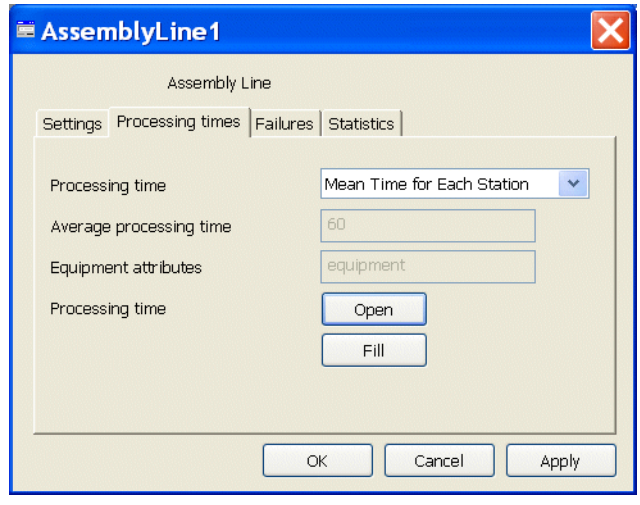
Setting "All Stations Alike"



All stations use the same processing time.

Average Processing Time: Enter the mean processing time for all stations into this text box.

Setting "Mean Time for Each Station"



Click Open and enter a specific processing time for each station into the table Processing times table. In addition you can define for each station if the processing time is constant or determined by an empirical distribution.

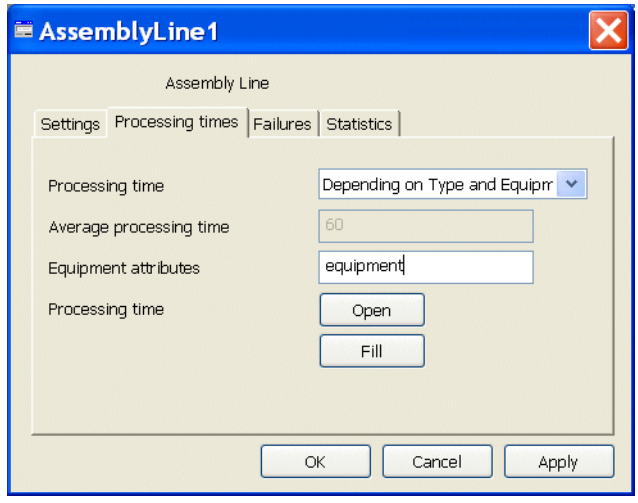
	object 0	string 1	real 2
string	station	discr. distribution	proctime [s]
1	*.ApplicationObjects.Test_Modelle.AssyBelt.attrDepAssy_belt1.S2		
2	*.ApplicationObjects.Test_Modelle.AssyBelt.attrDepAssy_belt1.S3		
3	*.ApplicationObjects.Test_Modelle.AssyBelt.attrDepAssy_belt1.S4		
4			

Processing Times Table: Click Fill to automatically enter the stations from the table Lower Bound and Upper Bound into the Process Times Table.

- station: The program automatically enters the names of the stations from the table Lower Bound and Upper Bound.

- discr. distribution: You can enter the name of the distribution object (*Distr* or *Distr_Ref*) into the cells of this column. These objects are located on the top-level *Frame* of the model. They contain a discrete empirical distribution. When you do not enter anything, the program interprets the value you entered as the proctime as a constant value.
- proctime: Enter the processing time into the cells of this column. When you enter a distribution into the cells discr. distribution, the program interprets the value in the column proctime as the mean value of this distribution. Note that you enter seconds here.

Setting "Depending on Type and Equipment"



The processing times on the stations of the conveyor section depend on the part type, i.e. the attribute Entitytype of the MU, and on the attribute Equipment. The attribute equipment determines which MU attribute will be used for selecting the equipment type in this conveyor section. Enter the processing times into the Processing times table.

Processing Times Table

	string 0	table 1
string	type	proctimes [s]
1	limo	t1
2	coupe	t2
3		

Enter the part types into the cells of the column *type*. If different types have the same processing times, you can enter them into a single cell. Separate them with commas.

Once you have entered the values you want into the cells of the column *Entitytype* clicking the button *Fill* automatically enters data into the cells of the column proctimes. The program creates this subtable in the column *proctimes* in each row of the *Processing times table*.

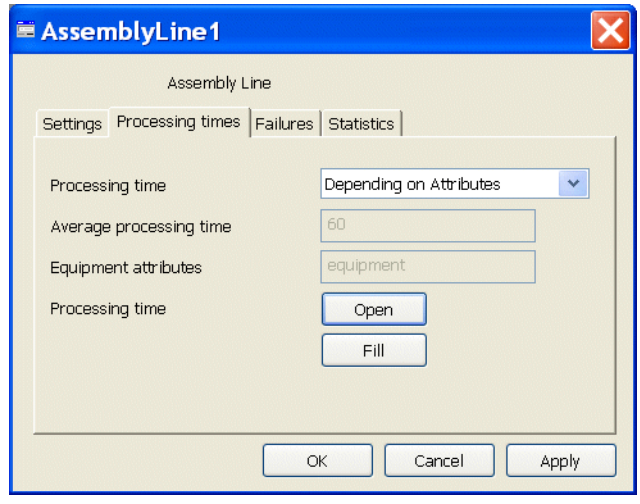
	object 0	string 1	real 2	real 3	real 4
string	station	discr. distributi	default		
1	~.s2		25.00		
2	~.S4		28.00		
3	~.S3		26.00		
4					

- station: The program automatically enters the stations into the cells of this column, which you entered into the stations table.

- discr. distribution: You can enter the name of a distribution object into the cells of this column. This object is located on the top-level *Frame* of the model. It contains a discrete empirical distribution. When you do not enter anything, the program interprets the value you entered into the column default or as the equipment type as a constant value.
- Default: Enter the processing time into the cells of this column, provided the processing time for all equipment types is the same for this station. In this case you do not have to fill in the following columns.

Settings "Depending on Attributes

Here the processing times at the stations of the line section depend on attributes which you define.



Processing Times : Click Fill to automatically enter the stations, which you entered into the table Lower Bound and Upper Bound into the Process Times Table.

	object 0	table 1
string	station	proctimes
1	~.s2	t1
2	~.S3	t2
3	~.S4	t3
4		

- station: The program automatically enters the names of the stations from the table Lower Bound and Upper Bound.
- proctime: Enter an identifier of your choice into the cells of this column to create a subtable. Then enter the attributes and the associated processing times into this subtable.

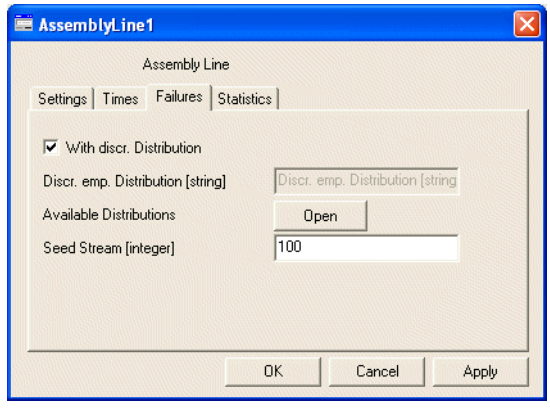
	string 1	string 2	string 4	string 5
string	proctime distribution	proctime parameter	EntityType	
1	const	25	limo	
2	uniform	20,28	coupe	
3				

Enter these values into the subtable:

- proctime distribution: Enter any of the distribution functions which eM-Plant provides into these cells. When you do not enter anything, the program uses a constant processing time.
- proctime parameter: Enter the parameters of the distribution function into these cells. When you selected a constant processing times, enter this time here.
- Following columns: Enter the part attributes which the *AssemblyLine* accesses into the cells of columns 4 to n. Then, enter the values into the cells of the columns, which the part attributes may take.

Note: When entering the parameters for the distribution, you do **not** have to enter the random number stream.

Tab Failures



With Discrete Distribution

Select this check box to use the discrete empirical distributions which you entered into the Processing Times Table. Clear it to use a constant processing time. This way you can switch to constant processing times without having to delete the discrete empirical distributions which you entered before.

Discrete Empirical Distribution

Enter the name of the discrete empirical distribution. It is assumed that this object is located on the topmost *Frame* of your model.

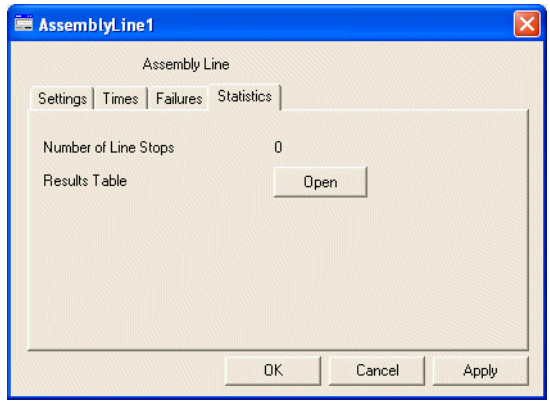
Available Distributions

This list contains the names of the distributions which you can enter into the column discr. distribution. Note that the program automatically updates the list, when you insert or delete distribution objects.

Random Number Stream

Enter the random number stream into this text box.

Tab Statistics

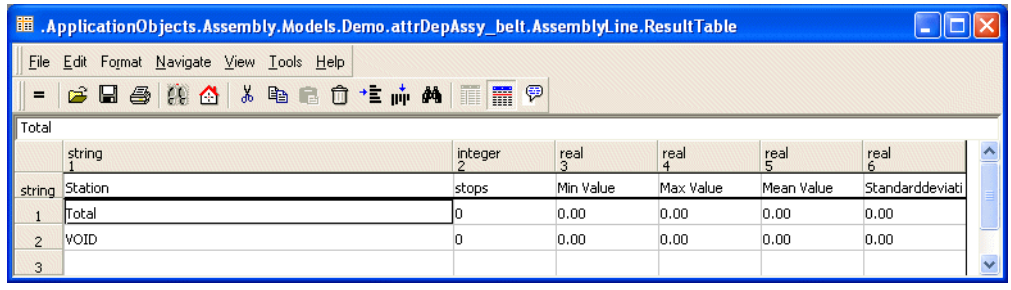


Number of Line Stops

This value shows how often the line was stopped.

Results Table

Click Open to view the values relating to line stops which Plant Simulation assembly recorded.



	string 1	integer 2	real 3	real 4	real 5	real 6
string	Station	stops	Min Value	Max Value	Mean Value	Standarddeviati
1	Total	0	0.00	0.00	0.00	0.00
2	VOID	0	0.00	0.00	0.00	0.00
3						

- station: This column shows which stations caused the line to stop.
- number: This column shows how often this station caused the line to stop.
- min_value: This column shows the shortest duration of a line stop this station caused.
- max_value: This column shows the longest duration of a line stop this station caused.
- mean_value: This column shows the mean duration of a line stop this station caused.
- standarddeviation: This column shows the associated standard deviation.

Course of Action

Plant Simulation Assembly calculates how often the *Worker* worked during this cycle for each station (worker cycle distribution). When this time is greater than the line speed, the exceeding time will be added as penalty time. When this time is shorter, the time will be subtracted.

When the failure time exceeds the permitted value (exceeding factor times cycle time), the line or the station (accumulating) respectively have to be stopped for the entire failure time. When you use a rigid conveyor all attributes called *UpperBound* will be reset. Failures and pauses of individual stations within the line section are also propagated to all stations.

To prevent the *Worker* from doing preliminary work, enter `0` as the LowerBound. To enforce that the job is done within the length of the station, set the UpperBound to `0`.

Note: When employing discrete empirical distributions, you can only use instances of the object class *.Distr_Reference* in which the distributed times relate to a reference value. To accomplish this, insert the object *Distr_Reference* into the topmost *Frame* of your model and point to this distribution with the instance name of the distribution object within the object *AssemblyLine*.

Attributes of the Object

objType: AssemblyLine

Methods of the object

Statistic

Usage:
```
<path>.statistic(<table>)
<path>.statistic(<table_path>)
```

This method returns the values from the tab Statistics and writes them into a table. You can pass the name of a table or an object of type table. Plant Simulation automatically formats the table.

3

Moving Objects

The moving objects (MUs) of Plant Simulation Assembly are located in the folder *PartsAndOrders*. We defined the attributes which all objects of the library require for these MUs.

Note: If your modeling needs require MUs with special attributes or special icons, we recommend to copy the MU class which most closely matches your needs and modify these. This ensures that all required attributes exist. You can place these MUs into the folder *MyObjects*.

The MUs are treated according to their internal class name. This mainly applies for the MUs *Skid*, *Carbody* and *Dolly*. For these MUs we assume that they transport parts and that the transported parts are to be processed. In particular *Skids*, *Carbody* and *Dolly* usually just pass through the stations. On some stations the processing time depends on the number of parts which a *Skid*, *Carbody* or *Dolly* transports.

Entity

The *Entity* represents the moving object (MU) which travels through the model during a simulation run. It has all the attributes which the objects in the application object library require. If you need to, you can define additional custom attributes.

Carbody/Dolly

The *Carbody/Dolly* represents a transport element, which has a drive of its own and which moves through the model on objects of type *Track*. It has all the attributes which the objects in the application object library require. As opposed to the *Entity*, the *Carbody* has a capacity it uses to load and transport parts/entities. If you need to, you can define additional custom attributes.

Skid

The *Skid* represents a passive transport element without a drive of its own. The *Skid* can load parts. One might compare it with a palette. You can use it to model any type of loading equipment. You can define the capacity of the *Skid* with its *XDim* and its *YDim*.

Note: Compare the setting Fixed Entity adjustable in the object *SourceN*. Compare the objects *Load*, *Unload* and *Transfer*.

4

Rework

In this chapter we describe the objects which you can use for modeling manufacturing errors, quality control and retouching. You can define rework in two different levels of detail.

When modeling rough retouching you just enter the percentage of parts which have to be retouched. When modeling detailed retouching you can, for each processing station, define the possible defect, the frequency with which this defect occurs and the duration of the repair job.

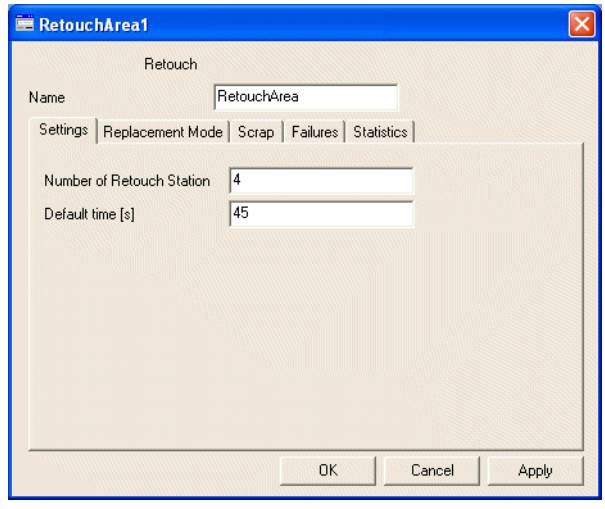
RetouchArea

You can model retouch jobs in two ways with the object *RetouchArea*:

- You enter a certain percentage which is to be treated as rework. Then, the object itself creates the defects which have to be treated accordingly.
- Enter defects which the object can fix. The object *Def_Retouch* creates these defects.

Defective parts will either be repaired or replaced with new parts of the same type.

In addition you can enter a percentage which will be treated as rejects and be completely removed from the line. Alternatively you can define a combination of defects which lead to the part being scrapped.



Name

This text box shows the name of the object.

Tab Settings

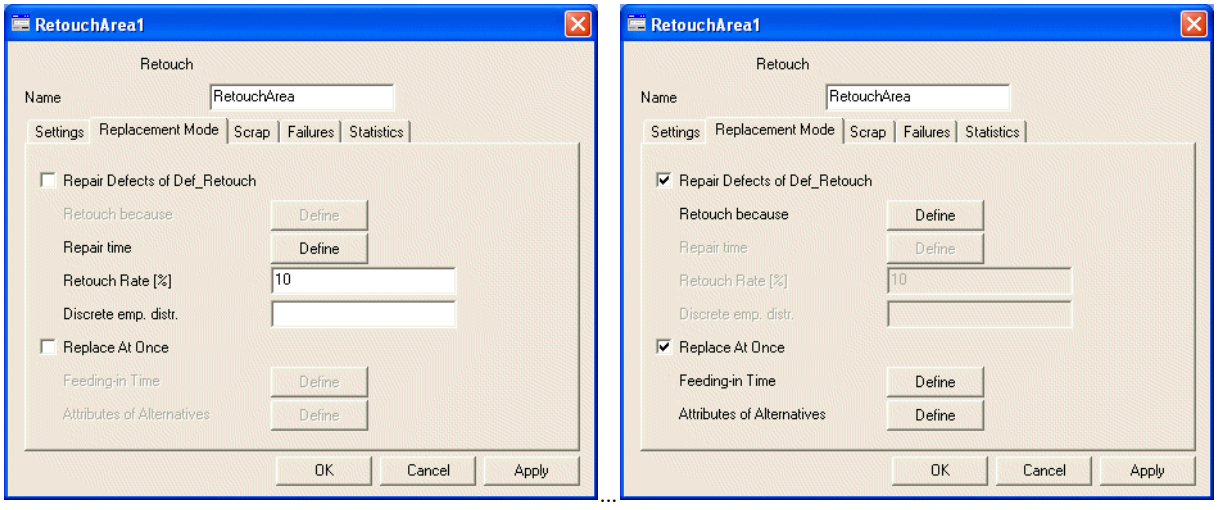
Number of Retouch Stations

Enter the number of retouch stations into this text box.

Default Time

When the MU does not have a defect, which can be fixed in this object, the program assigns the time you enter as the default time as dwelling time.

Tab Replacement Mode



Repair Defects Created by Def_Retouch

With this check box you can set the behavior described above. When you select the check box, the defects created by the object *Def_Retouch* will be repaired. When you clear the check box, you can use the object independent of the object *Def_Retouch*. Depending on what you select the program activates or deactivates additional dialog items.

Compare the description of the object *RetouchArea* in the user guide.

Retouch Because (Defects)

Click Define and enter which of the previously allocated retouch reasons will be processed in the object. When you leave the list empty each and every of the retouch reasons for the affected MU will be processed. This list is independent of the list which only causes parts with certain error symptoms to leave the line, compare the object *SingleprocRetouch*.

Note: Remember that you already defined the times for the retouch jobs in the object *Def_Retouch*, compare Def_Retouch, described on page 60. If several retouch jobs are performed on an MU, the program sums up the times.

Repair Time

Click Define to open a dialog in which you define the repair time. Note that you only have to enter data when the object is not to process any of the retouch reasons defined in *DefRetouch*.

Retouch Rate

Enter the scrap rate in percent into this text box. This percentage of parts will leave the line. Note that you only have to enter data when the object is not to process any of the retouch reasons defined in *DefRetouch*.

Discrete empirical distribution

Enter the name of a distribution object into this text box. This object is located on the top-level *Frame* of the model. It contains a discrete empirical distribution. Note that you only have to enter data when the object is not to process any of the retouch reasons defined in *DefRetouch*.

Replace at Once

You can use this check box to set if a part which has been removed from the line to be retouched is to be replaced by an identical part or if the removed part is to be repaired and then fed back into the line.

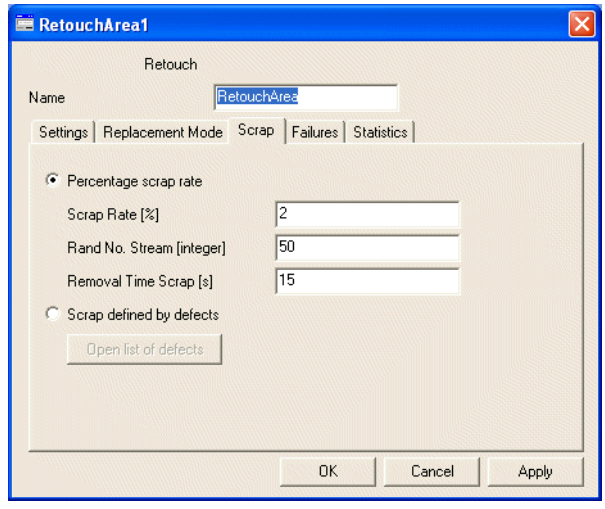
Feeding-in Time

Enter the allocated time for feeding the MU back into the line in the dialog of the *Variable* which opens.

Attributes of Alternatives

If a part removed from the line is to be replaced by an identical part at once, you can enter the custom attributes which will be used for the type definition into this list.

Tab Scrap



Scrap Rate Percentage

Select this radio button if you want to define the percentage of scrap parts which cannot be repaired. Enter values into the text boxes below.

Scrap Rate

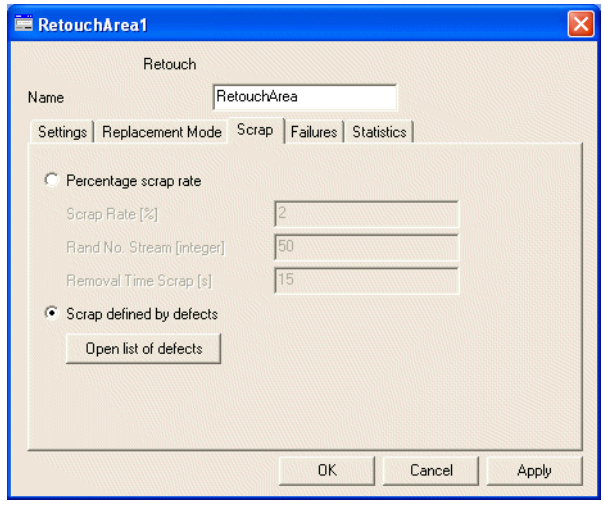
Enter the percentage of the retouched parts which is removed from the line as scrap.

Random Number Stream

Enter the random number stream into this text box.

Removal Time Scrap

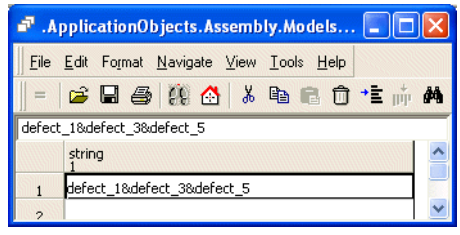
Enter into this text box how long it takes to recognize scrap parts and how long it take to remove them from the line.



Scrap Rate Defined by Defects

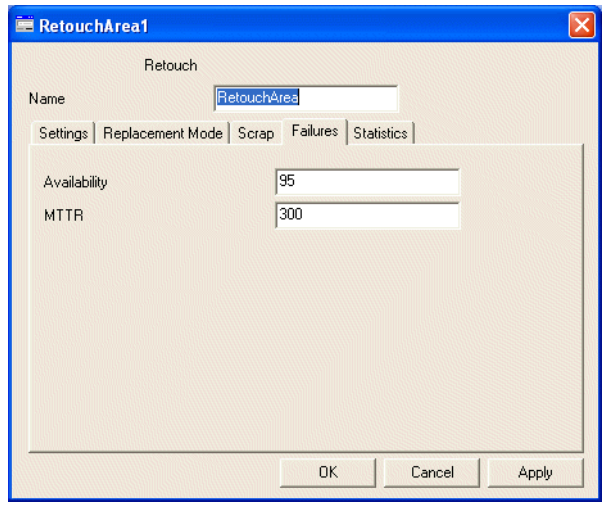
Next to defining the scrap rate by entering the percentage of the retouched parts, you can also define the scrap rate by combining a number of defects. When you select this radio button, you can click the button Open List of Defects and enter the defects which lead to scrapping this part.

Note: You can only select this setting when you selected the check box Repair Defects of Def_Retouch on the tab Replacement Mode.



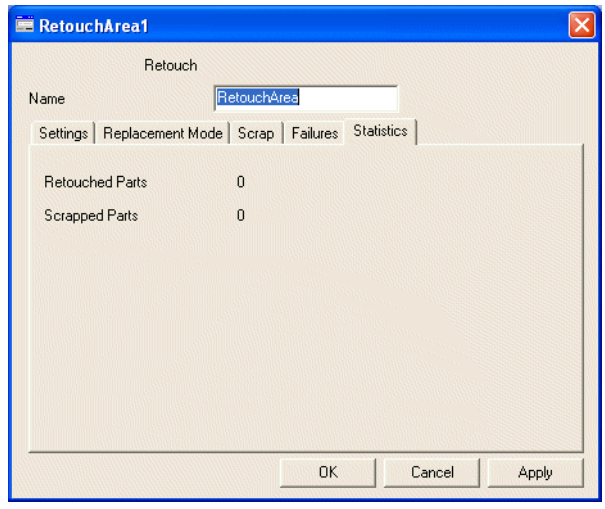
The individual rows within this table denote the alternatives (OR operation). Within a row you will enter an AND operation, defects linked with &. In the example above a part will be treated as scrap when it shows *defect_1 AND defect_3 AND defect_5*. All other combinations of defects will be retouched and the parts will not be treated as scrap.

Tab Failures



Enter the Availability in percent and the MTTR in seconds into the respective text boxes.

Tab Statistics



Retouched Parts

This values shows the number of retouched parts.

Scrapped Parts

This values shows the number of scrapped parts.

Attributes of the Object

objType: RetouchArea

statAvgRepairTime: The mean value of the repair time

statStdDevrepairTime: The standard deviation of the repair time.

number: the number of parts which are reworked.

scraped: the number of parts which are scraped.

Methods of the Object

Usage:
```
<path>.statistic(<table>);
<path>.statistic(<table_path>);
```

This method returns the statistical values of the retouchArea in a table. The content of the table is shown in the following picture

	integer 1	integer 2	time 3	time 4	time 5	time 6
string	No. of parts reworked	No. of parts scraped	Avg. rework time	Std. dev. rework time	Min. rework time	Max. rework time
1	2470	252	4:59.1255	6:05.4737	1:00.0000	59:00.0000
2						
3						

You can pass the name of a table or an object of type table. Plant Simulation automatically formats the table.

Description of the individual columns above:

No. of parts reworked: the number of parts which are reworked at this station

No. of parts scraped: the number of parts which are scraped at this station

Avg. rework time: the mean value of the rework time

Std. dev. rework time: the standard deviation of the rework time

Min. rework time: the minimum value of the rework time

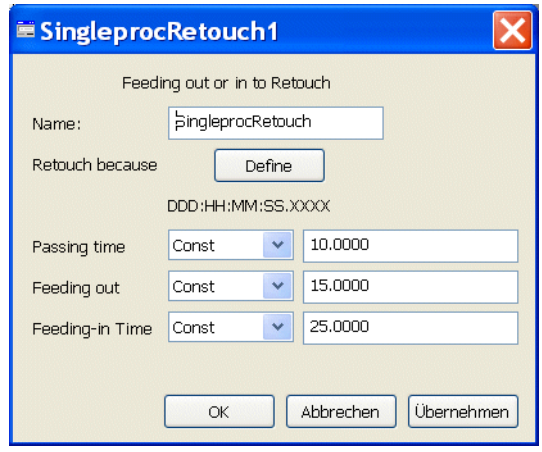
Max. rework time: the maximum value of the rework time

SingleprocRetouch

You can use this single processing station to remove parts to be retouched from the line and to feed them back into it once they have been retouched. You can, for example, use it to model a quality assurance station.

The station has two successors. The first one receives parts without defects, the second one the defective part which have to be reworked. This way you can model a quality assurance station. The station also has two entrances to feed retouched parts from the retouch area back into the line. When the parts are fed back into the line from the retouch station, the program does not check again if the part has additional defects.

When you connect the object with its predecessors and with its successors Plant Simulation opens a dialog in which you can select the respective entrances/exits.

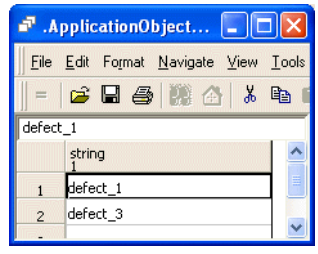


Name:

This text box shows the name of the object.

Retouch Because (Defects)

Click Define to open a list into which you enter the reasons why this part is to be removed from the line. In our example only parts will be removed which have at least one of the two defects *defect_1* or *defect_2*. All other parts are transferred from station to station along the line. If you want to remove all defective parts from the line, you can enter `all` as the reason.



Passing Through Time

This time accrues, when parts come from the line and enter the line, i.e. when they are passed through only.

Feeding Out to Retouch Time

This time accrues, when parts come from the line and are moved to the retouch area, i.e. they are removed from the line.

Feeding In from Retouch

This time accrues, when parts come from the retouch area and enter the line, i.e. they are fed back into the line.

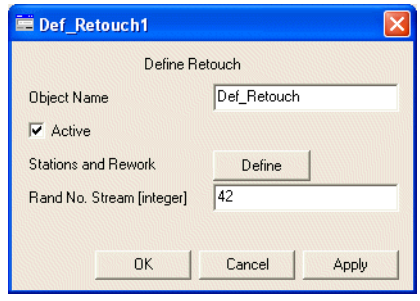
Attributes of the Object

objType: SingleprocRetouch

Def_Retouch

In the object *Def_Retouch* you can set at a single location which stations within the flow of materials cause retouching of MUs and which defects occur. The object marks the MUs in a way so that the objects *SingleprocRetouch and RetouchArea* recognize and retouch them. In addition at this time the required time for the respective repair will be calculated.

When parts are located on the Skids, these parts will be treated accordingly. Empty Skids are not marked for reworking. If several parts are located on a Skid, all parts on this Skid are marked for reworking.



Object Name

This text box shows the name of the object.

Active

Use this check box to activate or deactivate the object.

Stations and Rework

Click Define and enter the stations which create the defective parts as well as additional information about the defects.

	object 1	table 2
string	station	retouch because
1	~.sp1	reason
2	~.sp2	reason
3	~.sp3	reason

Enter the paths to the station which create defective parts into the cells of the column station Enter an identifier of your choice into the cells of the column retouch because. This creates a subtable. Enter the reasons which led to retouching, as well as the frequency and a distribution for the accruing retouch times.

	string 1	real 2	string 3	string 4	real 5	integer 6	real 7
string	reason	frequency	distribution	distribution parameter	cumulated	created til now	%
1	defect_1	40.00		25	40.00		
2	defect_2	40.00		35	80.00		

- reason: Enter the names of the retouch reasons (defects) into the cells of this column. These names have to be identical with the names which are uses in the object *RetouchArea* and *SingleProcRetouch* which do the actual retouching. You can enter any name of your choice.
- frequency: Enter the frequency with which the individual defects occur here.

Enter the associated retouch time into the next two columns:

- distribution: Enter the distribution for the retouch time here. Note that you can use any of the distribution functions which eM-Plant supports.
- distribution parameter: Enter the associated parameters of the distribution into these cells. Plant Simulation Assembly executes a method after closing the table, which checks the used distributions and the parameters for their plausibility.

The retouch reasons are considered to be cumulative and not as alternatives, i.e. when you define several retouch reasons for a station, the program always selects a single one. For this reason take care that the cumulated value in column 5 of the table does not exceed the value 100%.

The following columns only serve for informational purposes:

- cumulated: Shows the sum of the frequencies taken from the column frequency.
- Created until now: Shows the number of created defects.

Attributes of the Object

objType: Def_Retouch

Methods of the Object

Usage:
```
<path>.statistic(<table>);
<path>.statistic(<table_path>);
```

This method returns the statistical values of the Def_Retouch in a table. The table will be formatted by Plant Simulation. The content of the table is shown in the following picture

	string 1	string 2	real 3	integer 4	real 5
string	Station	Rework reason	Frequency	Created til now	% of parts
1	SP1	reason1	10	1281	40.7702100572884
2	SP1	reason2	15	1861	59.2297899427117
3	SP2	reason3	25	3168	100
4	SP3	reason4	15	1839	60.1373446697188
5	SP3	reason5	10	1219	39.8626553302812
6					

Description of the column values:

Station name: name of the station where the defect occurs

Rework reason: a namof the defect

Frequency: the frequency the defect occurs at this station

Created till now: number of parts marked with this defect

% of parts: the percentage of parts marked with this defect.

Note: The values of frequency and % of parts should be nearly the same.

5

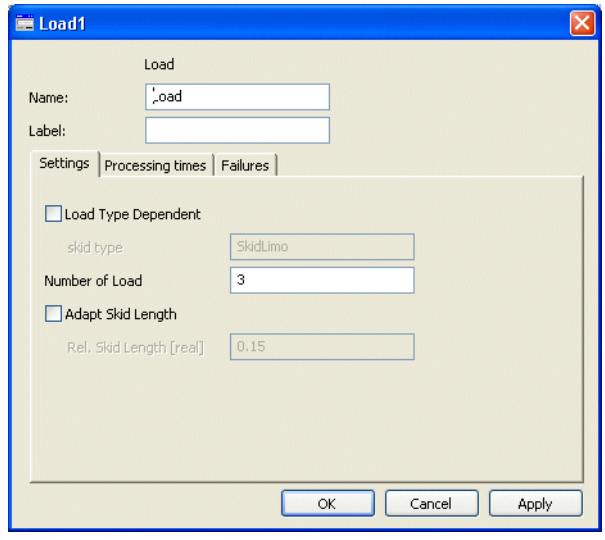
Skids

In this chapter we describe the objects which are required when using workpiece carriers. They encompass objects for loading and unloading workpiece carriers as well as objects for transferring workpiece carriers from one circle to the next.

Note: The object Load, Unload and Transfer could be replaced by the basic object TransferStation. But keep in mind that the transferStation has an other timing as the objects descriped here.

Load

You can use the object *Load* to load workpieces onto *Skids* (workpiece carriers). First the workpiece carrier has to be located in the station before the workpieces themselves can enter the station. You can define a capacity for the *Skids*, compare the setting Number adjustable in the *SourceN*. This way *Skids* can transport more than one part. The *Skids* can be loaded depending on the type. Then a *Load* station only loads a certain type of part. Other types of skids will just be passed through. The same holds true for occupied *Skids*. The object *Load* can accommodate one workpiece and one *Skid*.



Name:

This text box shows the name of the object.

Load Depending on the Type

Select this check box to only load a certain type of *Skid*. Clear it to load all types of *Skids*.

Skid Type

Enter the *Skid* type which you want to load in this station. This text box is active only, when you selected the check box Load depending on type.

Number of Load

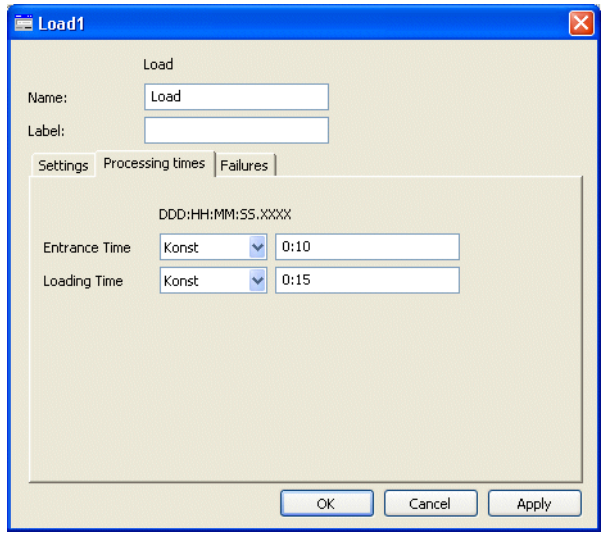
Enter the number of parts which you want to load onto the workpiece carrier. Naturally you have to make sure that the workpiece carrier provides the respective capacity.

Adapt Skid Length

Select this check box to adapt the length of the workpiece carrier to the length of the part. This is required when the workpiece carriers are located on a length-oriented material flow object and the loaded part is longer than the workpiece carrier. The part thus protrudes the workpiece carrier. To ensure that eM-Plant can exactly calculate the length, we recommend to adjust the length of the *Skid* to the length of the part.

Rel. Skid Length

Enter the difference in size between the workpiece carriers and the workpiece into this text box. Enter a positive value to make the *Skid* longer than the workpiece. You can also use the value you enter here to model the safety distance between two workpiece carriers. Note that the text box is active only when you select the check box Adapt skid length.

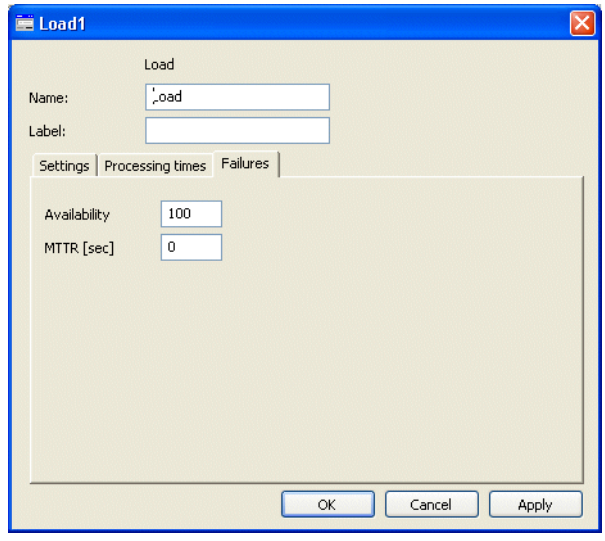


Entrance Time

Enter the entrance time the skid uses to enter the station.

Loading Time

Enter the loading time the skid uses to be loaded



Availability, MTTR

Enter the values for the Availability and the MTTR into the respective text boxes.

Attributes of the Object

objType: Load

Methods of the object

Statistic

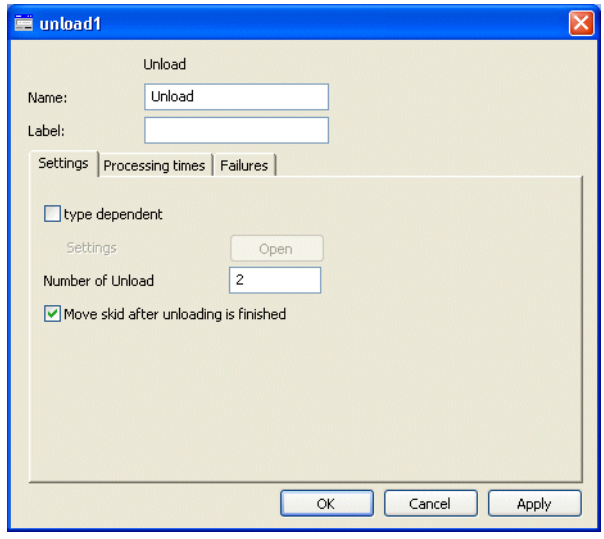
Usage:

```
<path>.statistic(<table>)
<path>.statistic(<table_path>)
```

Unload

You can use this object to unload workpieces from the *Skid*. After successfully unloading it, the now empty *Skid* is moved on. Empty *Skids* are passed through. A loaded *Skid* can only exit the object when either all workpieces or the defined number have been unloaded. When the unloaded workpieces are blocked this also blocks the Skid to be unloaded.

You can also select to only unload certain *Skid* types.



Name:

This text box shows the name of the object.

Type-dependent

Select this check box to make unloading the *Skid* depend on the part types. Click Open and enter the part type and the number of parts to be unloaded into the settings table.

Settings

This button is active only, when you select the check box Type-dependent.

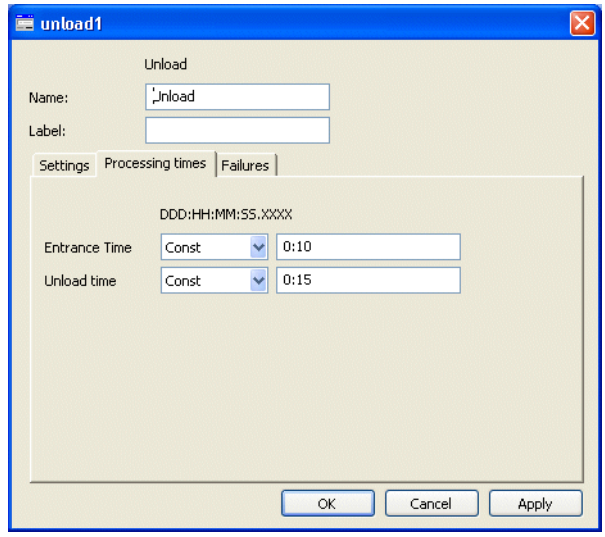
Enter the part type and the number of parts to be unloaded into the settings table

Enter the part type into the column type. Enter the number of parts of this type to be unloaded.

Number of Parts to be Unloaded

This button is active only, when you clear the check box Type-dependent.

Enter the amount of part to be unloaded from the workpiece carrier.

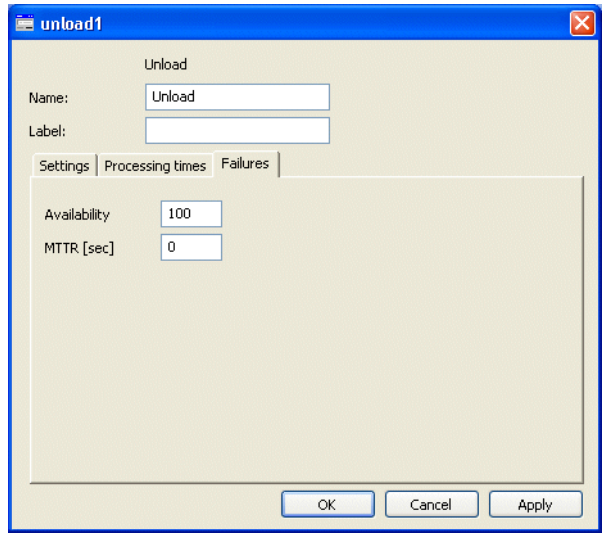


Entrance Time

Enter the entrance time the skid uses to enter the station.

Unloading Time

Enter the entrance time the skid uses to be unloaded.



Availability, MTTR

Enter the values for the Availability and the MTTR into the respective text boxes.

Attributes of the Object

objType: Unload

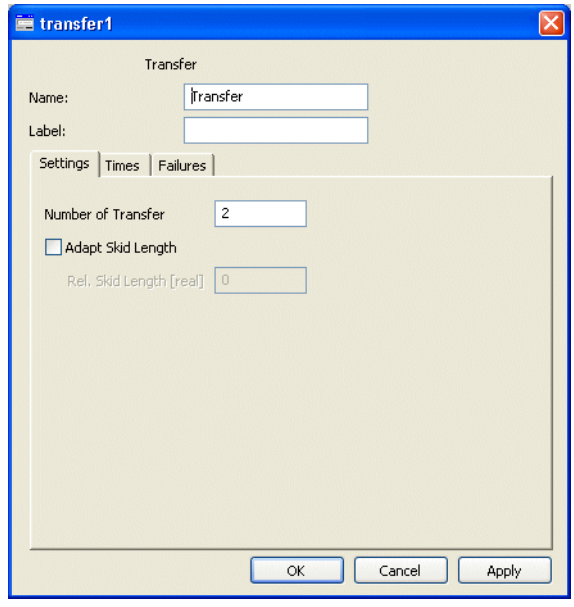
Methods of the object

Statistic

Usage:
```
<path>.statistic(<table>)
<path>.statistic(<table_path>)
```

Transfer

You can use this object to transfer parts from one workpiece carrier to another workpiece carrier which goes on a second tour through the line or section. Then the first empty workpiece carrier is passed on. Empty *Skids* are passed through. Parts are only transferred, when both workpiece carriers have entered the Transfer station in their entirety. The program ensures that only workpiece carrier each of each workpiece carrier cycle is located in the object *Transfer*. The workpiece carriers of both cycles can have different capacities. In this case one workpiece carrier exits the station as soon as it is empty and the other as soon as it is completely loaded.



Name:

This text box shows the name of the object.

Number of Transfers

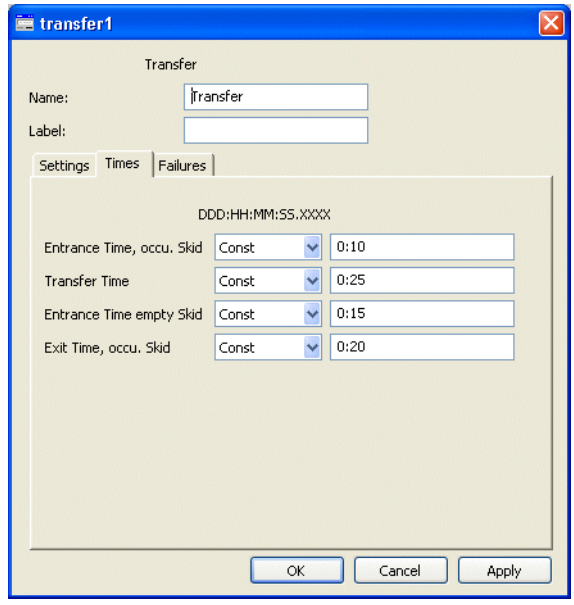
Enter how many parts are to be transferred into this text box. As a rule this is the number of parts the *Skid* to be loaded can carry.

Adapt Skid Length

Select this check box to adapt the length of the workpiece carrier to the length of the part. This is required when the workpiece carriers are located on length-oriented material flow object and the length of the workpiece carrier together with the workpiece depend on the size of the workpiece.

Rel. Skid Length

Enter the difference in size between the workpiece carriers and the workpiece into this text box. Enter a positive value to make the *Skid* longer than the workpiece. You can also use the value you enter here to model the safety distance between two workpiece carriers.



Entrance Time Full Skid

Enter the entrance time the skid uses to enter the station.

Transfer Time

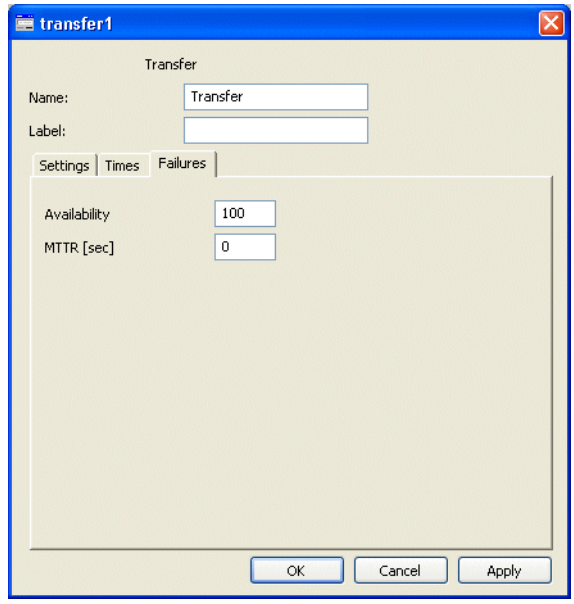
Enter the unloading time of the full skid.

Exit Time Full Skid

Enter the exit time of the loaded skid.

Entrance Time Empty Skid

Enter the entrance time of the empty skid.



Availability, MTTR

Enter the values for the Availability and the MTTR into the respective text boxes.

Attributes of the Object

objType: Transfer

6

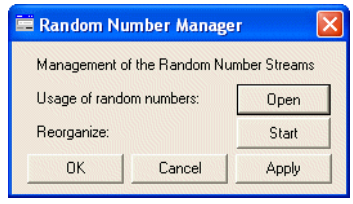
Tools

The objects described in this chapter serve for managing random number streams and for defining empirical distributions.

RandMgr

This object manages the random number streams for all user-defined objects.

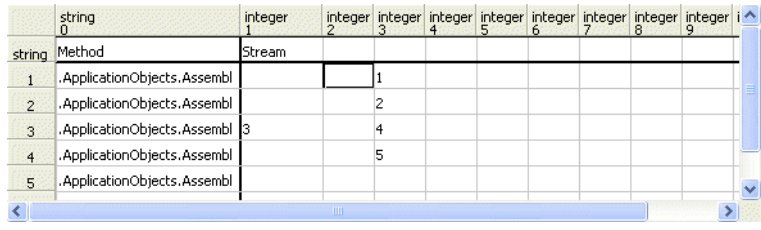
This ensures that all objects which you define use different random number streams and that they keep these even when you modify the structure of the model.



The object autonomously manages the random number streams. Click Open to view which object uses which stream.

The table lists the methods and object which were assigned random number streams by the *RandMgr*. Depending on how many streams an object needs, the table contains a different number of columns.

Existing assignments of streams to objects remain unchanged when other objects are added or deleted. This ensures that the random behavior of an object remains independent of any structural changes to your model.



	string 0	integer 1	integer 2	integer 3	integer 4	integer 5	integer 6	integer 7	integer 8	integer 9
string	Method	Stream								
1	.ApplicationObjects.Assembl			1						
2	.ApplicationObjects.Assembl			2						
3	.ApplicationObjects.Assembl	3		4						
4	.ApplicationObjects.Assembl			5						
5	.ApplicationObjects.Assembl									

Reorganize

Click Start to re-assign streams to objects. This makes sense, when you deleted a number of objects and want to re-assign streams which the now deleted objects used.

Attributes of the Object

objType: RandMgr

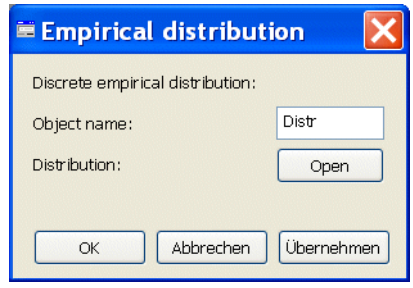
Methods of the Object

Usage: `<path>.getRandomStream(<obj>,<number>): integer;`

This method returns the number of the random number stream to the calling method (obj). The argument number sets which stream is used for this object. This makes sense then, when you need several different streams within a single method.

Distr

You can use the distribution object to model processing times with the discrete empirical distribution it contains. Note that you have to insert the object *Distr* into the top-level Frame, the root *Frame*, of your model.

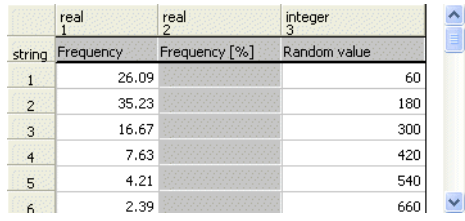


Object Name

This text box shows the name of the object.

Distribution

Click Open and enter the discrete empirical distribution into the table that opens.



	real 1	real 2	integer 3
string	Frequency	Frequency [%]	Random value
1	26.09		60
2	35.23		180
3	16.67		300
4	7.63		420
5	4.21		540
6	2.39		660

Enter the Frequency into the first column and the associated Value into the third column. The program norms the frequencies to 100% after you close the table.

Course of Action

To determine a frequency value, the program takes the value from the column Frequency and returns the associated value from the third column as result.

Usage

To use the empirical distribution, select Formula in the drop-down lists for the time of a built-in object and enter *<path_to_distribution>.calcite* as the processing time or as the formula.

Attributes of the Object

objType: Distr

Methods of the Object

Usage: `<path>.calcDistr(<stream>): real`

DistrReference

This distribution object realizes a discrete empirical distribution. Note that you have to insert the object *Distr_Reference* into the top-level *Frame*, the root *Frame*, of your model. You can use the object, when staffers work at stations and the processing times/cycle times fluctuate.

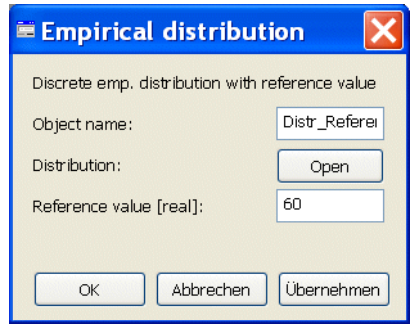
If you want to use the object, you have to enter a formula a the processing time.

As opposed to the object *Distr* the times in the distribution table refer to a reference value. When using this distribution a new reference value and the index of the random number stream are passed. First the program calculates a value from the distribution table according to the frequencies. The result of this calculation is then the new reference value multiplied with the quotient of the calculated value and the original reference value.

Return value = new reference value * (calculated value/original reference value)

You can thus use the object to model a shift of a known empirical distribution towards a new reference value. You will use this object when an empirical distribution in respect of a reference value is known, and when you want to use this same distribution but in respect of another reference value.

Example: `Suppose that the reference value is 60. When a mean value of 55 is passed and a value of 72 is calculated according to the defined frequencies, the return value yields: 55* (72/60) = 66`

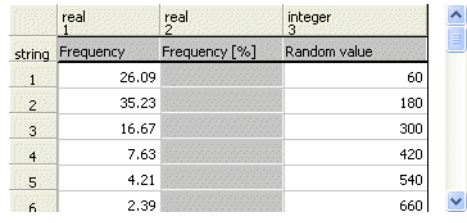


Object Name

This text box shows the name of the object.

Distribution

Click Open and enter the discrete empirical distribution into the table that opens.



	real 1	real 2	integer 3
string	Frequency	Frequency [%]	Random value
1	26.09		60
2	35.23		180
3	16.67		300
4	7.63		420
5	4.21		540
6	2.39		660

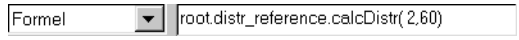
Enter the frequency of the value into the column Frequency. Enter the respective value into the column Random Value. Note that the values in the table refer to the reference value.

Reference Value

Enter the reference value of the distribution into this text box.

Usage

If you want to use this discrete empirical distribution, select Formula as the Processing time and enter the method call *calcDistr(stream, value)* into the text box of the processing time. Stream designates the index of the random number stream and Value designates the respective reference value for which the time is to be calculated.



Attributes of the Object

objType: Distr_Reference

Methods of the Object

Usage: `<path>.calcDistr(<stream>,<RefValue>) : real`

7

Evaluations

Plant Simulation Assembly provides graphical and numerical objects for evaluating your simulation runs.

Be aware that the results of numerical evaluations are available only at the end of a simulation run.

Statistics (Stats)

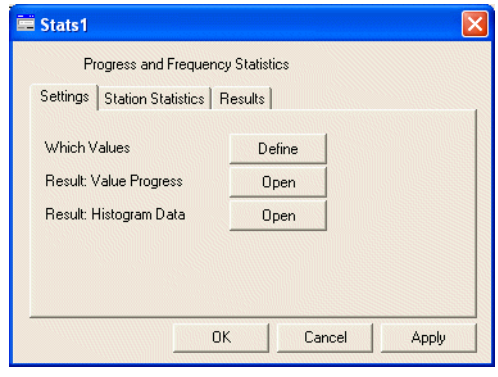
You can use the object *Stats* to centrally collect and manage the statistics of the material flow objects. In general you can choose between two types of statistics:

Progression statistics and frequency/histogram statistics. You define both in a table.

- Station statistics, which you define in the stations table.

Note: The results tables are filled only at the end of a simulation run. Enter the point in time at which the simulation run ends in the text box End on the tab Settings of the *EventController*.

When you use the *ExperimentManager* for executing several simulation runs, the program saves the results tables at the end of the individual simulation runs to disk.



Depending on which statistics you would like to record you have to, according to the type, enter a row into either the definition table or into the station table.

Progression and Frequency Statistics

The object can collect and process these values:

- Number of parts on the station. How long are how many parts located on the station.
- In-between arrival times of the parts at the respective station. Absolute points in time when the parts arrive.
- Interval between the arrival times of the parts at the respective station.
- Throughput times. Time intervals between creation of the parts until they arrive at the respective station.

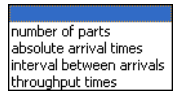
Tab Settings

Which Values Define

	string 1	object 2	string 3	boolean 4	real 5	string 6	integer 7	real 8	real 9
string	name	object	method	active	sample_Interval(sec	comment	step_widt	min_step	max_step
1	buffer	~.Buffer	number of parts	true	300.00	numMU of the buffer	1	0.00	10.00
2	S3_at	~.sp3	absolute arrival times	true		absolute arrival time	0	0.00	0.00
3	S3_iba	~.sp3	interval between arrivals	true		intervall between arr	30	0.00	300.00
4	S3_tpt	~.sp3	throughput times	true		throughput time	30	0.00	300.00
5									

The following columns of the definition table are filled in independent of the value type:

- name: The Histogram data table shows the name you enter here.
- object: Enter the path to the object whose statistics you want to record. (Enter the name of the respective built-in object here.)
- Method: Click in the cell in the column method and select which value will be collected for the object from the drop-down list:



  - number of parts: Number of parts on the station.
  - absolute arrival time: In-between arrival times of the parts at the respective station.
  - interval between arrivals: Interval between the arrivals of the parts at the respective station.
  - throughput time: Throughput times.
- active: In these cells you can activate or deactivate statistics collection without having to change any of the other settings in this row.
- sample interval: This is the interval for gathering data. You only need to enter a value, when you selected number of parts as the Method.
- comment: Enter any explanatory text here.
- step_width: Step size
- min_step: Minimal step
- max_step: Maximal step

Result: Value Progress

This table records all values which you defined in the definition table.

	string 1	list 2	string 3
string	name	list	comment
1	buffer	list21	numMU of the buffer
2	S3_at	list22	absolute arrival time
3	S3_iba	list23	intervall between arrival time
4	S3_tpt	list24	throughput time

The lists in the second column contains the individual values.

	real 1
1	0.00
2	0.00
3	0.00
4	0.00
5	0.00
6	0.00

Result: Histogram Data

	string 1	real 2	string 3	string 4	string 5	string 6	string 7	string 8	string 9	string 10	string 11	string 12	string 13
string	name	meanValue											
1													
2													
3	buffer		step_0	S1 - 1	S2 - 2	S3 - 3	S4 - 4	S5 - 5	S6 - 6	S7 - 7	S8 - 8	S9 - 9	step_ma
4		3.51	16.435	34.600	4.844	5.882	4.152	4.325	3.979	6.401	3.806	1.384	13.840
5													
6	deacivated												
7													
8													
9	S3_jba		step_0	S1 - 30	S31 - 60	S61 - 90	S91 - 120	S121 - 150	S151 - 180	S181 - 210	S211 - 240	S241 - 270	S271 - 2
10		89.82			91.624	0.312	0.416	0.416	0.624	0.520	0.676	0.208	0.520
11													
12	S3_tpt		step_0	S1 - 30	S31 - 60	S61 - 90	S91 - 120	S121 - 150	S151 - 180	S181 - 210	S211 - 240	S241 - 270	S271 - 2
13		572.49									31.876	3.276	0.936

This table records the processed data at the end of the simulation run.

- name: This column shows the name which you entered into the definition table.
- meanValue: This column shows the mean value of all values in this row.
- step_0, S1-1, ...: Percentage within this step.
- max_step: These columns show the percentage exceeding the maximal step.

Example: `We want to use these parameters:`

step_width=50, min_step=0, max_step=300

These settings create these steps:

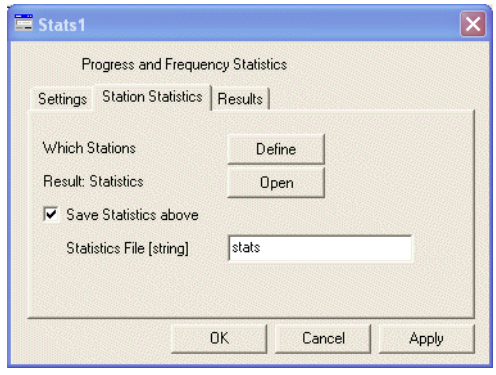
step1: 0

step2: 1 - 50

step3: 51 - 100

step4: 101 - 150 and so on until 300

Tab Stations Statistics



You can selectively collect the standard statistics of the material flow objects of the model. Just enter the names of the objects into the stations table for which standard statistics is collected.

Which Stations

Press button Define to open the station table.

	string 1	object 2	boolea 3	string 4
string	name	object	active	comment
1	SP1	~.sp1	true	Station 1
2	SP2	~.sp2	true	Station 2
3				

- name: Enter a name of your choice for the row.
- object: Enter the path to the object for which you would like to record standard statistics. (Enter the name of the respective built-in object here.)
- active: In these cells you can activate or deactivate statistics collection without having to change any of the other settings in this row.
- comment: Enter any explanatory text here.

Result: Statistics

The program writes the standard statistics of the objects, which you entered into the stations table, to this table at the end of the simulation run.

	string 1	time 2	time 3	integer 4	integer 5	integer 6	integer 7	integer 8	integer 9	real 10	real 11
string	Name	Zeit	Start(R)	Eingaenge	Ausgaenge	Inhalt	Vorbelegun	Min_Inh	Max_Inh	%_RelBel	%_RelBelUB
1	SP1	10:00:00:0	0.0000	8711	8710	1	0	0	1	99.99	99.99
2	SP2	10:00:00:0	0.0000	8710	8709	1	0	0	1	87.83	88.01

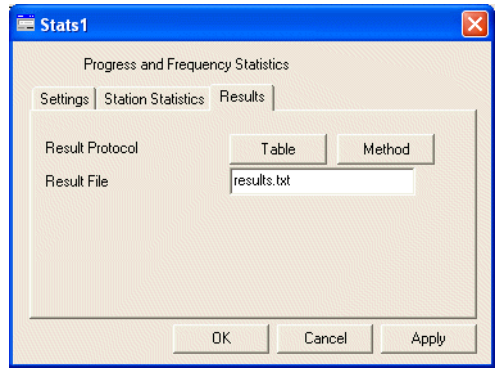
Save the Above Statistics

You can use this check box to save or to not save the statistics at the end of the simulation run.

Statistics File

Enter the name of the statistics file into this text box. For histogram data the program automatically adds the suffix *.his*. For the standard statistics file it adds *.mas*.

Tab Results



Result Protocol

At the end of the simulation run the program saves a results file, named results.txt, to disk. You can save values to this file, such as individual statistics values of critical stations.

Click Table and enter the respective object into the column object. Enter the attribute or the method of the object whose value you would like to record into the column Attribute.

You will, most likely, use some or all of these statistics methods: *statNumIn*, *statNumOut*, *statWaitingPortion*, *statWorkingPortion*, *statFailPortion*, *statBlockingPortion*, *statPausingPortion*, *statEmptyPortion*, *statNumMU*, *statRelativeOccupation*, *statRelativeOccupationIR*.

The statistics methods are described in the eM-Plant Help.

Results File

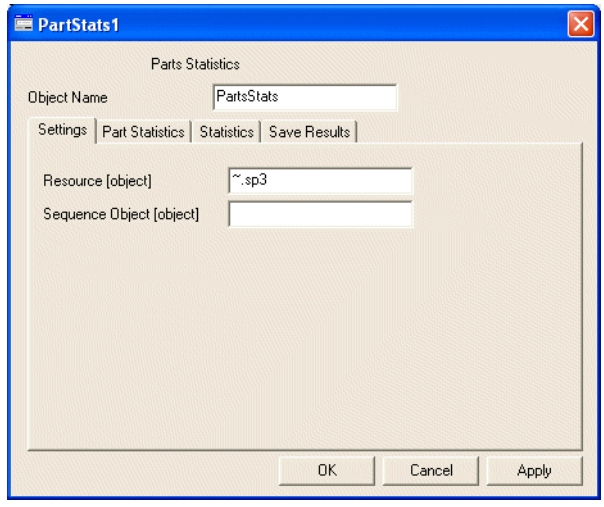
Enter the name of the results file into this text box.

PartsStats

The object PartsStats collects different statistics of the parts. You can attach it to any material flow object.

The object can record these values:

- Absolute Throughput
- Throughput per hour/per day
- Average Throughput Time
- Resource utilization, i.e. Production, Transport, Storage, until now.
- Recording of the throughput for individual hours/days.



Object Name

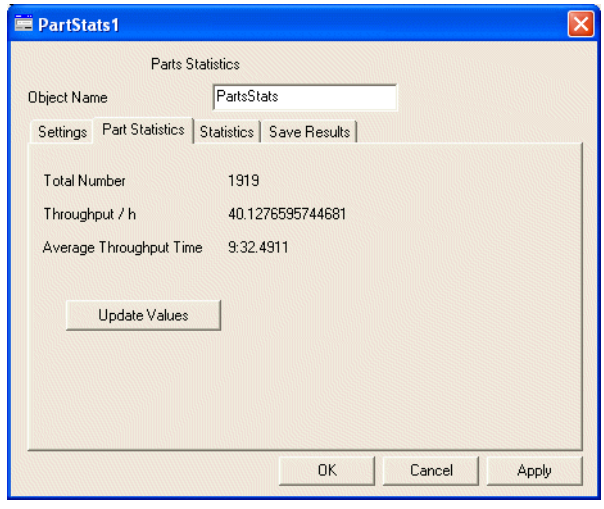
This text box shows the name of the object.

Tab Settings

Resource

Enter the name of the resource at which you would like to gather the parts statistics.

Tab Part Statistics



This tab shows the part statistics of the selected resource.

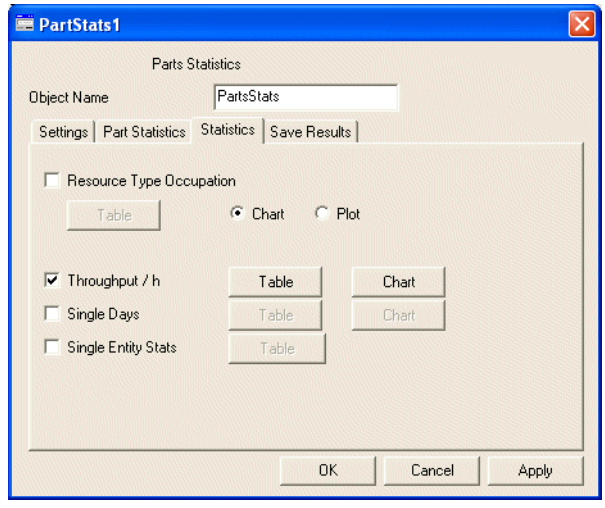
- Total Number: Number of part which passed the watched station.
- Throughput/h: Throughput of parts per work day.
- Average Throughput Time: Average throughput time of the entering parts.

Note: All displayed values are reset at a statistics start.

Update Values

Click Update Values to update the shown values.

Tab Statistics



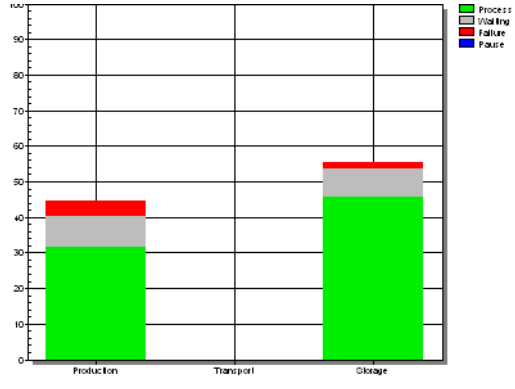
Resource Type Occupation

With this check box you can set if you want to record the resource occupancy of the parts until now. Remember that you can select the resource types Production, Transport or Storage in the dialogs of the material flow objects. In addition we record if a part was processed or moved on a resource, if the part waited because it was blocked or because the material flow object was failed or paused.

The program writes the results into a table or shows them in a *Chart*.

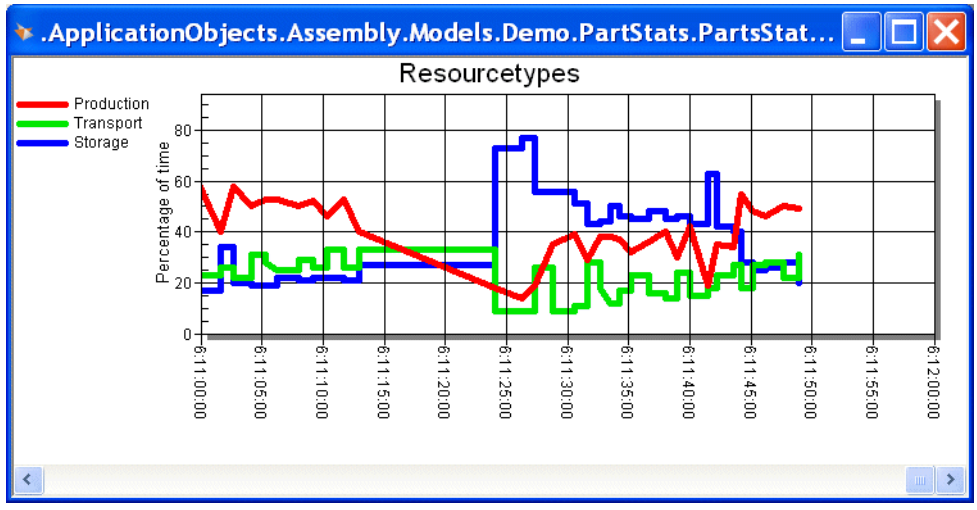
	string 0	real 1	real 2	real 3
string		Production	Transport	Storage
1	Process	31.44	0.00	45.65
2	Waiting	8.69	0.00	7.78
3	Failure	4.54	0.00	1.90
4	Pause	0.00	0.00	0.00

The column indexes of this table contains the resource types, the row indexes the states of the objects. In our example above the parts were located 15.98% of the statistics collection time on resources of type Production and were processed.



This chart shows the same values as the table shows as a graphic.

The plotted graphic above shows the occupancy of the resource types along the time axis.



Throughput/h

Select this check box to set if you want to save the throughput for each individual hour in a table or not.

Table: Throughput/h

This table records the throughput per hour of parts.

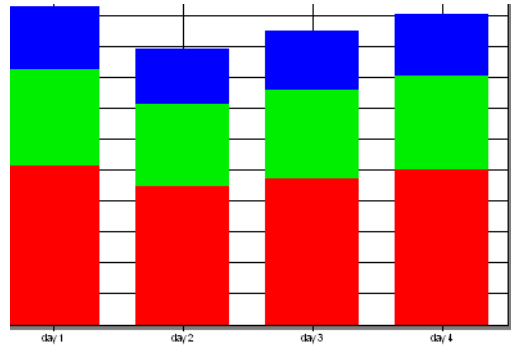
Single Days

Select this check box to set if you want to record for each day how many parts of which type passed the watched station.

The values are written to a table and can be visualized in a chart.

	string 0	integer 1	integer 2	integer 3	integer 4
string	Types	day 1	day 2	day 3	day 4
1	Total Cumulated		1919	2869	3873
2	Total per day	1028	891	950	1004
3	pickup	513	447	473	502
4	limo	310	265	285	302
5	coupe	205	179	192	200
6					

Create a column for each day. The first row shows the cumulated number of parts which passed through the watched station. The second row shows the number of parts per day. From row 3 on the program records the number of parts per day for each part type which passed through the watched resource.



The chart show the individual work days as stacked columns.

Single Entity Stats

If you want to protocol the times and type data of the individual parts in a table, select the check box Single Entity Stats.

Note: Be aware that recording the part statistics requires large amounts of RAM.

Table: Single Entity Stats

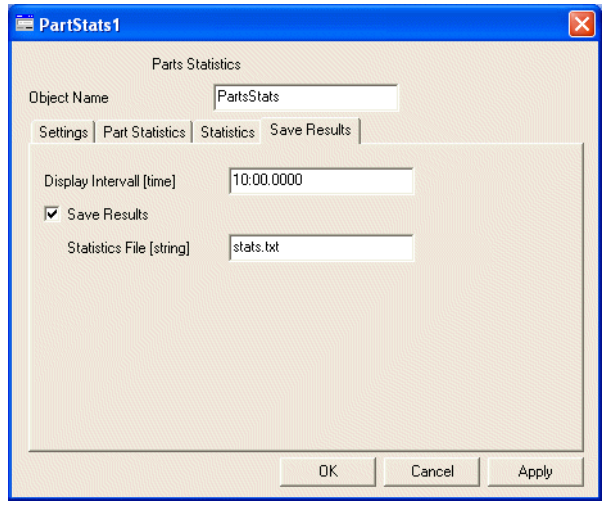
The table shows this information about the destroyed parts:

- Part type
- Sequence type
- Starting point in time of the part
- Finishing point in time of the part

Tab Save Results

Interval

Enter the display interval used for updating the display of the chart into this text box.



Save Results

Select this check box to save the results of the statistics at the end of the simulation experiment to disk.

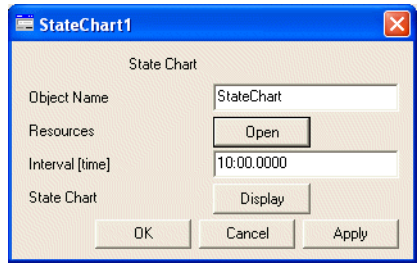
Statistics File

Enter the name of the statistics file into this text box.

StateChart

The object *StateChart* shows in a state chart how long the selected stations were in one of these states:

- Waiting: The station is waiting and is empty.
- Working: The station is working.
- Blocked: The station is occupied, but does not work any longer as the part cannot be moved on because the successor cannot take in any parts.
- Disrupted: The station is failed.
- Pause: The station is paused.

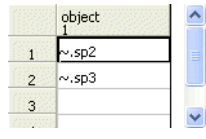


Object Name

This text box shows the name of the object.

Resources

Click Open and enter the stations whose values you would like to record into the table that opens.



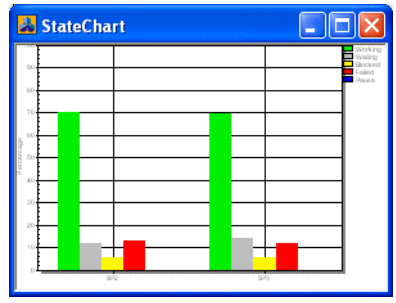
Note: The program checks the entries of the table after you close it. It transforms absolute paths into relative paths and, if required, adds *.workplace* to the name of the station. This enables dragging stations over the table and dropping them there. You can also drag the stations over the icon of the *StateChart* and drop it there.

Interval

Enter the display interval used for updating the display of the chart into this text box.

State Chart Display

Click Display to open a chart which shows the recorded values as a histogram.

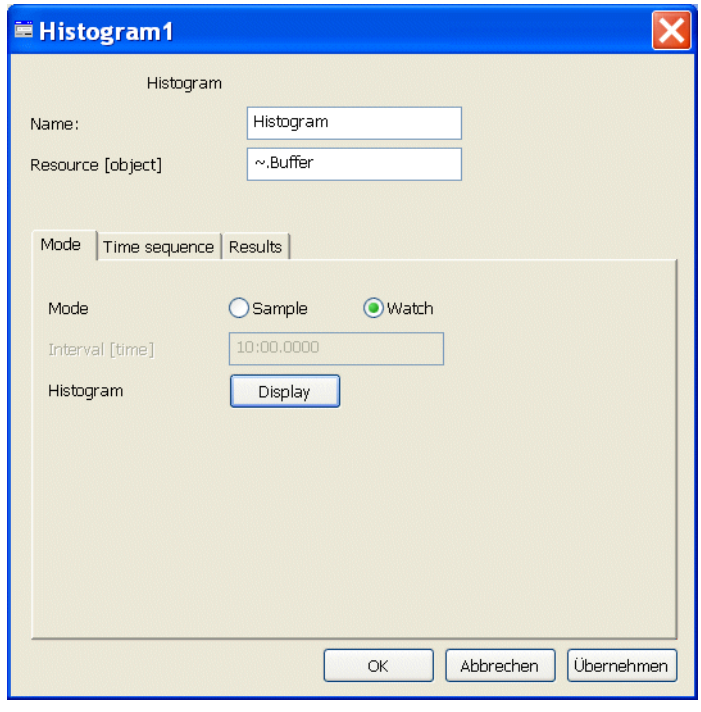


Note: This object requires a large amount of your computers resources. We thus recommend to only use it if you are going to show the values in the *Chart*. Only use the object to show a histogram of complex objects. If you want to just show a histogram of the built-in objects, you can use the built-in object *Chart*.

Note: Only use the object to show a histogram of complex objects. If you want to just show a histogram of the built-in objects, you can use the built-in object *Chart*.

Histogram

The object *Histogram* shows in a histogram which parts stayed for how long on a station.



Name:

This text box shows the name of the object.

Resource

Enter the object whose values you want to show into this text box.

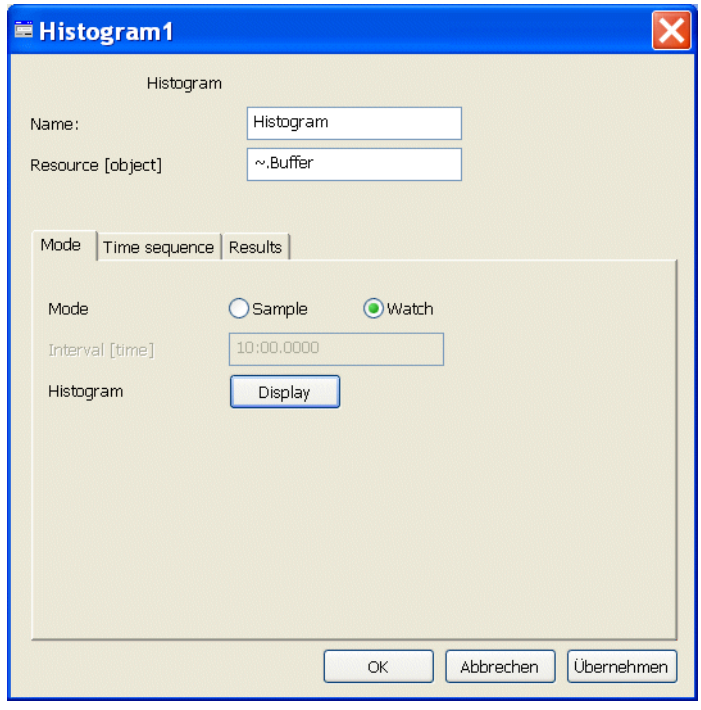
Note: If required, the program automatically appends *.workplace*. You can also drag the station over the icon of the *Histogram* and drop it there.

Mode

Use these radio buttons to set if data will be recorded in Watch or in Sample mode. In Watch mode each change of the occupancy is recorded immediately. The histogram is accurate. In Sample mode the object only records the occupancy during the interval which you enter. Sample mode has the advantages, that it uses up less processing time when the interval is reasonable. It also records the resource occupancy in fixed intervals.

The disadvantage of Sample Mode is: Peak occupancy between two sampled points in time are not recorded. The histogram thus becomes less accurate.

Tab Sample Mode



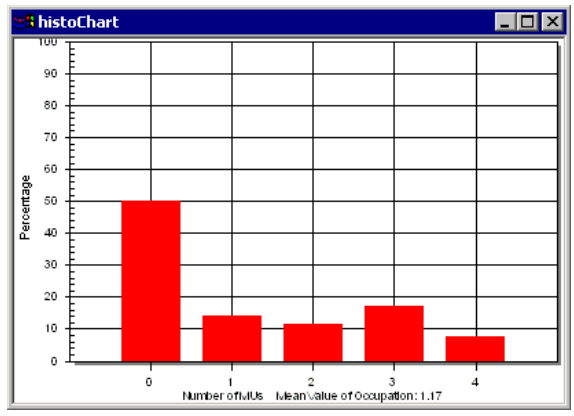
Interval

Enter the interval used for recording the occupancy of the resources into this text box.

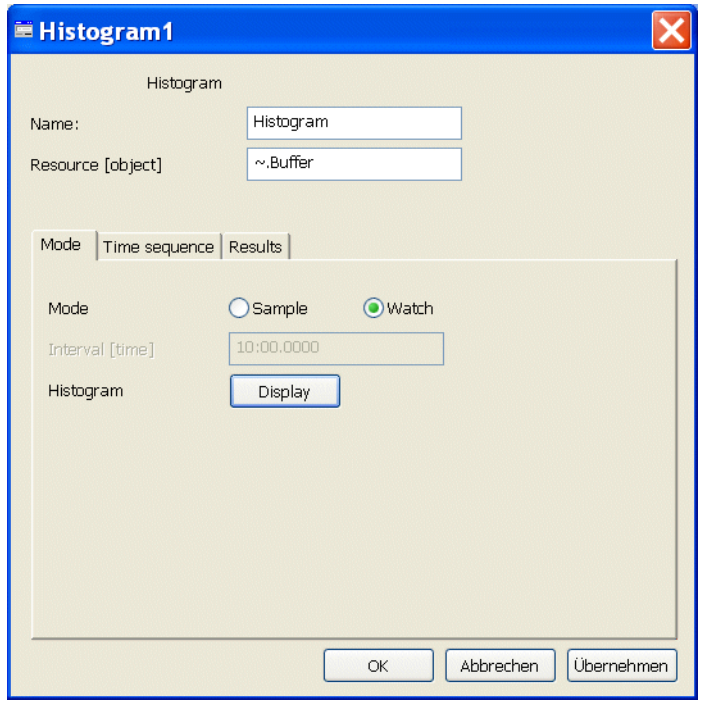
This text box is active only when you select Sample mode.

Histogram Display

Click Display to open a chart which shows the recorded values as a histogram.



Note: This object requires a large amount of your computers resources. We thus recommend to only use it if you are going to show the values in the *Chart*.



With Time Series

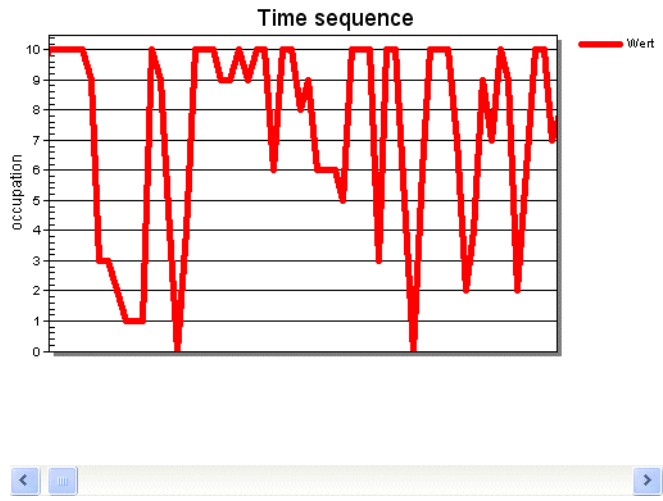
Use this check box to set if the resource allocation will also be displayed along the time axis. This text box is active only when you select Sample mode.

Interval TimeSequence

Enter the interval which the *TimeSequence* uses for recording the resource occupancy into this text box.

Plot

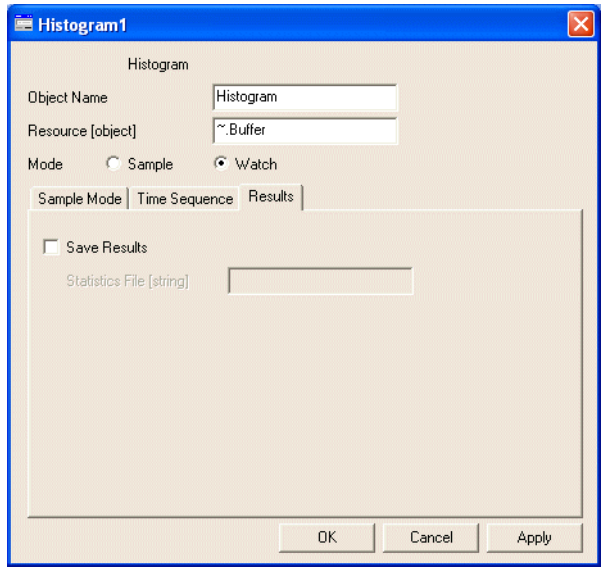
Click Display to show the *TimeSequence Chart*.



Display Interval

Enter the display interval used for updating the display of the chart into this text box.

Tab Results



Save Results

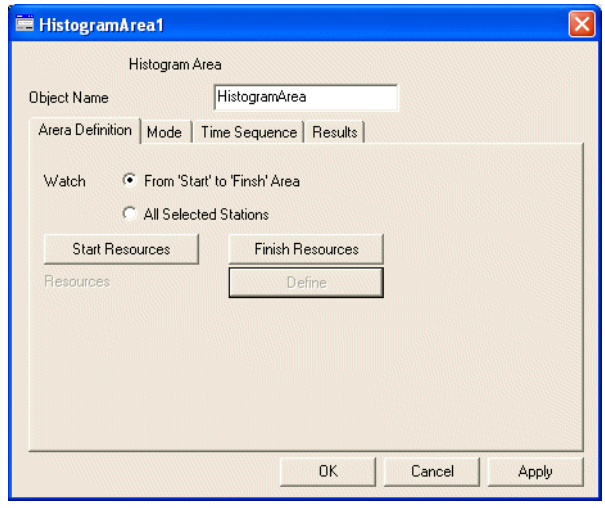
Select this check box to save the results of the statistics at the end of the simulation experiment to disk.

Statistics File

Enter the name of the statistics file into this text box.

HistogramArea

The *HistogramArea* records the occupancy of several stations and shows it in a histogram. The functionality of the *HistogramArea* is pretty much the same as that of the *Histogram*. The main difference is that you can record values of several material flow objects at the same time.



Object Name

This text box shows the name of the object.

Tab Area Definition

Watch

You can use these check boxes to set how to define the stations whose data the *HistogramArea* records.

- All Selected Stations: For this setting the program watches all stations which you enter into a list.
- From Start to Finish Area: For this setting the program watches the area between the Start and the Finish resources which you defined. To receive a correct evaluation of the occupancy, you have to make sure that you enter all stations into the start list, across which parts enter the area to be watched. Also make sure to enter all resources from which parts exit the area, into the Finish Resources list. This setting is especially helpful if you want to evaluate the occupancy of longer lines. In this case you only have to enter the one Start resource and the one Finish resource and watch it later.

Start Resources

This button is active only when you select Watch mode and the radio button From Start to Finish Area.

Click Define and enter all stations into this list across which parts enter the area to be watched.

End Resources

This button is active only when you select Watch mode and the radio button From Start to Finish Area.

Enter all stations into this list across which parts exit the area to be watched.

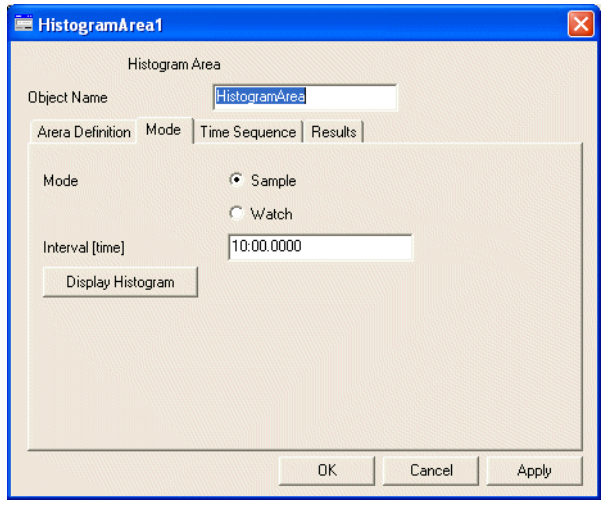
Resources

This dialog item is active only when you select Watch mode and the radio button All Selected Stations.

Click Define and enter the resources, whose values you would like to view, into the table that opens. When you insert *Frames*, which contain material flow objects, the HistogramArea considers all of these material flow objects.

Note: If required, the program automatically appends *.workplace*. You can also drag the station over the icon of the Histogram and drop it there.

Tab Mode



Mode

This dialog item is active only when you select Watch mode and the radio button All Selected Stations. When you select Watch mode and the radio button From Start to Finish Area, the program naturally employs Watch mode.

Use these radio buttons to set if data will be recorded in Watch or in Sample mode. In Watch mode each change of the occupancy is recorded immediately. The histogram is accurate. In Sample mode the object only records the occupancy during the interval which you enter. Sample mode has the advantages, that it uses up less processing time when the interval is reasonable. It also records the resource occupancy in fixed intervals.

Interval

Enter the interval used for recording the occupancy of the resources into this text box.

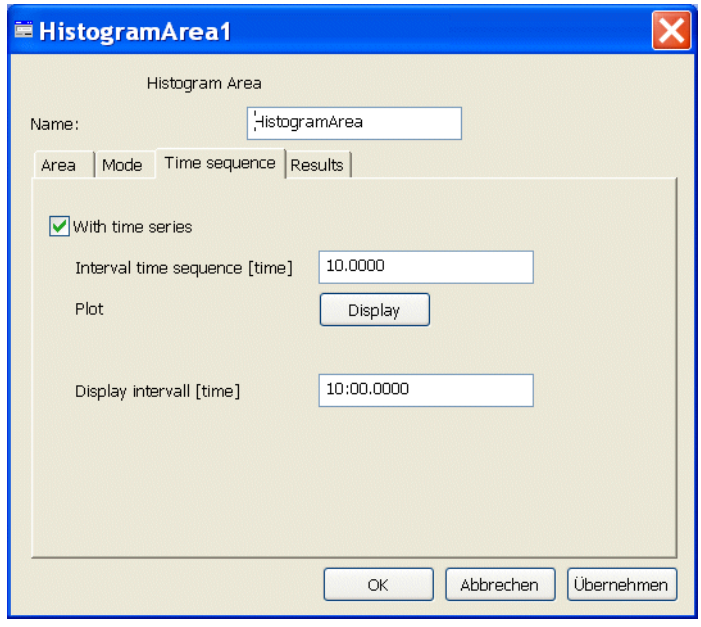
This text box is active only when you select Sample mode.

Histogram Display

Click Show Histogram to open the *Chart* which shows the recorded vales.

Note: This object requires a large amount of your computers resources. We thus recommend to only use it if you are going to show the values in the *Chart*.

Tab TimeSequence



With Time Series

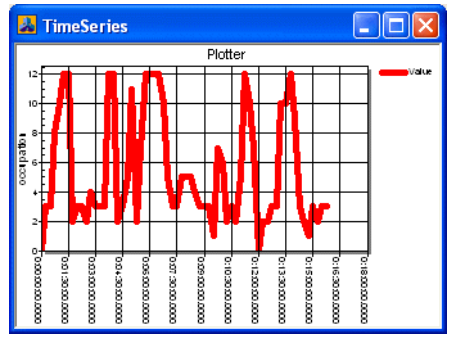
Use this check box to set if the resource allocation will also be displayed along the time axis. This text box is active only when you select Sample mode.

Interval TimeSequence

Enter the interval which the *TimeSequence* uses for recording the resource occupancy into this text box.

Plot

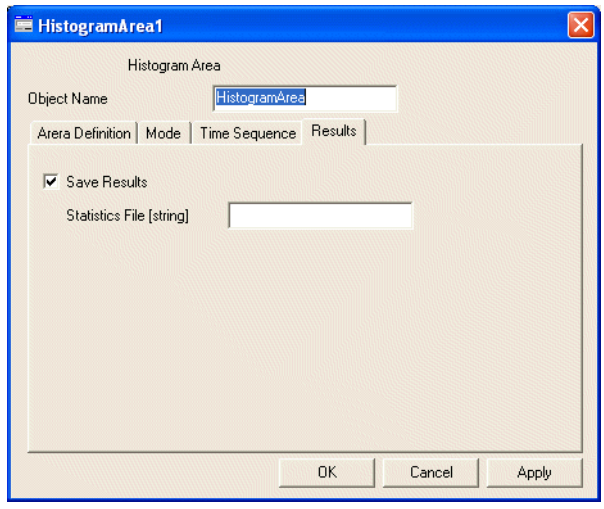
Click Display to open the *Chart* which shows the recorded values.



Display Interval

Enter the display interval used for updating the display of the *Chart* into this text box.

Tab Results



Save Results

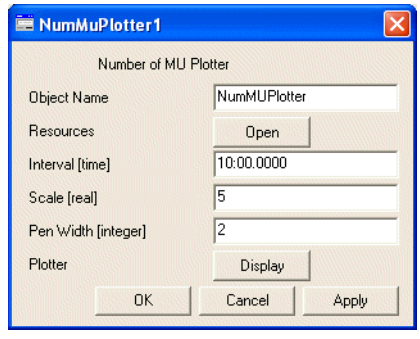
Select this check box to save the results of the statistics at the end of the simulation experiment to disk.

Statistics File

Enter the name of the statistics file into this text box.

NumMUPlotter

The *NumMUPlotter* plots the occupancy of resources, preferably of *Buffers*, over time.



Object Name

This text box shows the name of the object.

Resources

Click Open and enter the stations whose values you would like to record into the table that opens.

Note: The program checks the entries of the table after you close it. It transforms absolute paths into relative paths and, if required, adds *.workplace* to the name of the station. This enables dragging stations over the table and dropping them there. You can also drag the stations over the icon of the *NumMUPlotter* and drop it there.

Interval

Enter the display interval used for updating the display of the *Chart* into this text box.

Scale

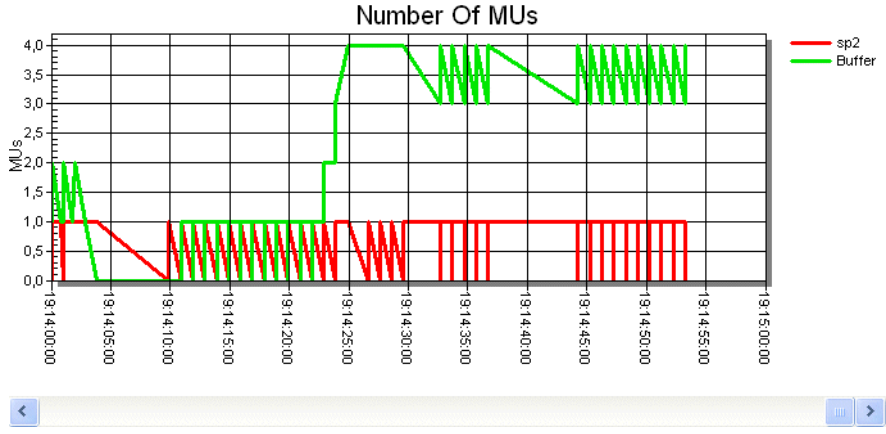
Enter the scaling factor which the *Plotter* uses into this text box.

Pen Width

Enter the pen width of the curves which the Plotter shows into this text box.

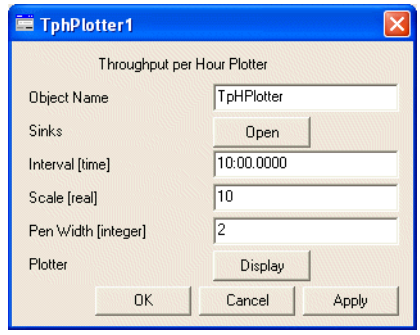
Display

Click Display to open the *Plotter*, which shows the recorded values as a histogram.



TpHPlotter

The *TpHPlotter* plots the throughput per hour along the time axis. The *TpHPlotter* visualizes values which the objects of type *Sink* record.



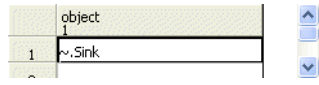
Object Name

This text box shows the name of the object.

Drains/Sinks

Click Open and enter the *Sink* objects, whose values you would like to record, into the table that opens.

Note: The program checks the entries of the table after you close it. It converts absolute paths into relative paths. This enables dragging stations over the table and dropping them there. You can also drag the stations over the icon of the *TpHPlotter* and drop it there.



Interval

Enter the display interval used for updating the display of the chart into this text box.

Scale

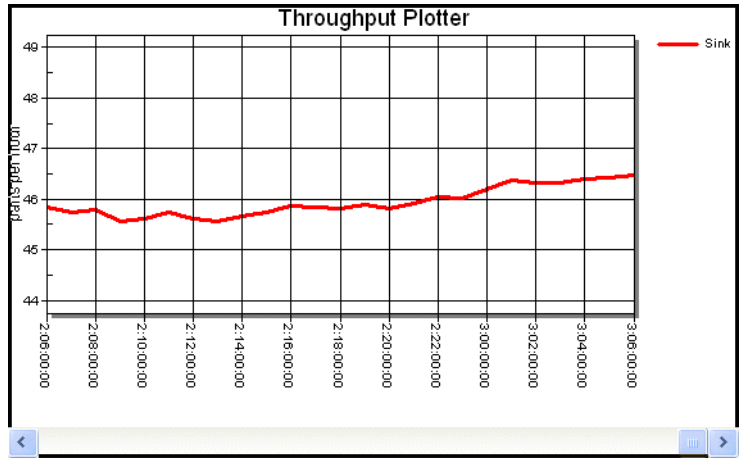
Enter the scaling factor which the *Plotter* uses into this text box.

Pen Width

Enter the pen width of the curves which the Plotter shows into this text box.

Display

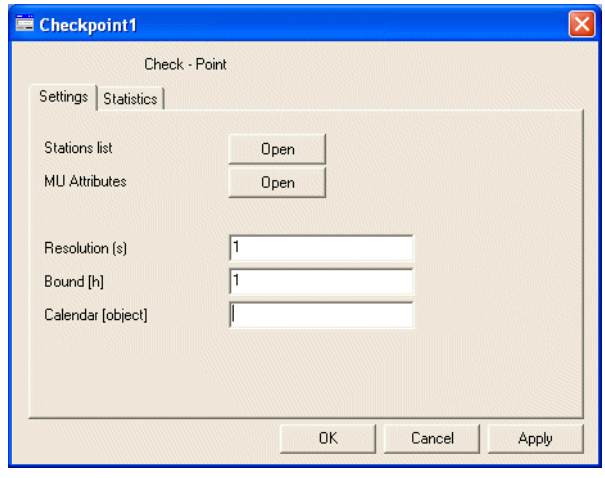
Click Display to open the *Plotter*, which shows the recorded values as a histogram.



Checkpoint

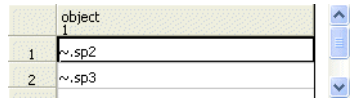
You can use the *Checkpoint* to define check points along a production line. These check points record state values, such as attributes and times, which will be evaluated at a later point in time. The check point record the points in time when the MUs arrived at the check point, the MU numbers used to visualize the random mix of MUs, the net time between the individual check points indicating processing times, and custom attributes which you can define.

Tab Settings



Stations List

Click Open and enter the stations, at which you would like to establish check points, into the table that opens. Note that the check points are located at the entrance of the station.



Note: The program checks the entries of the table after you close it. It transforms absolute paths into relative paths and, if required, adds *.workplace* to the name of the station. This enables dragging stations over the table and dropping them there.

MU Attributes

Click Open and enter the MU attributes which will be recorded at the check points.

Example: `Resolution`

Define the format of the histogram by entering its Resolution and its Bound.

Enter the step width of the histogram into the text box Resolution. Note that you enter seconds here.

Bound

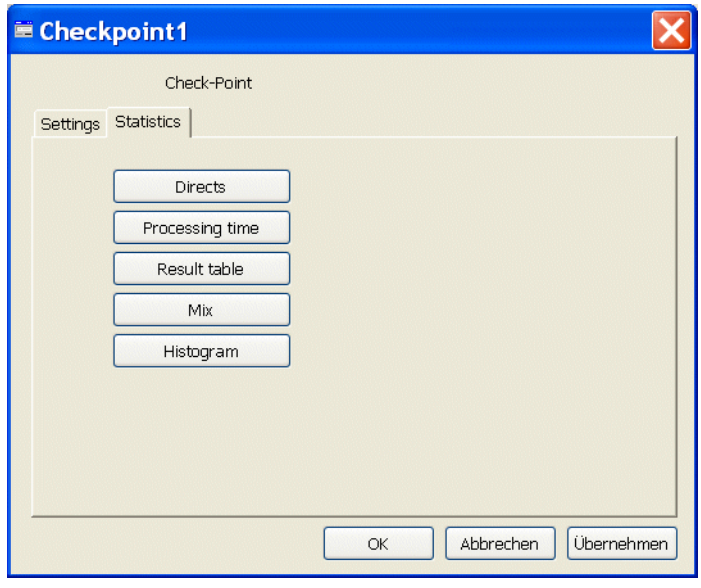
Enter the maximal value, which the *Histogram* records into this text box. Note that you enter hours here.

Calendar

Enter the *Calendar* object, which also contains the shift times of the watched stations, into this text box.

The program uses the *Calendar* for calculating the net processing times between the individual check points. The object *Checkpoint* deducts the pauses of the stations for this.

Tab Statistics



Directs

The program writes the names of all MUs, which passed a check point, into this table It also shows how often an MU passed a check point. This way you can easily detect which MUs pass through several areas and which parts are directs.

	integer 1	integer 2	integer 3	integer 4	integer 5	integer 6
string	CP: sp2 BE-No.	Appearance	CP: Buffer BE-No	Appearance	CP: SingleprocA	Appearance
1	1	1	1	1	1	1
2	2	1	2	1	2	1
3	3	1	3	1	3	1
4	4	1	4	1	4	1

The table contains two columns for each check point. The first column records the MU number the second the frequency with which the MU passed this check point. In the example above all MUs were directs. The respective columns appearance thus always show 1. For each resource you entered as a *Resource* a pair of columns is created.

Processing Time

For each checkpoint the difference in time of the arrival time will be calculated. Also the difference of the arrival time of the (n-1)nth part to the n-th part will be calculated. Please keep in mind that the sequence the parts are created must not be the same at the checkpoint. If a parts will arrive several times at a checkpoint, only the last arriving time ill be calculated.

	real 1	real 2	real 3
string	CP: sp2 (net)	CP: Buffer (net)	CP: SingleprocAttr (net)
1	60.00	53.70	64.34
2	64.34	50.43	60.00
3	60.00	81.95	60.00
4	60.00	59.10	73.09

Mix

The program writes the numbers of the MU, which pass this check point, into this table. Looking at this table shows the random mixing of the MU sequence.

	integer 1	integer 2	integer 3
string	CP: sp2	CP: Buffer	CP: SingleprocA
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6

The program records the MU number of each MU which passed the check point. Keep in mind: if a part will arrive several times at the checkpoint, only the last arrival time will be used.

Histogram

The program shows the processing times divided in classes, which you define, in this table. Define the format of the histogram by entering its Resolution and its Bound. Looking at this table shows the variance of the processing times.

	integer 1	integer 2	integer 3	integer 4	integer 5	integer 6	integer 7	integer 8
string	lower	< upper	CP: sp2 net.	CP: sp2 brt.	CP: Buffer net.	CP: Buffer brt.	CP: SingleprocA	CP: SingleprocA
1	0	10	589	558				
2	10	20						
3	20	30						
4	30	40						
5	40	50						
6	50	60						
7	60	70					462	434
8	70	80					93	91
9	80	90					15	15
10	90	100					2	2
11	100	110						
12	110	120						

The first two columns show class related information. The following columns show check point related data in two columns each. The first column records the number of net processing times located in the area of the respective class. The second column of them check point records the number of the gross processing times.

Report Table

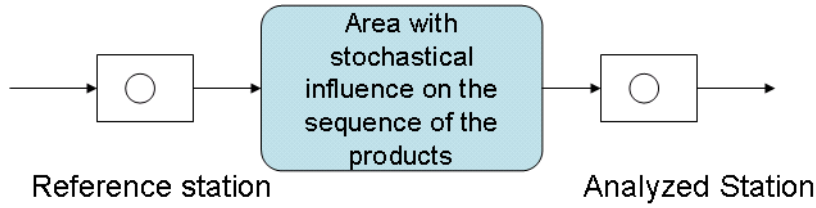
During the simulation run this table records all values required to fill in the tables described above. In addition it records the throughput times from creation to the last check point as net and as gross values. It also writes the values of the MU attributes which you defined at the check points into the subtables of the Report Table.

	string 1	real 2	table 3	time 4	time 5	time 6	table 7	time 8	table 9	string 10
string	BE	BE_Order	BE_StatAttr	Leadtime net.	Leadtime	CP: sp2	BE_DynAttr	CP: Buffer	BE_DynAttr	
1	AA	1	table31			2:00.0000	table71	4:00.0000	table91	
2	AA	2	table32			3:00.0000	table72	5:00.0000	table92	
3	AA	3	table33			4:00.0000	table73	6:00.0000	table93	
4	AA	4	table34			5:00.0000	table74	7:00.0000	table94	

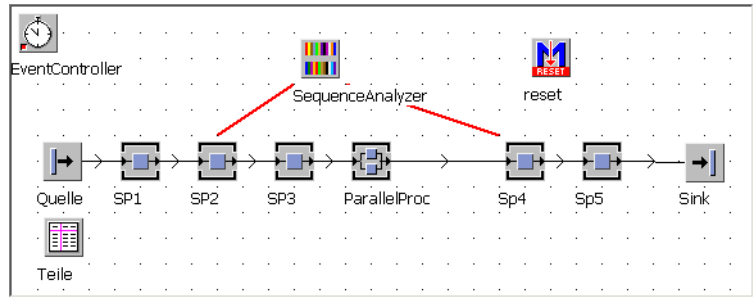
- MU: Name of the part.
- BE_Order: MU number at the first check point. This is the ID of the MU.
- BE_StatAttr: This subtable records the static MU attributes.
- Leadtime net.: Net throughput time from the point in time the MU was created to the last check point.
- Leadtime: Gross throughput time from the point in time the MU was created to the last check point.
- CP: <name>: Absolute point in time at which the MU arrived at the check point.
- BE_DynAttr: This subtable records the static MU attributes.

SequenceAnalyzer

You can use the object *SequenceAnalyzer* to compare a sequence of products (string of pearls) with a reference sequence and to show the differences. Within a production line a given string of pearls oftentimes will be mixed up by stochastic influences of the producing areas. The *SequenceAnalyzer* has to detect the degree of mixing and to provide statistical information about the reference sequence. To do so, you first have to select a reference station at which the reference sequence will be recorded. With a second you define the point at which the comparison sequence will be recorded.

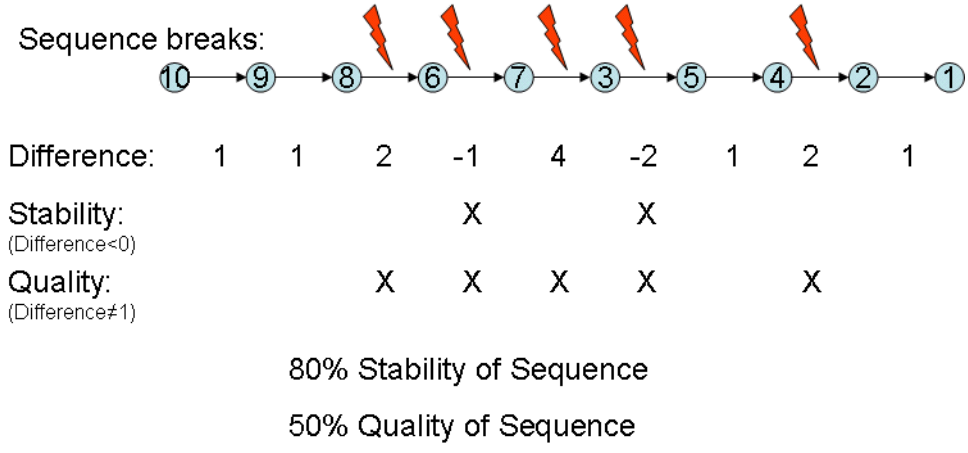


The *SequenceAnalyzer* returns two values: the quality of sequence and the stability of a sequence. Both values appear as absolute and as relative values. The following example illustrates the importance of these two values.



At the station SP2 arriving MUs are stamped with the number of their sequence of arrival. The station SP4 watches the sequence of arrival of the MUs at this station. We suppose that between these two stations an area is located, which changes the sequence with stochastic influences. In the model above the *ParallelProc* mixes up the sequence.

The picture below shows the sequence of the arriving products (vehicles). The numbers refer to the original sequence at the station SP2.



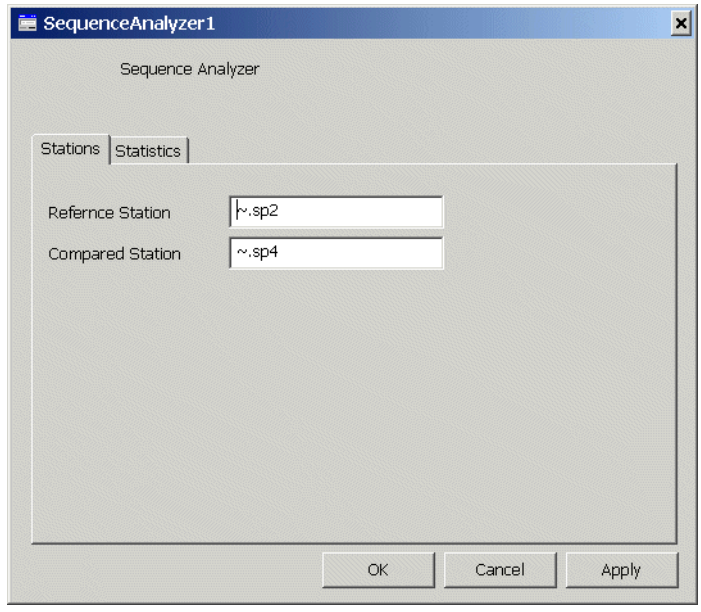
It shows that the watched sequence contains five sequence breaks altogether. Using the numbers of the original sequence, we establish the difference between the successor and the predecessor. For the stability of the sequence we now count the number of difference values, which are *smaller than zero*. For the quality of the sequence we count the number of difference values, which are *unequal to zero*. A difference value of one means that the products are in the original sequence, i.e. that it matches the original sequence.

Our example yields a stability value of two and a quality value of five. The quality thus defines the number of sequence breaks. The stability yields the value of how many products are offset within the sequence compared with the original sequence. In the example above these are the products 3 and 6.

You can insert the object *SequenceAnalyzer* anywhere in the model.

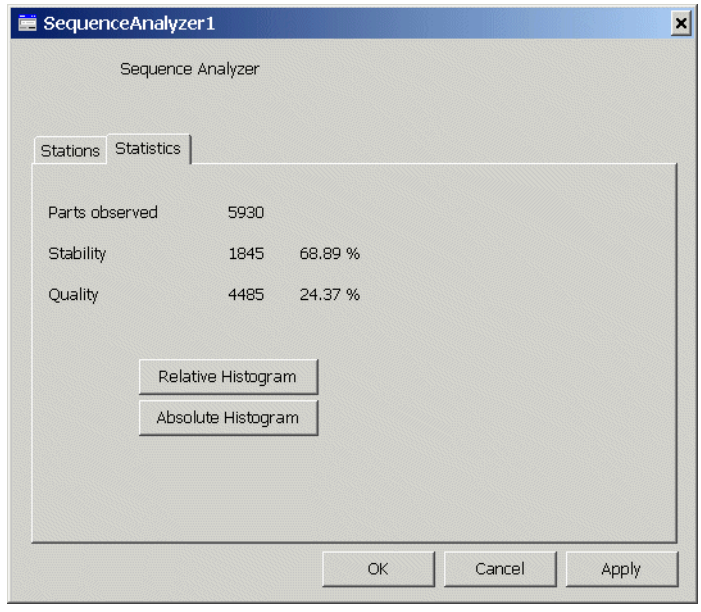
Tab Stations

On this tab you can enter the stations for watching the product sequences. You can also define the reference station and the compared station with drag-and-drop.



Tab Statistics

This tab shows the recorded and the calculated values.

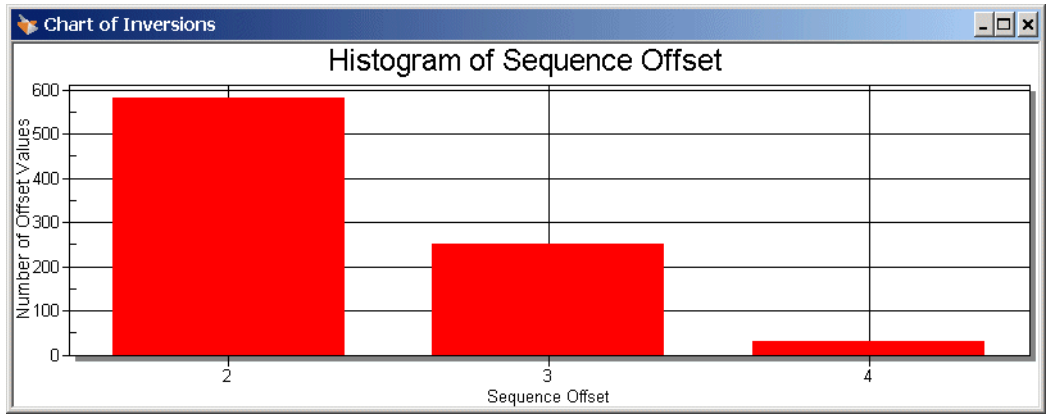


Watched parts: The number of watched parts. The *SequenceAnalyzer* only counts those parts, which passed both watched points.

Stability: Defines how many parts in the sequence were offset compared to the original sequence. It shows the number of sequence offsets, i.e. the number of difference values smaller than 0. The *SequenceAnalyzer* shows this value as the absolute number of parts as well as as a relative value based on the number of watched parts.

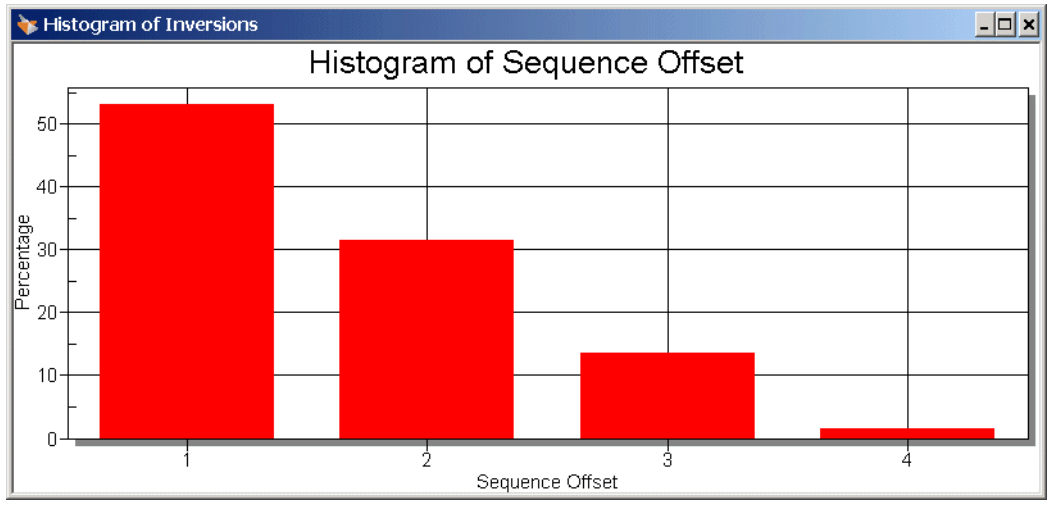
Quality: Defines the number of sequence breaks of the watched sequence compared with the original sequence. The *SequenceAnalyzer* shows this value as an absolute value as well as a relative value based on the number of watched parts.

Absolute Histogram: Clicking this button opens a histogram, which shows the number and the value of the offset.



Note: The chart shows the maximum offset of a product within the sequence. In our example, this is the value 4. Consequently a buffer with the capacity 4 suffices to return the sequence to the original sequence.

Relative Histogram: Clicking this button opens a histogram, which shows how often the offset value occurs based on the number of watched parts.



Attributes of the SequenceAnalyzer

objType: SequenceAnalyzer

Methods of the SequenceAnalyzer

Usage:

```
<path>.statistics(<table>);
<path>.statistics(<table_path>);
```

The method *statistics* returns the values on the tab Statistics and writes them to the table, which you pass as the argument <table>, or to the object of type table, which you pass as the argument <table_path>. The table will be correctly formatted.

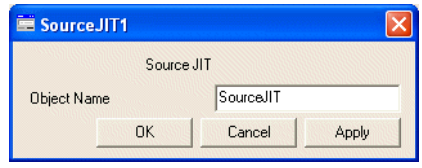
8

Just in Time (JIT)

For the just in time objects selected stations control how parts are to be produced.

SourceJIT

You will use the *SourceJIT* exclusively together with the *JIT* object, which controls when and how the *Sources* create parts.



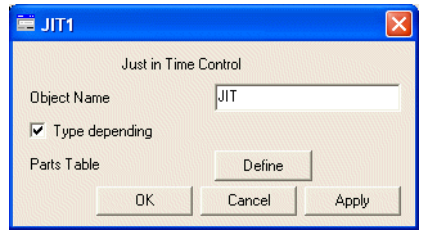
Object Name

This text box shows the name of the object.

Just in Time Control (JIT)

The object *JIT* controls the just in time creation of parts in the special *JIT sources*, compare *SourceJIT*.

This works as follows: When a part enters a station which you define, a JIT source starts creating parts. The *JIT sources* create the number of parts, which you enter, independent of the part type.



Object Name

This text box shows the name of the object.

Type-dependent

You can use this check box to set if the parts are created depending on the type of the part which initiates the creation process.

Parts Table

The parts table looks like this:

	object 0	table 1
string	station	parts
1		
2		
3		

- station: Enter the stations into this column which cause the objects of type *SourceJIT* to create parts.
- parts: Enter an identifier of your choice into the cells of this column. This creates a subtable. Enter the part types, together with the creation lists here, compare below.

	table 1	string 2	string 3	string 4	string 5
string	parts	type			
1					
2					
3					

- parts: Enter an identifier of your choice into the cells of this column. This creates a subtable. Enter the creation lists here, compare below.
- type: Into the columns 2 to n enter the names of the attributes into the column indexes. Enter the values which the attributes can take into the cells of these columns. Depending on which combination of values fits, the sources create the parts, which you entered into the column parts during the simulation run.

	object 1	object 2	string 3	integer 4
string	sourceJIT	entity	type	number
1				
2				
3				

- SourceJIT: Enter the *SourceJIT* objects here, which are going to create additional parts.
- entity: Enter the MU class of which you would like to create instances.
- Entitytype: Enter the names of the part types into the cells of this column.
- number: Enter the number of the respective parts.

Not Type Dependent

The parts table looks like this:

	object 0	table 1
string	station	parts
1		
2		
3		

- station: Enter the stations into this column which cause the objects of type *SourceJIT* to create parts.
- parts: Enter an identifier of your choice into the cells of this column. This creates a subtable. Enter the creation lists here, compare below.

	object 1	object 2	string 3	integer 4
string	sourceJIT	entity	type	number
1				
2				
3				

- SourceJIT: Enter the SourceJIT objects here, which are going to create additional parts.
- entity: Enter the MU class of which you would like to create instances.

- type: Enter the names of the part types into the cells of this column.
- number: Enter the number of the respective parts.

About Siemens PLM Software

Siemens PLM Software, a division of Siemens Automation and Drives (A&D), is a leading global provider of product lifecycle management (PLM) software and services with 4.6 million licensed seats and 51,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software's open enterprise solutions enable a world where organizations and their partners collaborate through Global Innovation Networks to deliver world-class products and services. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

SIEMENS

Division headquarters

United States
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
972 987 3000
Fax 972 987 3398

Regions

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
800 498 5351
Fax 972 987 3398

Europe
Norwich House Knoll Road
Camberley, Surrey
GU15 3SY
United Kingdom
44 1276 702000
Fax 44 1276 705150

Asia-Pacific
Suites 6804-8, 68/F., Central Plaza
18 Harbour Road, WanChai
Hong Kong
852 2230 3333
Fax 852 2230 3210

© 2007. Siemens Product Lifecycle Management Software Inc. All rights reserved. Siemens and the Siemens logo are registered trademarks of Siemens AG. Teamcenter, NX, Solid Edge, Tecnomatix, Parasolid, Femap, I-deas, JT, UGS Velocity Series, Geolus and the Signs of Innovation trade dress are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks used herein are the property of their respective holders.

10/07