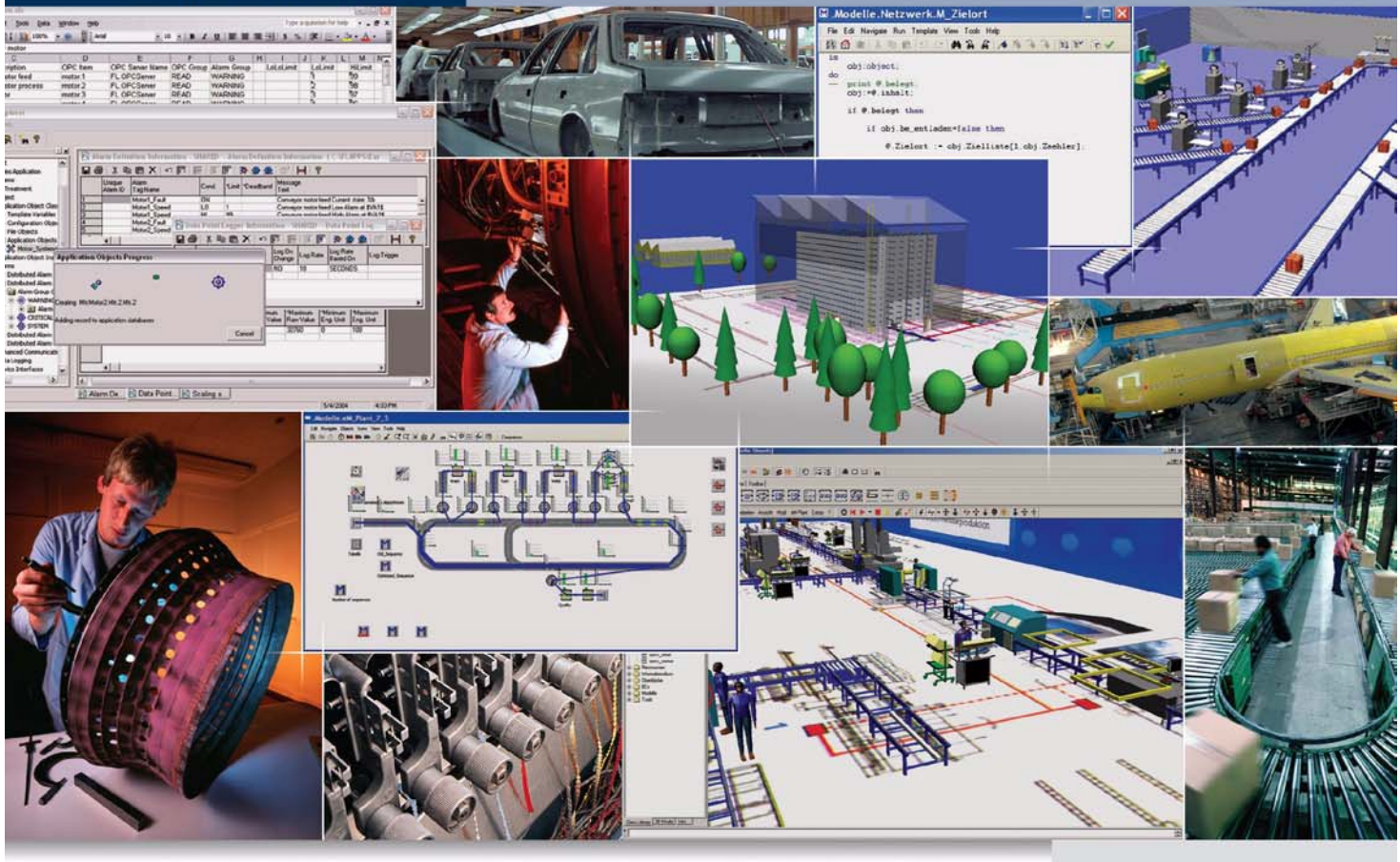
Plant Simulation Assembly library

Reference manual

Siemens PLM Software

www.siemens.com/plm



TECNOMATIX

SIEMENS

Proprietary and Restricted Rights Notice

TECNOMATIX Plant Simulation

SIEMENS

Plant Simulation

Transport Library

Version 9.0

December 2008

Table of Contents

Introduction

Plant Simulation Assembly is an application object library for modeling, simulating and evaluating assembly areas.

As opposed to previous versions, we divided the application object library into several modules in the present version. This is justified by the fact, that as a rule, you will only use the material flow objects in your project. If any of your projects require the extended functionalities, such as *FiniteStateMachine* or *Kanban*, you can always add these libraries at a later point in time to your model. This results in lean Assembly models which, in addition, require less memory. All sub-libraries are still an integral part of the Assembly license.

Part of restructuring the Assembly library also consisted in grouping objects with similar functions in the respective sub-folders. This makes it easier to extend or to append parts or functionalities of the library. You can use the method *updateClassLibrary* which Plant Simulation 7 provides, to quickly and easily update the individual modules of a library. Restructuring also adds to the clarity of the library and thus allows to find objects faster.

Below we describe the individual modules of the library. We describe the objects themselves as well as the changed functionalities as compared to previous versions. You will notice that in general the library makes increased use of the new features which Plant Simulation provides. We especially tried to migrate the original *Frames* which contained the built-in objects and the methods of the object, to basic objects whose functionality we extended with user-defined methods.

Changes in Version 7.1

In addition to a number of bug fixes, Plant Simulation Assembly Version 7.1 provides these changed/extended features.

SingleprocRetouch: You can now define, which error symptoms you would like to feed into the model. This way you can create error-specific repair stations.

RetouchArea: In addition to define the percentage of scrap, you can now also define error combinations which lead to scrapping the parts.

We added the objects *AssyWork*, *AssyTrack* and *AssyLineChart*. These names are intentionally close to the names of the existing objects *Assy_Work* and *Assy_Track*. The new objects are based on the built-in object *Track* and not modeled in a *Frame*. This way you can use the built-in curve mode to draw curved segments of the *Track*.

We adapted the *Assembly-Explorer* and the *PartManager* according to the new objects.

Changes in Version 7.5

In Version 7.5 we added the object *AssemblyType* to the library. Use it to quickly define an assembly process.

In addition we changed how to enter times into a number of objects. We also added the method statistics to some objects which you can use to write the statistics data shown in the dialog to a table. We describe the format of the table in the respective objects.

Now all objects have the attributes *AOLType* and *objType*. For eM-Plant Assembly the attribute *AOLType* generally has the value assembly. The attribute objType will be defined for each object. Note that both attributes are case-sensitive.

Changes in Version 7.6

In version 7.6 the KANBAN library was removed. These objects are now part of the basic objects of Plant Simulation.

We added the object *SequenceAnalyzer*. You can use it to analyze production sequences before and after mixing the sequence. It shows the degree of mixing. At the same time you can draw conclusions from the values of the *SequenceAnalyzer* about the buffer capacity of a succeeding sorter-buffer.

Changes in Version 8.1

In version 8.1 some minor bugs are fixed.. For some objects which are implemented in frames the new functionality of the inteface object was used and exit controls could be removed.

For the object SourceN there is a new setting which allowes you to reset the statistic data of MUs at the time the MU leaves the sourceN object. This can be used to remove the blocking time on the sourceN object from the lifetime of the MU.

Changes in Version 8.2

In version 8.2 only some minor bugs are fixed. For all objects the Localization folder was added. Please refer to the localization document to get more inforation how to localize the library.

The settings of the object *SourceN* was extended. It is possible to use the object *VaraintGenerator* to define the MUs and the attributes of the MUs. Please refer to the online manual to read more about the usage of the object *VariantGenerator*

The Library Modules

Transport

You can use the objects from this folder to simulate transport processes within the production.

Updating Existing Models

You can upgrade models, which you created in previous versions of Plant Simulation Assembly, at any time. First use the Update-Patches to update the model to eM-Plant Version 6. Then, load this model into eM-Plant 7 and use the command Save/Load > Update Class Library to update it.

Starting with versionn 7 of Plant Simulation you can update the model with the library modules by using Update Class Library.

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.

Features Planned for Future Versions

For future versions we plan to implement these features:

- The Method statistics for all material flow objects
- We plan to add an additional setting to the object *SourceN*, which allows you to create MU types according to any list/table. This way you might, for example, use existing production programs for creating the MUs.
- We plan to revise the *Kanban* module.
- We plan to revise the *Transport* module.
- We plan to add a *Logistics* module for transporting parts within the production.

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.

1

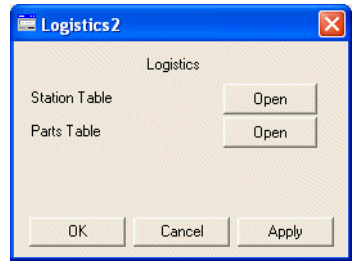
Logistics

You can use the objects described in this chapter to model simple transport processes within the production line.

Logistics

You can use the objects *Logistics*, *Transport* and *Shelf* to model the logistics area of your plant. The object *Logistics* serves for managing all data relating to logistics operations. You will use the object *Transport* to model the delivery of parts with the *Transporter*.

You can enter the parts required for processing for any number of stations into the tables of the object *Logistics*. It manages the parts, updates the number of parts located at the line and starts internal transport orders, when the order limit has been exceeded or stops the station when not enough parts to be processed are present. Note that the parts will not show up as MUs in your model but only appear in tables and are managed there too.



Stations Table

Enter all data relating to the stations and the parts they require into this table:

	object 0	string 1	integer 2	integer 4	integer 5	integer 6	time 7	time 8	boolean 9
string	station	partname	parts_buil	stock_line	orderlimit	changelimit	tripThere	returnTrip	attr dep log
1	.ApplicationObjects.Asse	AA	2	100	20	0	4:00.0000	2:00.0000	false
2	.ApplicationObjects.Asse	BB	3	100	20	0	4:00.0000	2:00.0000	false
3									

- station: Enter the path to the station into this column. If a station requires different types of parts, you have to enter it into the respective cells of the part types.
- partname: Enter the part into the cells of this column. You also must enter the part, which you enter here, into the Parts Table.
- parts_build: Enter the number of parts of this type which the station needs to process the main part, namely the MU.
- stock_line: This column shows the actual number of parts located at the station. You do not enter anything here, the program just displays the value.
- orderlimit: Enter the threshold value at which the station orders new parts. This activates the transport system which you entered into the parts table.
- changelimit: Enter the number of parts from which on the transport system passes the parts to the station. If this vale is not reached, when the transporter arrives, it waits for the time, which you define, and then leaves the station with the parts it was to deliver.

- tripThere: Enter the time which the transporter needs to get from its location to the station.
- returnTrip: Enter the time which the transporter needs to get from the station back to its location.

Note: When you use the *Logistics* object together with the *PartManager* and the assembly sections *Assy_Work* the parts are assembled depending on their type. Depending on the part types which are processed in the assembly section, the Logistics object deducts the assembled mounting parts. Remember that you entered into the *PartManager* which parts are assembled in which assembly section.

Parts Table

Enter into this table which transport system transports the part to the station. Also enter the bin size for transporting parts.

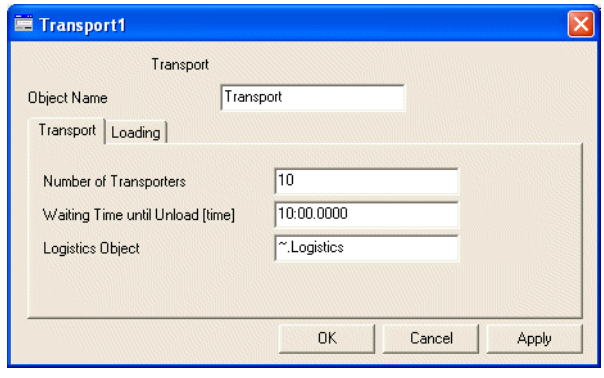
	string 0	integer 1	integer 2	string 3	string 4
string	partName	stockStore	binSize	transporter	station_Waiting
1	AA	0	10	tr1	
2	BB	0	10	tr1	
3					

- partname: Enter the name of the part into this column.
- stock_store: This column shows the inventory during the simulation run. This inventory is reduced each time parts are delivered. This value may fall below 0. The parts will be delivered anyway.
- binsize: Enter the bin size into the cells of this column. Parts are always transported to the station in a bin, meaning that the value you enter is the scope of supply.
- transporter: Enter means of transport of this part. You are going to model a means of transport with the object Transport, which you have to insert into the same *Frame* as the object *Logistics*. Enter the name of the object *Transport* here.

Transport 

You can use the object *Transport* to model the transport of parts from the object *Logistics*, which represents the warehouse, to the stations. You can also define the loading behavior of the *Transporter*. You can define how many bins the *Transporter* loads, how long it waits for additional order and how many bins the *Transporter* can load. Remember that you define how long it takes to travel from the warehouse to the various stations in the object *Logistics*.

Note: You can only use the object *Transport* in conjunction with the object *Logistics*, not together with the object *Shelf*.

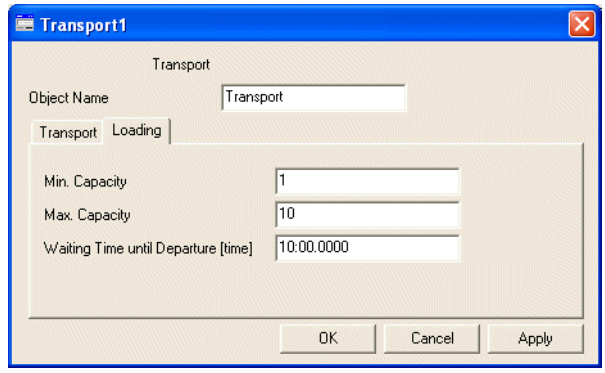


Object Name

Enter the name of the means of transport into this text box. Enter this name into the *Logistics* object, it represents the connection of the means of transport to the *Logistics* object.

Number of Transporters

Enter the number of *Transporters* into this text box. The object *Transport* can provide any number of transport elements.



Waiting Time Until Unloading

Enter the time which the *Transporter* waits at the station until it can unload its load. Enter the corresponding criteria into the *Logistics* object. The Transporter can unload its load when the stock reached the size you entered (stock_line <= changelimit).

Min. Capacity

Enter the number of bins, which have to be loaded onto the *Transporter*, before it drives to the stations. The *Transporter* also starts moving when fewer bins are loaded, when the Waiting time until departure has been reached.

Max. Capacity

Enter the maximum number of bins which the *Transporter* can load.

Waiting Time Until Departure

Enter the time after which the *Transporter* starts moving from the warehouse towards the stations. In this case the values you enter for the Minimal capacity and for the Maximal capacity have no meaning after this time has elapsed. The interval starts with the first order for this *Transporter*.

Logistics object

Enter the *Logistics* object the *Transport* object is connected with into this text box. The *Transporter* receives its transport orders from this Logistics object.

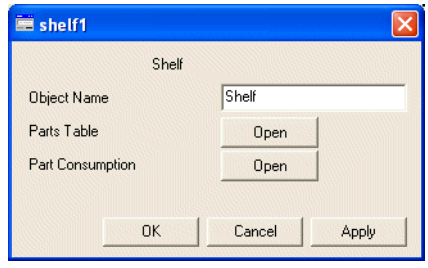
Shelf 

You can use the object *Shelf* to model a shelf storing parts. The stock is managed in a table. Note that no parts are physically moved. The object *Shelf* is intended to be mainly used together with the object *AssemblyAttr*.

The parts are assembled by the object *AssemblyAttr*. When the order limit has been reached it orders new parts from additional *Shelf* objects. When you did not enter any other object for supplying MUs, the current supply is increased immediately.

An *FSM* can transport the MUs from one shelf to the next. The objects communicate via signals.

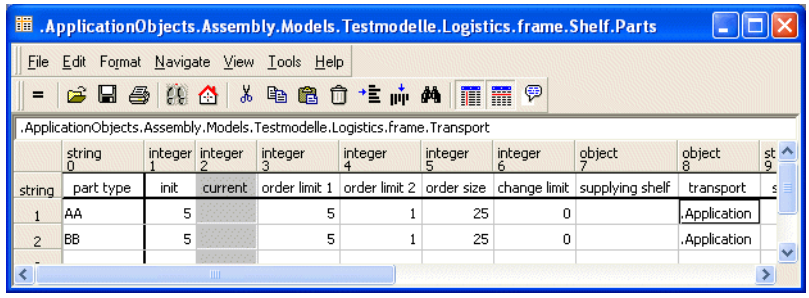
Modeling the parts in a table is beneficial when the total amount of parts is very high. In this case the model requires considerably less RAM than modeling the parts with objects would use up.



Object Name

This text box shows the name of the object.

Parts Table



This table manages the parts which can be stored in the warehouse.

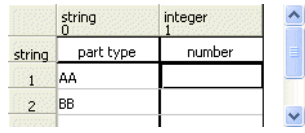
Enter this data into the columns:

- part type: Name of the part.
- init: Initial number of parts which will be entered at the beginning of the simulation run into the column current.
- current: Current stock, is administered by the *Shelf*.
- order limit 1: When the current stock reaches this value, an order is triggered for the object which you entered into the column supplying shelf. When you entered an *FSM* into the column transport, the program sends the signal you entered into the column signal 1 to it.
- order limit 2: When the current stock reaches this second value, an order is triggered for the object which you entered into the column supplying shelf. When you entered an *FSM* into the column transport, the program sends the signal you entered into the column signal 2 to it.
- order size: The size of the order. This is the number of parts which is going to be ordered.
- change limit: Enter the number of parts from which on the transport system passes the parts to the shelf. If this vale is not reached, when the *Transporter* arrives, it waits for the time, which you define, and then leaves the station with the parts it was to deliver.
- supplying shelf: Enter the *Shelf* or the *Logistics* object here, which provides the parts. When you do not enter anything here, the shelf will be restocked immediately.

- transport: Enter the object into this column which handles transporting the parts from *shelf* to *shelf* or from the *Logistics* object to the shelf. This can be a *Transport* object or an *FSM*. If the part is to be transported by an *FSM*, the *Shelf* and the *FSM* communicate via signals, which you enter into the following columns.
- signal 1: This signal will be sent to an *FSM* as soon as the current stock has reached order limit 1.
- signal 2: This signal will be sent to an *FSM* as soon as the current stock has reached order limit 2.
- signal wait: This is the signal for which the *Shelf* waits. As soon as it receives the signal, the *Shelf* increases the current stock by the amount of the *order size*.

Parts Consumption

This table records how many parts this shelf uses up.



The program enters the part type into the column part type. It enters number of stocked parts of this type into the column number.

Transporter

You can use this object to model the *Transporter* which the objects *Transport* and *Logistics* use. We already added the attributes which these objects need.

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

10/07