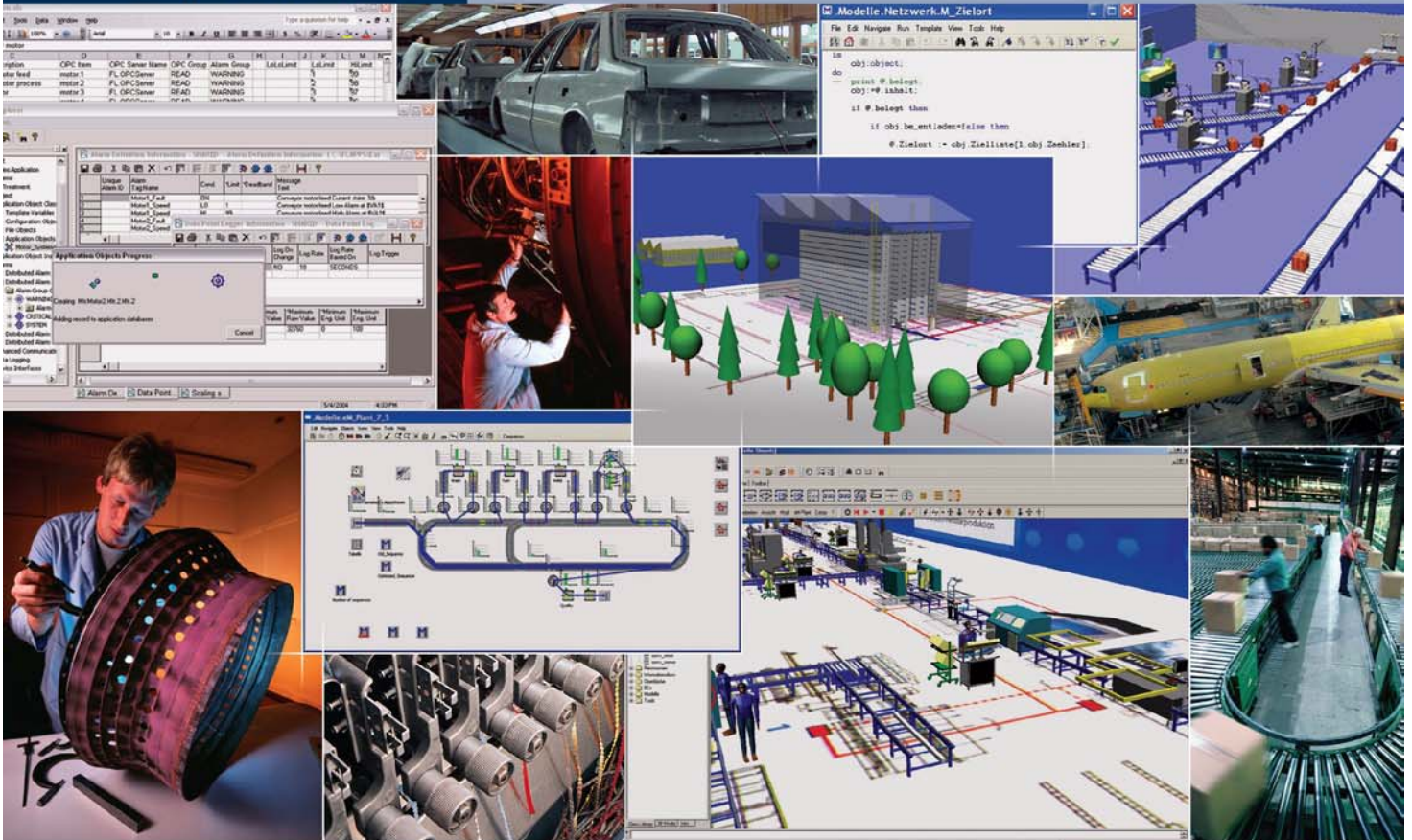


Plant Simulation Personnel library

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.



Plant Simulation

Personnel Library

Version 9.0

December 2008

Table of Contents

Personnel	1
Servicepool	1
Settings	1
Managing Service Stations	3
Evaluations	3
Exporter Evaluation	3
Service Evaluation	4
Station Evaluation	5
Methods of the ServicePool	5
addWorker	5
setWorkerAttribute	6
setWorkerAttribute	6
Service Station	6
Settings	7
The tab Objects	7
The tab Services	7
Methods of the ServiceStation	8
setServices	8
setServiceType	8
setMachine	9
setServicePool	9
Positioning	9

Personnel

Servicepool



Icon:

Use the *Servicepool* in connection with the *ServiceStations*. It manages the exporters and workers. These objects in Plant Simulation) provide services for the *ServiceStations*. The workers might, for example, provide types of workers described by their qualifying profile. An exporter can be used to model tools, like measuring instruments.

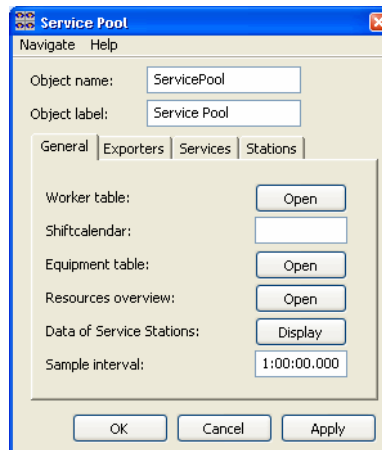
The *ServiceStations* attached to the *Servicepool* always request services, not the workers directly. The priority of the worker you set determines which worker the *Servicepool* assigns to provide the service.

Insert several instances of the *Servicepool* into your simulation model, to model several workspaces.

To reproduce shift models, you may attach each Exporter to a *Calendar*. The *Calendar* will then control the availability of the Exporter.

Settings

You may enter values for the following settings:



All settings for the *Service Pool* are done on the tab General

Object Name, Object Label: This data entry field holds the name of the object.

Exporter Table:

Define the worker in this table.
The left column contains the unique name.
Open the subtable in the column 'services' by the context menu of the cell.

	string 0	integer 1	integer 2	table 3	boolean 4	string 5
string	worker	number	priority	services	active	shift
1	Early	2	11	t	true	Early Shift
2	Late	2	12	t	true	Late Shift
3						
4						

Definition of workers with assigned services

Worker table

- **Worker:** Enter the names of the worker here. Each row represent a type of worker. Please note that the name must be a valid identifier of Plant Simulation.
- **number:** Enter the number of worker.
- **priority:** Enter the priority of the worker here. When the *Servicepool* receives a request for new services, it will use the exporter with the highest priority. (Naturally the exporter has to be able to provide this service and have available capacities.)
- **services:** Enter any identifier here to create a sublist. Enter the services the exporter is able to provide into this sublist.
- **active:** Enter *true* to activate a specific exporter, *false* to deactivate a specific exporter. This way you do not have to remove an entire line. If you do not enter any value then *true* is assumed.
- **shift:** If the workers is not available at all times, assign it to a time model that you define in a *Calendar* object. In the column **shift** you enter the name of the shift. If you leave this column empty, the worker is available at all times. The name of the shift is identical to the name of the shift in the calendar object which defines the shift times.

Schichtzeiten		Kalender		Ressourcen		Benutzerdefinierte Attribute					
	Schicht	Von	Bis	Mo	Di	Mi	Do	Fr	Sa	So	Pausen
1	EarlyShift	6:00	12:00	X	X	X	X	X	X		9:00-9:15;11:00-11:15
2	LateShift	13:00	17:00	X	X	X	X	X			15:00-15:30

Defining the shifts in the basic object *Calendar***Equipment table**

If the worker also needs tools, like measuring instruments, then you can define the available amount of equipments. The description of the columns is analogous to the description of **Worker table**.

Define the equipment in this table.
The column 'equipment' contains the unique name.

	string 0	integer 1	integer 2	boolean 3
string	Tool	number	priority	active
1	special Tool		2	1 true
2				
3				

Equipments and tools

Resources overview

- **Service:** This column contains all services which are needed in the service stations.

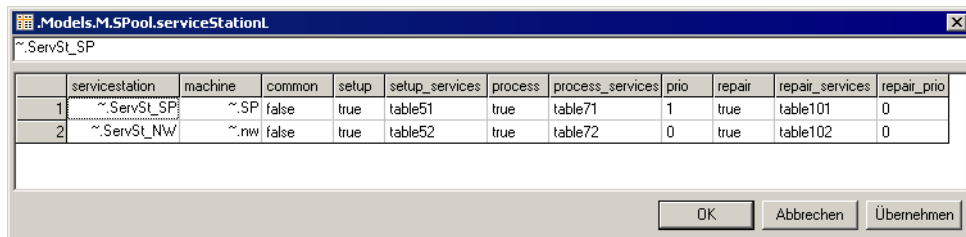
- **available amount:** Total capacity of all workers and tools, which offer the corresponding service.
- **max. needed amount:** Maximal amount of the service, which are needed in a service station.
- **Service Stations:** The subtable contains all service stations and the corresponding service type (*Setup, Process, Repair*), which needed the service.

This table can also be opened by the entry **Resources Overview** of the context menu.

Managing Service Stations

All *ServiceStations* that are connected to the *Servicepool* may be managed by it.

Data of Servicestations: The data of all connected *ServiceStations* is written to the administration table which you may open by clicking on this button.



	servicestation	machine	common	setup	setup_services	process	process_services	prio	repair	repair_services	repair_prio
1	~.ServSt_SP	~.SP	false	true	table51	true	table71	1	true	table101	0
2	~.ServSt_NW	~.nw	false	true	table52	true	table72	0	true	table102	0

The table file *serviceStationL*

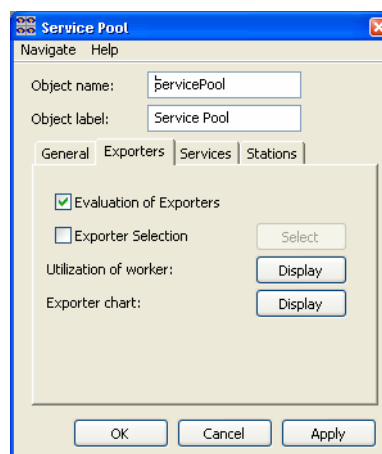
Sample Interval: Enter the interval in which the statistic data are calculated and the display of charts will be updated here.

Evaluations

There are evaluations for workers, used services and stations. For each evaluation there is a corresponding tab. In the first row there is a checkbox for activating or deactivating the evaluations. If an evaluation is not desired then all other dialog elements are inactive.

Exporter Evaluation

You may graphically display the statistical data that is relevant to the exporter. The object collects data about set-up, working, repair, waiting and non available (pause).



Service Pool

Navigate Help

Object name: servicePool

Object label: Service Pool

General Exporters Services Stations

☒ Evaluation of Exporters

☐ Exporter Selection Select

Utilization of worker: Display

Exporter chart: Display

OK Cancel Apply

Settings for exporter statistics

Exporter Selection

Use this button **Select** to set for which exporter data is graphically displayed: for all exporters, for selected exporters or for no exporters at all.

	string 1	boolean 2
string	exporter	eval active
1	exporterA	true
2	exporterB	true
3		

Select the exporters under consideration

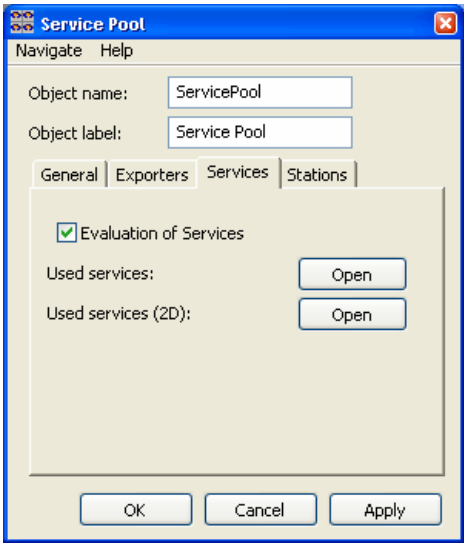
Enter the exporters whose data will be gathered.

Exporter Chart

Click on the button **Display** to show the state chart of the selected worker.

Service Evaluation

This object allows to record the services each worker provided, i.e. how often and for how long did it provide the service.



Settings for service statistics

Used Services: The object enters the workers that provided services during the simulation run:

	string 0	table 1
string	Exporter	Services
1	exporterA	table11
2	exporterB	table12
3		

It creates a line for each worker. The column **Services** contains a subtable that lists the services.

Used Services (2D): When you open the table *Used Services* that has several columns, the object converts it into a two-dimensional table:

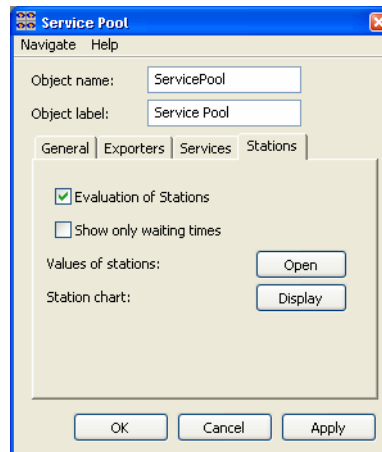
	string 1	string 2	integer 3	time 4
string	Exporter	Service	Number	Operating Time
1	exporterA	process_limo	34	8:17:36.0000
2	exporterA	help	12	3:05:04.0000
3	exporterA	simple_repair	25	1:05:53.8948
4				

The table provides these columns:

- **Exporter** This column holds the names of the exporters.
- **Service** holds the services.
- **Number** holds the amount of services provided.
- **Operating Time** holds the sum of the times for these services.

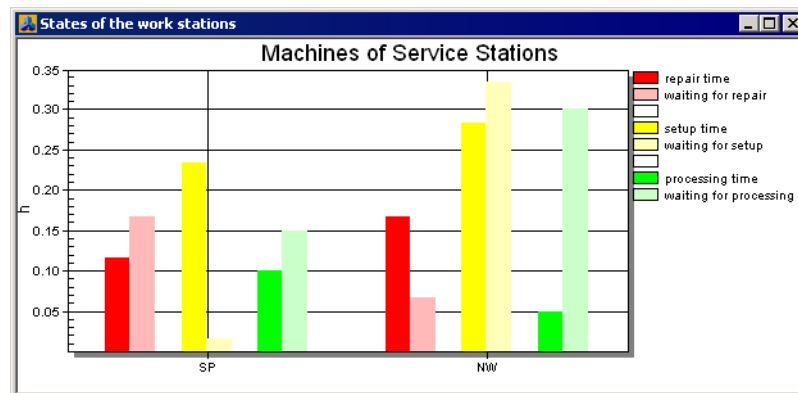
Station Evaluation

The object collects statistical data for all stations attached to *ServiceStations* of the *Servicepool*.



Settings for station statistics

Show only Waiting Times: If this checkbox is not selected then the object displays the time values for *Repair*, *Setup* and *Processing* together with its waiting time for the corresponding service.



Methods of the ServicePool

addWorker

Usage: `<path>.addWorker(name, amount, priority, services, active, shift)`

The method *addWorker* adds a worker (exporter) to the Servicepool designated by `<path>`. The method has the following parameter:

- **name:** Name of the exporter. This notation must be a valid name for an Plant Simulation object (type *string*).

- **amount:** Capacity of the exporter (type *integer*).
- **priority:** Priority of the exporter (type *integer*).
- **services:** Table consisting of one column, which contains the services of the exporter (type *table[string]*).
- **active:** Flag, indicating whether the exporter is active (true) or not (type *boolean*).
- **shift:** Name of the shift, in which the exporter is working. If you do not use shift, you enter an empty string. The name of the shift must defined in the calendar object. (type *string*).

setWorkerAttribute

Usage: <path>.setWorkerAttribute(Worker, AttributeName, AttributeValue): boolean


The method *setWorkerAttribute* set the value *AttributeValue* of the attribute *AttributeName* for the object *Worker*. The type of the attribute must be string. If the worker is not equipped with the attribute, then the attribute is created.

setWorkerAttribute

Usage: <path>.getWorkerAttribute(Worker, AttributeName): AttributeValue


The method *getWorkerAttribute* checks, whether the worker is defined. If the worker is defined then it checks, whether the worker is equipped with the attribute. If so the method returns the current value. If neither the worker nor the attribute are defined an empty string is returned.

Service Station


Icon: 

Attach the object *ServiceStation* to any station, for example an assembly station or a SingleProc. The *ServiceStation* controls the properties for importing services of the material flow object.

Services required for processing parts or for removing failures are requested from the *Servicepool*. The *Servicepool* the at-tempts to assign a worker providing that service to the station requesting it. Then processing starts or failures are re-moved. If none of the services required is available at the time, the station switches to the state *Waiting for Importer*. At request time you may order several groups of services. The *ServiceStation* assigns the group that is available first to the material flow.

During the simulation the icon of the *ServiceStation* shows the current state: The colors red, yellow and green symbolize repair, setup and work, respectively. Small flags show the ordering of the corresponding service .

At an arbitrary point of time you can open a table with all satisfied and non-satisfied importer by the context menu entry **Satisfied Importer**.

.Models.F.ServicePool.SatisfiedImporters			
Datei Bearbeiten Format Navigieren Ansicht Extras ?			
			
	object	string	boolean
	1	2	3
string	Importer	Type	Satisfied
1	~.SP	repair	true
2	~.SP	work	true
3			
4			
5			

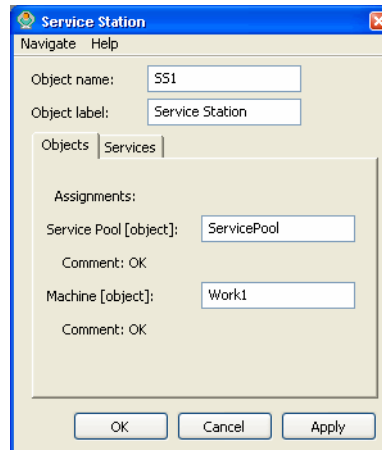
Satisfied and non-satisfied importer

Settings

Object Name, Object Label: This data entry field holds the name or the label of the object, respectively.

The tab Objects

You may enter the assignments of all objects of the library Plant Simulation personnel:



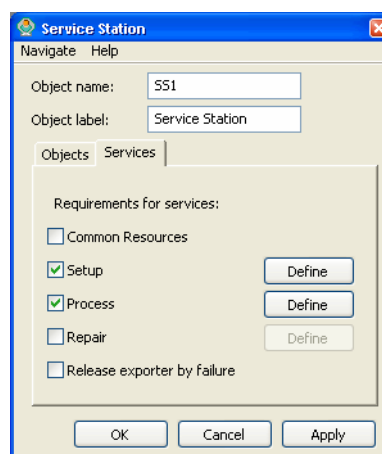
Service Pool [object]: Enter the path to the *Servicepool* which assigns the services to the *ServiceStation* here. Instead of entering the name of the *Servicepool* object into the data entry field you may also drag and drop it to the icon of the *ServiceStation*.

Machine [object]: Enter the path to the destination object the **ServiceStation** is going to be connected to here. It is the so-called importer for the services.

A relative path you enter has to begin with *location* (~) as you enter data within the station. Instead of entering the name of the *Station* object into the data entry field you may also drag and drop it to the icon of the *ServiceStation*.

Please note that after pressing the button **Apply** the **Comment** shows you whether there are valid objects entered.

The tab Services



Common Resources

Use this checkbox, to set if the station requests the same services for set-up and processing or if it requests separate services for set-up and processing. Depending on your choice, the object activates or deactivates corresponding dialog elements.

Setup, Process and Repair: Use these checkboxes to set if the station requests services for set-up, process or repair, respectively. Click on the buttons **Define** to define the services required for a specific type in a table. The service for set-up and process is defined dependent on the type of the entity, which is the value of a custom attribute *entityType* or the name of the movable element. The service for repair is defined dependent on the duration of the failure. The entered value is an upper bound for the failure duration. Therefore the table for defining the services for removing failures has a first column of type *time* and the table for defining the other services have a first column of type *string*.

	string 1	string 2	integer 3	string 4	integer 5	string 6	integer 7
string	entityType	Service	Amount	Tool	Number	ool	Number
1	all	Serv Setup	1				
2	xy	Setup xy	1	special Tool	1	pecial Tool	1
3							

Services and Tools for set-up and processing

- **entityType / Failure Duration:** Enter the types of parts or the duration of the failures into this column, respectively. When all types of parts require the same services, enter *all*.
- **Service / Amount:** Enter the names and the number of the requested services.

	time 1	string 2	integer 3	string 4	integer 5
string	Failure Duration	Service	Amount	Tool	Number
1	1:00:00.0000	Serv Repair	1		
2					

Services and Tools for repairing.

- **Tool / Number:** Enter the tools and the number to be requested

Release exporter by failure: Use this checkbox to set if the exporters return to the *Servicepool*, when the associated material flow object is failed. Otherwise the exporters remain at the associated material flow object during failures and continue processing after the failure ends.

Methods of the ServiceStation

setServices

Usage: `<path>.setServices(ServiceType, EntityType, Services);`

The method *setServices* set the services which are requested by the *ServiceStation*.

- **ServiceType:** The number of the types of services, 0 is for repair, 1 is for setup and 2 is for processing (type *integer*).
- **EntityType:** The name of the products for which the services are requested (type *string*).
- **Services:** A table containing the services together with the corresponding amount (type *table[string,integer]*).

setServiceType

Usage: `path.setServiceType(Fail, Together, Setup, Process);`

The method *setServiceType* set the kinds of services, which are requested by the *ServiceStation*.

- **Fail:** The parameter is True, if services for repairing must be requested (type *boolean*).
- **Together:** The parameter is True, if common services for setup and processing must be requested. The parameter is False, if different services are needed for setup and processing. (type *boolean*).
- **Setup:** The parameter is True, if services for setup must be requested. (type *boolean*).
- **Process:** The parameter is True, if services for processing must be requested. (type *boolean*).

setMachine

Usage: `<path>.setMachine(Machine);`

The method *setMachine* defines the station (machine) which is assigned to the *ServiceStation*.

- **Machine:** The station which is assigned to the *ServiceStation* (type *object*).

setServicePool

Usage: `<path>.setServicePool(Pool);`

The method *setServicePool* defines the *ServicePool* for the *ServiceStation*.

- **Pool:** The *ServicePool*, which is assigned to the *ServiceStation* (type *object*).

Positioning

Usage: `<path>.positioning;`

The method set the position of the *ServiceStation* nearly to the machine. This method is mainly needed, if there are extensive changes in the model and the *ServiceStations* should appear by its assigned machines.

in case a lot of objects are moved to new locations, you can attach the serviceStation to the corresponding machine by calling:

```
root.callevery( "positioning" );
```

This will attach the ServiceStations to the machines.

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