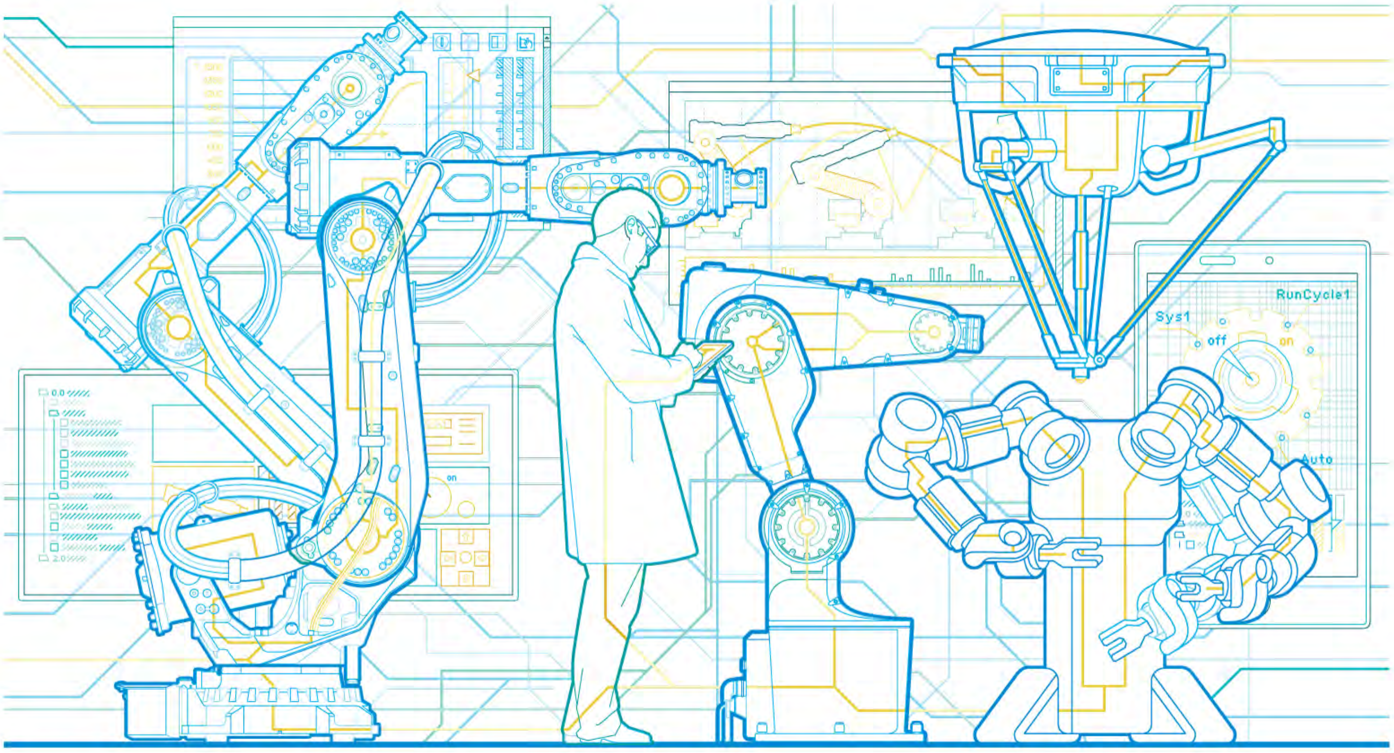


Operating manual
Machine Tending PowerPac

Trace back information:
Workspace Main version a205
Checked in 2017-02-27
Skribenta version 4.6.334

Operating manual

Machine Tending PowerPac

RobotStudio 6.04

Document ID: 3HAC044396-001

Revision: H

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damages to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Additional copies of this manual may be obtained from ABB.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

Original instructions.

ABB AB
Robotics Products
SE-721 68 Västerås
Sweden

Table of contents

Overview

About this manual

This manual contains information and instructions for installing, configuring, programming, and running Machine Tending PowerPac.

Usage

This manual should be used during installation and configuration of Machine Tending PowerPac. It describes Machine Tending PowerPac and includes step-by-step instructions to perform the tasks.

Who should read this manual?

This manual is intended for:

- System integrators
- End Customers
- Offline Programmers
- Service Technicians
- ABB engineers

Prerequisites

The reader should:

- Have experience with RobotStudio
- Have experience of installation and configuration work
- Good skills in the IRC5 robot controller and RAPID programming

References

References	Document ID
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - Trouble shooting IRC5	*3HAC020738-001*
Technical reference manual - System parameters	*3HAC050948-001*
Technical reference manual - RAPID kernel	*3HAC050946-001*
Technical reference manual - RAPID overview	*3HAC050947-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Operating manual - IRC5 with FlexPendant	*3HAC050941-001*
Application manual - RobotWare Machine Tending	*3HAC044398-001*
Application manual - RAPID development guidelines for handling applications	*3HAC046417-001*
Operating manual - RobotWare Machine Tending	*3HAC044397-001*
Application manual - RobotWare Machine Tending	*3HAC044398-001*

Continues on next page

Other references

References	Description
http://www.robotstudio.com/forum/	RobotStudio Support Forum

Revisions

Revision	Description
-	Released with RobotWare 5.15.01 First edition.
A	Released with RobotWare 5.15.02 Minor updates.
B	Released with RobotStudio 5.60 • Updated the section *MTPP gripper settings on page 38*. • Restructured the sections *Configuring a station on page 39* and *Program template on page 49*. • Updated the images in the section *Define Cycles on page 68*.
C	Released with RobotStudio 5.61 • Added the new section *Custom models on page 56*. • Added the new section *Excel approach for creating the smart component for stations on page 58*.
D	Released with RobotStudio 6.0 • Added the new section *Data editors on page 53*. • Added the new section *Synchronize to Station on page 109*.
E	Released with RobotStudio 6.01 Minor updates.
F	Released with RobotStudio 6.02 • Added information about auto create in the section *Define Cycles on page 68*. • Updated the section *Production view on page 112*.
G	Released with RobotStudio 6.03 Updated the following sections: • Updated the section *Define Cycles on page 68*. • Updated the section *HomeRun on page 90*.
H	Released with RobotStudio 6.04 Updated the following sections: • Added the new section *Test Move Editor on page 88*. • Updated the section *The Path View tool bar on page 101*.

Product documentation, IRC5

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

All documents listed can be ordered from ABB on a DVD. The documents listed are valid for IRC5 robot systems.

Product manuals

Manipulators, controllers, DressPack/SpotPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with exploded views (or references to separate spare parts lists).
- Circuit diagrams (or references to circuit diagrams).

Technical reference manuals

The technical reference manuals describe reference information for robotics products.

- *Technical reference manual - Lubrication in gearboxes*: Description of types and volumes of lubrication for the manipulator gearboxes.
- *Technical reference manual - RAPID overview*: An overview of the RAPID programming language.
- *Technical reference manual - RAPID Instructions, Functions and Data types*: Description and syntax for all RAPID instructions, functions, and data types.
- *Technical reference manual - RAPID kernel*: A formal description of the RAPID programming language.
- *Technical reference manual - System parameters*: Description of system parameters and configuration workflows.

Continues on next page

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, DVD with PC software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and trouble shooters.

The group of manuals includes (among others):

- *Operating manual - Emergency safety information*
- *Operating manual - General safety information*
- *Operating manual - Getting started, IRC5 and RobotStudio*
- *Operating manual - IRC5 Integrator's guide*
- *Operating manual - IRC5 with FlexPendant*
- *Operating manual - RobotStudio*
- *Operating manual - Trouble shooting IRC5*

Safety

Safety of personnel

A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.

Therefore, it is important that all safety regulations are followed when entering safeguarded space.

Safety regulations

Before beginning work with the robot, make sure you are familiar with the safety regulations described in the manual *Operating manual - General safety information*.

This page is intentionally left blank

1 Introduction

1.1 About Machine Tending PowerPac

Overview

RobotStudio Machine Tending PowerPac (MTPP) is an add-in for RobotStudio. MTPP provides a platform for quick, easy creation, and editing of machine tending robot cells in a 3D virtual environment. The cells are designed to work together with RobotWare Machine Tending.

Since MTPP has a library of common grippers and station types, and built-in support for most machines and peripheral equipments, getting a cell up and running in the virtual environment is easy . In addition, safety is simplified with the capability to define safe home position movements in a virtual environment.

Some of the features of Machine Tending PowerPac are:

- Define and configure parts and grippers.
- Define and configure stations and robot movements.
- Define HomeRun strategy.
- Define simulation, validate, and optimize cycles.

1.2 Terms and concepts

Machine Tending PowerPac concepts

The following table lists the terminologies and concepts used in Machine Tending PowerPac:

Concept	Description
Project	A project is the container of all information about a specific application. It is deployed on the Robot controller. There can be several projects in system, using RobotWare Machine Tending, a specific project can be selected for execution from the FlexPendant.
Station	A uniquely identifiable physical equipment or location in the cell where the robot moves to with the part to perform a specific action.
Programming concept	A guideline to programming MT applications which includes modularization of RAPID based on movement and administrative modules, naming conventions for variables and signals, special instructions, and so on.
Station Templates	A set of predefined RAPID templates installed along with the PowerPac for handling the common Machine Tending stations.
Station Load and Station Unload routines	Station Load routine is used when the robot loads the part in the machine. Station Unload routine is used when the robot removes the part from the machine. These routines are defined in the Station templates.
HomeRun strategy	A RAPID module which describes the path taken by the robot to move to the Home position from any position in the Machine Tending cell.
Movement Modules / Routines	All movements in a Machine Tending cell are defined as separate routines in Movement module. For example, `mv10_11( ), mv11_12( )..`
Cycle routines	Cycle routines define the sequence of stations that are tended by the Robot. There can be several cycle routines in a material handling application, such as, Production, Startup, Completion, and so on. These routines are defined for the part being produced.
RW Machine Tending data types	The following data types required by RobotWare Machine Tending are configured from MTPP: • `StationData` – Used to visualize the station on the FlexPendant • `GripData` – Control the gripper operations • `CycleData` – Used to visualize and select the corresponding cycle routine from the FlexPendant or remote interface • `PartData` – Defines the part being produced and the routines that are executed. • `Project.mtp` file – This file lists all the details about the the Project, such as, list of modules, system parameters, images, and so on.

Project concept

A project is the container of all information about a specific application. The Project builds the interface between the MTPP and RobotWare Machine Tending. A project created from MTPP contains the following information:

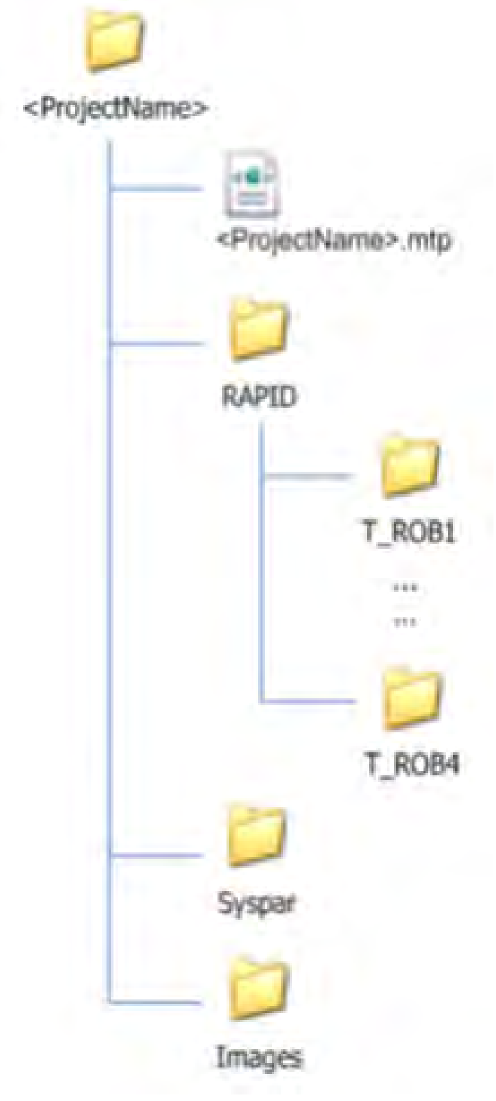
- Program and system modules
- System parameters
- Part and station related images

Continues on next page

- RobotStudio *Pack & Go* stations

The *Project.mtp* file is present in the Project folder and it lists all the above information. This information is used by the RobotWare Machine Tending user interface to load the corresponding modules into the controller. The Projects can be operated from the RobotWare Machine Tending user interface on the FlexPendant.

The project folder and file system are described in the following image:



xx1300000514

Folder / File	Description
`<ProjectName>`	The project folder contains all the required files and folders of a RobotWare Machine Tending project. The folder name is same as the project name.
`<ProjectName>.mtp`	XML-based project file which specifies all RAPID modules (`*.mod` and `*.sys`), system parameters, images and Pack&Go file, belonging to this specific project. The name of the project file and the project folder must be the same.

Continues on next page

Folder / File	Description
`RAPID`	Folder which holds the RAPID modules (`*.mod` and `*.sys`) of the project for up to 4 motion tasks.
`T_ROB1 – T_ROB4`	Contains the RAPID modules (`*.mod` and `*.sys`) for a specific motion task.
`Syspar`	Folder which holds all the system parameters of the project.
`Images`	All icons and pictures that are relevant for the project are located in this folder.

Structure of file `<ProjectName>.mtp`, filled with some example modules, parameters, images, and a pack and go file:

```
<?xml version="1.0" encoding="utf-16"?>
<!--RobotWare Machine Tending project file V1.0-->
  <Project>
    <Description Version="1.0" Date="2012-04-29">
      <Title>Bumper</Title>
      <Details>Producing bumpers</Details>
    </Description>
    <Rapid>
      <Task Name="T_ROB1" Program="Bumper">
      <Module>MT_MAIN.mod</Module>
      <Module>IMM.mod</Module>
      <Module>FLAMING.mod</Module>
      <Module>CNV.MOD</Module>
      <Module>Movement_T1.mod</Module>
      </Task>
    </Rapid>
    <Syspar>
      <Param>EIO.CFG</Param>
      <Param>PROC.CFG</Param>
    </Syspar>
    <Images>
      <Image>Picture1.png</Image>
      <Image>Picture2.png</Image>
    </Images>
    <PackandGo>Bumper station.rspag</PackandGo>
  </Project>
```

Station concept

A station in Machine Tending context is the uniquely identifiable physical equipment or a location in the cell where a robot moves with a part to perform a specific action.

Some examples of stations are:

- Processing stations
  - Lathe machine, IMM, Die-Casting Machine, Cooling station, Cutting station, Trimming Press, and so on.
- Sensor stations
  - Automatic or Manual inspection station.

Continues on next page

- Feeder stations
  - Palletize station, Infeeder or Outfeeder station.

From RAPID's perspective, the station operations are handled in a separate module for each station in the cell. For example, `IMM.mod, Conveyor.mod, Inspection.mod`, and so on. These modules contain the following:

- Initialization routine
- Load or unload routines
- Error handling within the station
- Signals mapping

Configured RobotWare Machine Tending data types needed to visualize the station details on the FlexPendant

Programming concept

MTPP creates programs based on the proposed guidelines for programming material handling applications based on the RobotWare Machine Tending. Following is an overview of the concepts relevant to MTPP:

- RAPID modularization
  - Demarcation between Administrative and Motion routines

    From MTPP, separate modules are created for Station, Part, Movement, Cycles, HomeRun, Main etc.
- Naming convention
  - Unique naming convention for RobTargets, Signals, Variables etc.
    - # All the targets positions in a Station are uniquely identified by a numbering scheme, pXX / pXXX, where X is the number. Example: IMM station positions are numbered between p10 to p19. Cutting station targets are numbered between p20 to p29.
    - # Additional positions are described as Intermediate positions Example: Intermediate positions between p10 and p11 are defined as p101101, p101102

      From MTPP, the template modules and Movement modules follow the prescribed convention.
- Motion
  - Movement instructions between two points are defined in separate routines
  - All the movement routines are stored in a separate module (Movement.mod)

```
!From: p11 : IMM loading Preposition
!To: p12 : IMM loading endposition;
PROC mv11_12()
MT_MoveJ
     11,p11,v50,z10,Tooldata1\WObj:=mtsProcessing1_Wobj\NoMove;
MT_MoveLDO
     12,p12,v1500,z10,Tooldata1\WObj:=mtsProcessing1_WObj,
     sdoGeneric1_Outside,1;
ENDPROC
```

Continues on next page

- The movement to a specific position is typically called from the station load or unload routines or cycle routines using a special instruction MT_MoveTo 12, where 12 is the position number.
- Depending on the current position, the corresponding movement routine is executed. Example: If the Robot is in position p11, then MT_MoveTo 12 would call the mv11_12() routine.

For more details, refer Application manual - RAPID development guidelines for handling applications.

Station templates concept

In Machine Tending PowerPac, pre-defined RAPID templates are provided to get started with the programming.

There are templates to handle machines such as Injection Moulding, Die Casting machines with pre-defined signal interfaces. There are generic templates for Loading, Unloading, Loading and Unloading, which is applicable for any type of station. For more information refer to *Configure stations on page 39*.

2 Installing Machine Tending PowerPac

2.1 System requirements

The following are the prerequisites for installing the Machine Tending PowerPac.

- A computer that meets or exceeds the system requirements as specified by RobotStudio.
- A log on account with administrator rights on the computer.
- RobotStudio 5.60

Hardware requirements	Software requirements
CPU: 2.0 GHz or faster processor, recommended is multicore processor	Windows 7
Memory: 1 GB RAM or more (more is recommended).	RobotStudio 5.60
2 GB RAM is running Windows Vista, Windows 7, Stations with several robot systems, or large CAD models	RobotWare 5.60
Available disk space: 5+ GB on the system disk, 250+ MB on the installation disk	.NET 4.0 Framework SP1
Graphics card: High performance OpenGL-compatible graphics card with the corresponding up-to-date drivers installed	
Screen resolution: 1280 x 1024 pixels (Recommended)	
Colors: 256 or higher	
DPI: Normal size (96 dpi)	
Mouse: Three-button mouse	

2.2 License key

Introduction

The license certificate document contains information about ABB contact information, an activation key, a list of licensed products, and expiry date.

For more information about licensing, see *Operating manual - RobotStudio*.

Prerequisites

You need a license key to use Machine Tending PowerPac. The license key is valid only on the computer on which the RobotStudio and the Machine Tending PowerPac are installed.

The license key is invalidated if you change your file system (for example, from FAT32 to NTFS).

Requesting a license key

To request your license key:

1. Log on to your computer.
2. Launch RobotStudio and then navigate to **Information: Manage Licences**
3. Click on **Activation Wizard** and select **I want to request a licence file**.
4. Enter the Activation key received along with RobotStudio.
5. Save the report and upload the file to the SOFA server.
6. On receiving the Licence key, install manually or activate over the Internet.

Trial and conversion

You can use Machine Tending PowerPac for a trial period of 30 days. The trial period license file `MachineTending_PowerPac.bin` is found in the installation folder of Machine Tending PowerPac folder, usually in `C:\Program Files\ABB Industrial IT\RoboticsIT\Machine Tending PowerPac 5.60\trial_license\` (for 32 bit) or `C:\Program Files (x86)\ABB Industrial IT\RoboticsIT\Machine Tending PowerPac 5.60\trial_license\` (for 64 bit). Activate this license file to use Machine Tending PowerPac for a trial period.

Once the trial period is over, procure and activate a licence to continue using Machine Tending PowerPac.

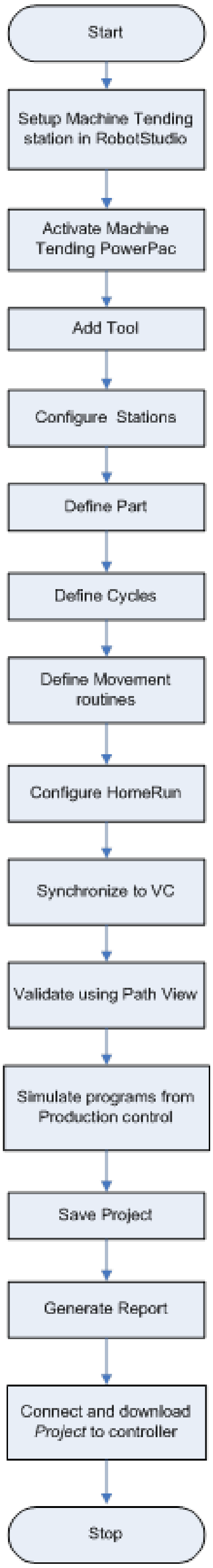
3 Workflow

3.1 Machine Tending PowerPac

3.1.1 Operational flow

The following flowchart shows the recommended flow for working with the Machine Tending PowerPac:

Continues on next page



en1300000512

Continues on next page

Task	Description
Setup Machine Tending station in RobotStudio	Create system with additional option RobotWare Machine Tending and include them to a RobotStudio station.
Activate Machine Tending PowerPac	Click **Add-Ins** Tab and click **Machine Tending**. The Machine Tending tab opens with a default project listed in the browser tree. The positions and movement routines between home and service positions (`p999, p991-993`) are created automatically in the movement module upon activation.
Add Tool	Attaches a tool to the robot. `ToolData` and `GripData` can be defined in the Tool.
Configure Stations	Add Machine Tending stations to the cell. The *StationData* data type is defined for each station. The stations are created based on the selected models and RAPID templates. The station is added to RobotStudio together with the corresponding movement routines.
Define Part	In RobotWare Machine Tending, selecting a `Part` is the first step to execute the cycles. The Part is identified by the data type `Partdata` in this step.
Define Cycles	These define the sequence of stations that are tended by the robot. There can be several cycle routines in a material handling application such as production, startup, completion, and so on. These routines are defined for the part being produced (updates the `PartData` with the cycle routine name). All the cycle routines are saved in the `Cycle.mod` in the project folder. The data type `CycleData` is created in this step, this enables selecting the cycle for execution from the RobotWare Machine Tending user interface and from Production simulation in MTPP.
Define Movement routines	All the station related movement routines are highlighted in the grid. Iteratively teach the different movement routines between the stations. Define the missing movement routines: • between stations based on the cycles. • From or To the Home position to stations.
Define HomeRun	Define the strategy to move the robot to the home position from any position in the cell from the grid. This is based on the various possible movements defined from the previous steps. The `MT_Home_User` RAPID module is created in the Project folder in this step.
Validate using Path view	In this step, a test movement routine with different movement routines from the selected Movement module is created. Arrange the order of the routines similar to the cycle sequence and verify the reachability, configuration, and simulation of the movements.
Synchronize to RAPID	In this step, synchronize all the content of the Project from MTPP to the Virtual controller.
Simulation cycles Production control	In this step, the cycles for the selected part can be executed. The configured HomeRun sequence for each position can also be verified.
Save Project	In this step, the project modules on the Virtual controller are saved into the Project folder in the file system.

Continues on next page

Task	Description
Generate Report	Creates a `*.doc` report for the project. It contains all the information configured from MTPP.
Connect and Download to controller	Connect to a real controller and download the selected project.

Following are the few data types required by RobotWare Machine Tending that are configured by Machine Tending PowerPac:

- `StationData` – To visualize the station on the FlexPendant.
- `GripData` – To control the gripper operations.
- `CycleData` – To visualize and select the corresponding cycle routine from the FlexPendant or remote interface.
- `PartData` – Defines the part being produced and the routines that are executed.
- `Project.mtp` file – Lists all the details about the project such as list of modules, system parameters, images, and so on.

3.2 Project and data type overview

When the Machine Tending add-in is activated, a project is created and listed in the browser tree of Machine Tending. The corresponding project folder and `Project.mtp` file are created in the file system.

The workflow for MTPP defines the following RobotWare Machine Tending data types during the different steps.

Step	Data type	Description
Activate MTPP – Creates Project	ProjectInfo	Defines a RobotWare Machine Tending project
Add Tool	GripData	Controls the gripper operations
Define Part	PartData	Defines the data for the part being produced including the corresponding cycle routines.
Configure Stations	StationData	Displays the station on the RobotWare Machine Tending user interface with all the status information.
Define Cycles	CycleData	Enables the selection of cycle routines from the RobotWare Machine Tending user interface on the FlexPendant to support production of part.

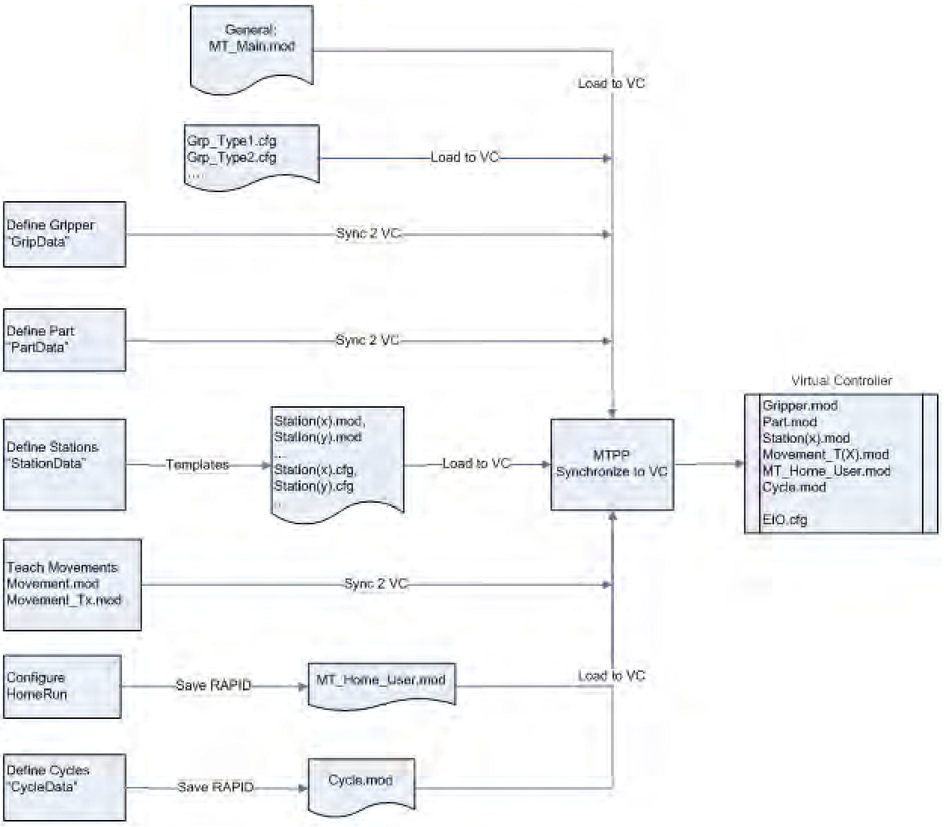
A Project contains different modules for Station(s), Movement, Parts, Gripper, Cycles, HomeRun, and so on. From MTPP, these modules are added or created from various sources during **Synchronize to VC** as described in the following table:

Module	Description	Source	RobotWare Machine Tending Data type
`MT_Main.mod`	Contains the main( ) routine which instantiates the RobotWare Machine Tending execution.	Copied into system folder by default by MTPP. Load module to include into the VC.	`ProjInfo` `CycleData`
`Station(s).mod`	Contains the logic for handling the Station. There is a module for each station in the cell.	Created from MTPP using templates. Automatically saved into the system folder. Load module to include into the VC.	`StationData`
`Parts.mod`	This includes the declaration of the data type `PartData`.	Created from MTPP when synchronize to the VC.	`PartData`
`Gripper.mod`	This includes the declaration of the data types `GrpData` and `ToolData`.	Created from MTPP when synchronize to the VC.	`GripData`
`Cycles.mod`	Contains the cycle logic for handling the part.	Created from MTPP To be saved manually into the system folder. Load module to include into the VC.	

Continues on next page

Module	Description	Source	RobotWare Machine Tending Data type
`MT_Home_User.mod`	Contains the strategy to move the robot to the home position.	Created from MTPP. To be saved manually into the system folder. Load module to include into the VC.	
`Movement_T(X).mod`	Contains movement routines for the station and cell. There can be several movement modules for each part type.	Created from MTPP when synchronize to the VC.	

The process of project modules being created on the VC is shown in the following image:



en1300000513

3.3 Setting up the Machine Tending system

Installation and system generation

Use the following procedure to install and create a RWMT system:

1. Install RobotStudio 6.0 or higher on you PC.
2. Open RobotStudio and click the **Controller** tab.
3. Click **Installation Manager**.

   The **Installation Manager** window is displayed.
4. Click **Controllers**, and then click the **Virtual** tab.
5. Click **New**.

   The **Create New** window opens.
6. In the **Name** field, type a name for the new system.
7. Click **Next.**

   The **Products** tab is displayed.
8. Click **Add.**

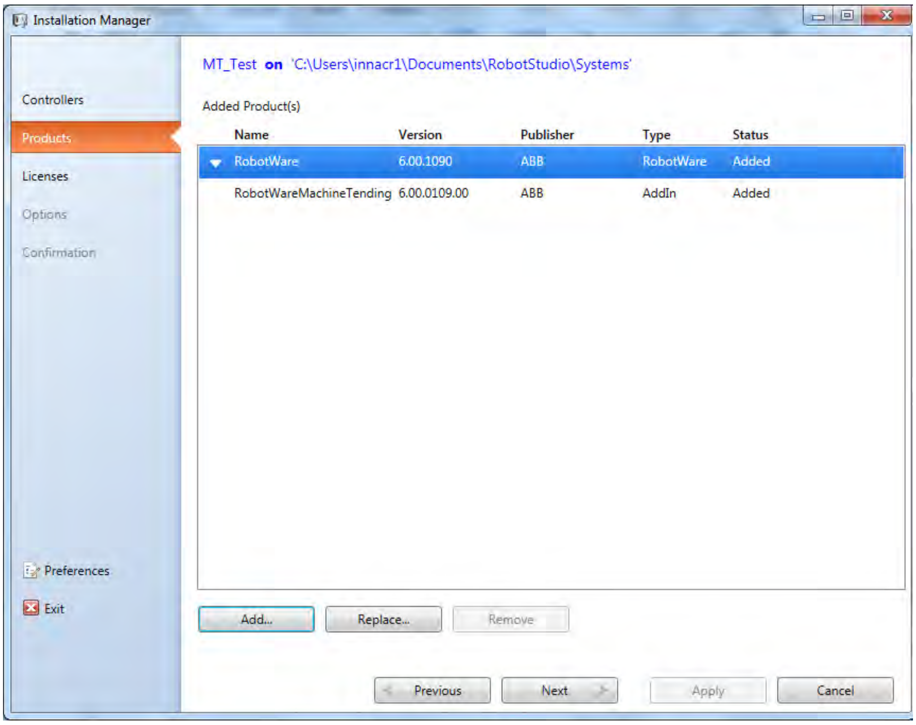
   The **Select Product** window opens.
9. Click **Browse** and select the product manifest file of RobotWare, and click **Open**.

   The selected product is added to the **Added Product(s)** window.
10. Click on **Add** again.

    The **Select Product** window opens.
11. Click **Browse** and select the product manifest of Robotware Machine Tending and click **Open**.

Continues on next page

The selected Robotware Machine Tending product is added to the **Added Product(s)** window.



MT_Test **on** 'C:\Users\innacr1\Documents\RobotStudio\Systems'

Added Product(s)

Name	Version	Publisher	Type	Status
RobotWare	6.00.1090	ABB	RobotWare	Added
RobotWareMachineTending	6.00.0109.00	ABB	AddIn	Added

xx1400002481

12 Click **Next**.

The **Licenses** tab is displayed.

13 Click **Add**.

The **Select License** window is displayed.

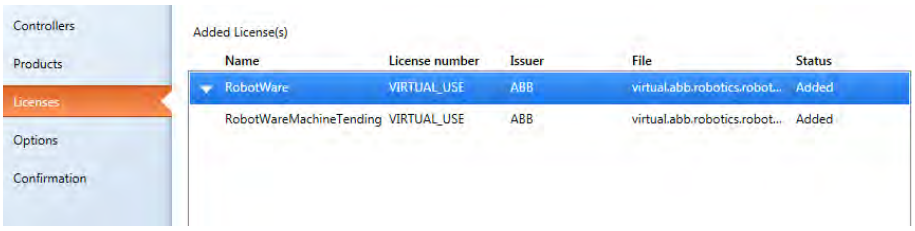
14 Click **Browse**, select the Robotware license file, and click **Open**.

The Robotware license file is added to the **Added License(s)** window.

15 Repeat the same step to add RobotWare Machine Tending License files to your system.

Note

For working with Virtual Controller add the RWMT virtual license. To work with real controller two RWMT licenses should be added.



Added License(s)

Name	License number	Issuer	File	Status
RobotWare	VIRTUAL_USE	ABB	virtual.abb.robotics.robot...	Added
RobotWareMachineTending	VIRTUAL_USE	ABB	virtual.abb.robotics.robot...	Added

xx1400002482

16 Click **Next**.

Continues on next page

The **Options** tab is displayed. This tab displays the **System Options**, **Drive Modules**, and **Applications**. You can customize your options here.

17 Click the **Applications** tab and select the **1167-1 RW Machine Tending** option from the **Application Machine Tending** group.

18 Click **Next.**

The **Confirmation** tab is displayed and shows an overview of the selected options.

19 Click **Apply**.

The **Apply Changes** confirmation window is displayed.

20 Click **Yes.**

The added products (RobotWare and RWMT) are installed and the system is created.

This page is intentionally left blank

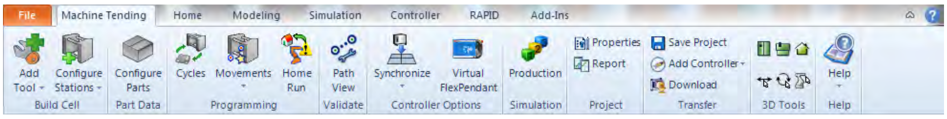
4 User interface

4.1 Introduction

4.1.1 Graphical user interface

Machine Tending ribbon

Click the Machine Tending tab, the Machine Tending page is displayed. The following image and table provides details about the elements in the Machine Tending ribbon.



xx1300002615

Group	Button	Description
Build Cell	**Add Tool**	Allows you to import a tool library into RobotStudio and attaches the selected tool to the robot.
	Configure Stations	Allows you to add a station to the cell.
Part Data	**Configure Parts**	Allows you to configure part data.
Programming	**Cycles**	Allows you to build the cycle data and define the cycle logic.
	Movements	Allows you to define movements between stations or within stations.
	Home Run	Allows you to ensure that the robot reaches the Home position from any position in the cell.
Validate	**Path View**	Displays the targets in a path by representing them with instruction icons.
Controller Options	**Synchronize to RAPID**	Allows you to synchronize the selected program on the browser to virtual controller
	Virtual FlexPendant	Allows you to open the virtual FlexPendant.
Simulation	**Production**	Allows you to enable simulation of cycles in RobotStudio.
Project	**Properties**	Allows you to manage a selected project.
	Report	Allows you to generate project reports in .doc format for reference purpose.
Transfer	**Save Project**	Allows you to save a selected project.
	Add Controller	Allows you to manage controller.
	Download	Allows you to download a project to the selected controller.

Continues on next page

Group	Button	Description
3D Tools	View and Move tools	Tools in the **3D Tools** group allows you to manage the movement of the robot, and manage the view of the robot system. • - The **Top** tool displays the top view of the station with reference to the coordinate system. • - The **Front** tool displays the front view of the station with reference to the coordinate system. • - The **Right** tool displays the right view of the station with reference to the coordinate system. • - The **Move** tool allows you to drag an item relative to the reference coordinate system. • - The **Rotate** tool allows you to enable rotation around the various axes of an object determined by the reference coordinate system. • - The **Jog Joint** tool allows you to jog the different axes of a robot.
Help	Manual	Displays the Online Help.
	About	Displays the product version and support contact information.

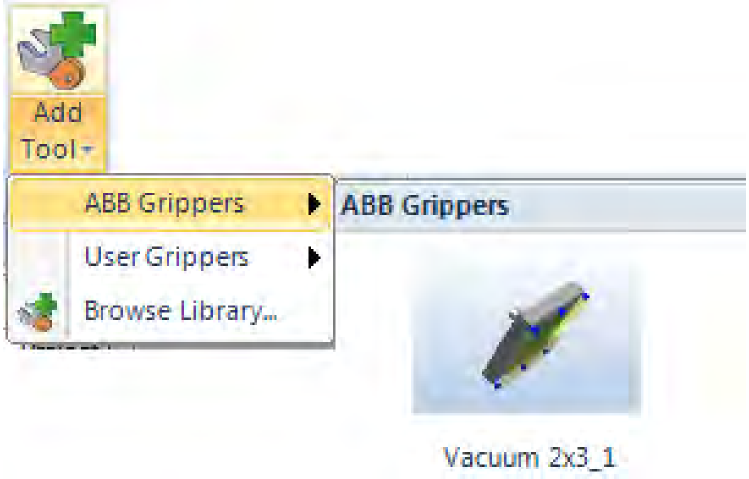
4.2 Build cell

4.2.1 Add tool

4.2.1.1 Adding tool

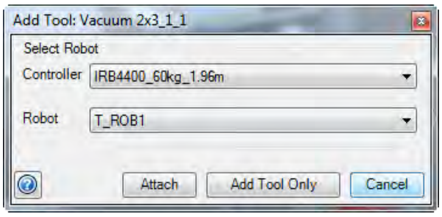
Adding tool

The **Add Tool** option imports a tool library into RobotStudio and attaches the selected tool to the robot. Adding a tool is the first step to build a cell with different stations.

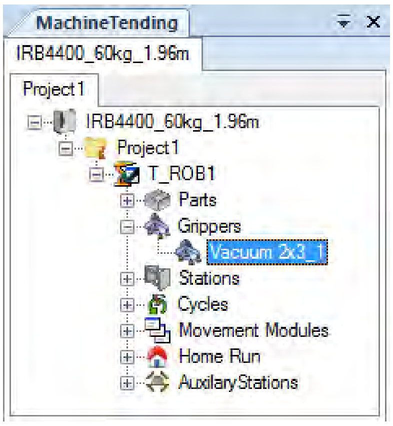


xx1300000358

To add a tool:

	Action	Description
1	Click **Add Tool**. The **Add Tool** menu is displayed.	
2	Select **ABB Grippers**. The **Add Tool** window is displayed.	 Add Tool: Vacuum 2x3_1_1 Select Robot Controller: IRB4400_60kg_1.96m Robot: T_ROB1 xx1300000359
3	Select a controller from the **Controller** list and a robot from the **Robot** list.	

Continues on next page

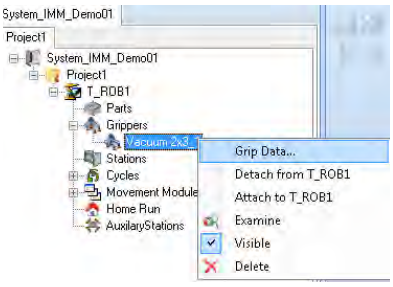
	Action	Description
4	Click **Attach**. The gripper is attached to the robot and is listed under the browser tree.	 MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Grippers Vacuum 2x3_1 Stations Cycles Movement Modules Home Run AuxilaryStations xx1300000495 The **Add Tool Only** option attaches the gripper to the station, but not attaches the same to the robot.

The tool added from PowerPac includes the `ToolData` and `GripData` declarations which are used in the stations.

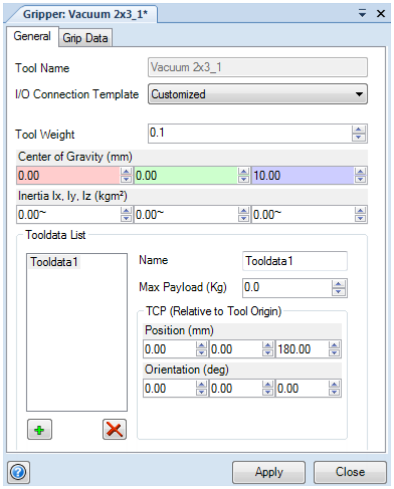
Editing GripData

The data type `GripData` is used to control the gripper operations in RobotWare Machine Tending. The Grippers attached to the Robot are listed under the browser tree.

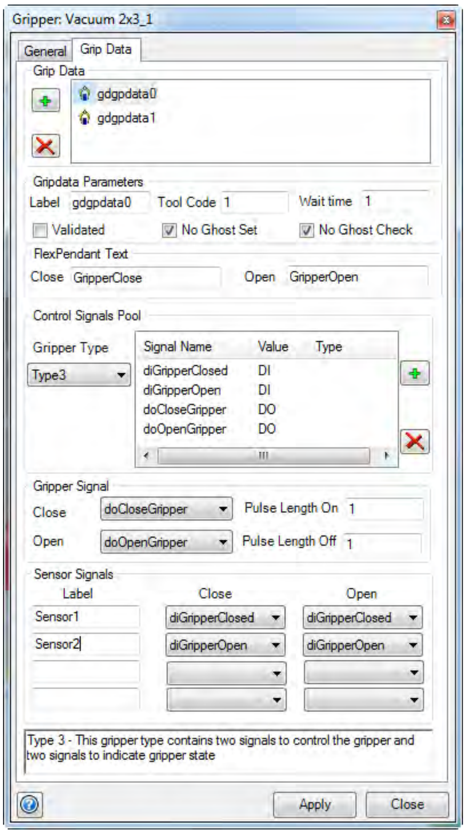
To edit `GripData`:

	Action	Description
1	Right-click on the gripper tool in the Browser tree and select **Grip Data...** from the browser tree. The editor for `ToolData` and `GripData` is displayed. By default, the tool includes definitions of `ToolData` and two `GripData`.	 System_IMM_Demo01 Project1 System_IMM_Demo01 Project1 T_ROB1 Parts Grippers Vacuum 2x3 Stations Cycles Movement Module Home Run AuxilaryStations Grip Data... Detach from T_ROB1 Attach to T_ROB1 Examine Visible Delete xx1300000546

Continues on next page

	Action	Description
2	In the **General** tab, you can edit `ToolData`.	 Gripper: Vacuum 2x3_1* General | Grip Data Tool Name: Vacuum 2x3_1 I/O Connection Template: Customized Tool Weight: 0.1 Center of Gravity (mm): 0.00 0.00 10.00 Inertia Ix, Iy, Iz (kgm²): 0.00~ 0.00~ 0.00~ Tooldata List: Tooldata1 Name: Tooldata1 Max Payload (Kg): 0.0 TCP (Relative to Tool Origin) Position (mm): 0.00 0.00 180.00 Orientation (deg): 0.00 0.00 0.00 Apply Close xx1300000547 Click + to create ToolData or click ✕ to delete a selected Tooldata. **Note** If a ToolData is used in a station then it is not possible to delete the ToolData.

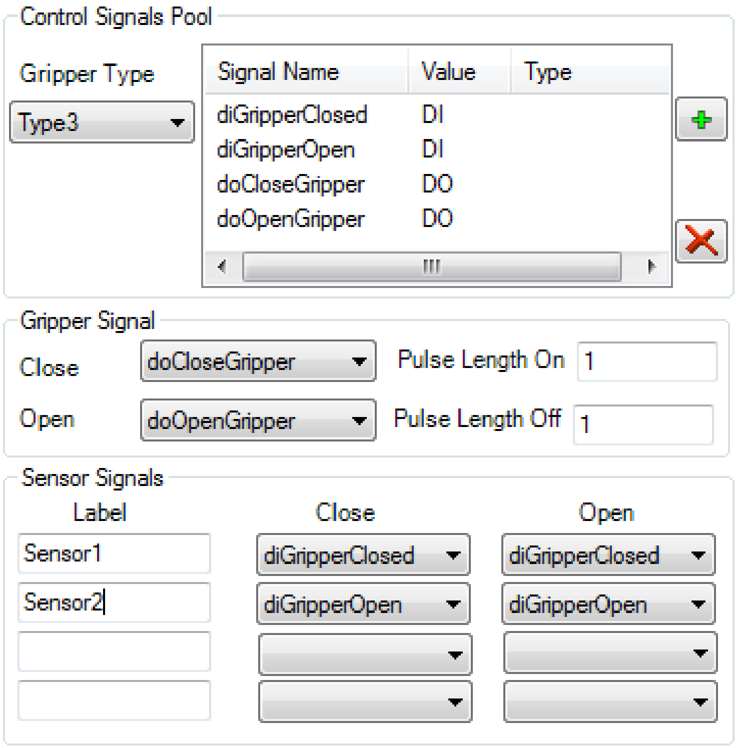
Continues on next page

	Action	Description
3	In the **Grip Data** tab, you can edit `GripData`. Select a `GripData` to display and edit the parameters. The complete list of properties can be accessed from the *Application manual - RobotWare Machine Tending*.	 Gripper: Vacuum 2x3_1 Grip Data gdgpdata0 gdgpdata1 Gripdata Parameters Label gdgpdata0 Tool Code 1 Wait time 1 □ Validated ☑ No Ghost Set ☑ No Ghost Check FlexPendant Text Close GripperClose Open GripperOpen Control Signals Pool Gripper Type Type3 Signal Name / Value / Type diGripperClosed DI diGripperOpen DI doCloseGripper DO doOpenGripper DO Gripper Signal Close doCloseGripper Pulse Length On 1 Open doOpenGripper Pulse Length Off 1 Sensor Signals Label / Close / Open Sensor1 diGripperClosed diGripperClosed Sensor2 diGripperOpen diGripperOpen Type 3 - This gripper type contains two signals to control the gripper and two signals to indicate gripper state xx1300000360 You can add, delete, or edit the RobotWare Machine Tending data type `GripData` from this tab. This is used to control the gripper actions and is used in RAPID together the instruction `MT_GripSet`. There can be several `GripData` used in an application. Click + to add a new `GripData` or click ✕ to delete a selected `GripData`. The signals for closing and opening the gripper (DO) and sensor signals (DI) can be selected. This process is made simple by providing a signal pool from where you can choose the signals. The signals can be easily added to this pool by choosing a gripper type which represents the tool being used.
4	Click **Apply**. The changes in the `ToolData` and `GripData` are saved in the gripper.	After synchronizing to the VC, the corresponding data types are declared under the *Gripper.mod* module in the controller.

Continues on next page

Controls signals pool

In the `GripData` declaration, you must select the signals for closing and opening the gripper (DI) and sensor signals to indicate whether the gripper is closed or opened (DO). This process is made simple by providing a signal pool from where you can choose the signals.



Control Signals Pool

Gripper Type: Type3

Signal Name	Value	Type
diGripperClosed	DI	
diGripperOpen	DI	
doCloseGripper	DO	
doOpenGripper	DO	

Gripper Signal

Close: doCloseGripper — Pulse Length On: 1

Open: doOpenGripper — Pulse Length Off: 1

Sensor Signals

Label	Close	Open
Sensor1	diGripperClosed	diGripperClosed
Sensor2	diGripperOpen	diGripperOpen

xx1300000362

The signals can be easily added to this pool by choosing a gripper type which represents the tool being used. The different signal types are listed in the following table.

Type	Description
Type1	Contains one signal to control the gripper operation.
Type2	Contains two signals to control the gripper operation.
Type3	Contains two signals to control the gripper and two signals to indicate gripper state.
Type4	Contains two signals to control the gripper. One signal to indicate gripper is open and another to indicate whether the gripper is closed completely, this indicates whether part is gripped or not.
Type5	Contains two signals to control the gripper. One signal to indicate whether the gripper is closed completely and another signal to indicate whether the part is in the gripper.
Type6	Contains two signals to control the vacuum gripper. One signal is to indicate whether the vacuum is OK.

You can add signals from the controller to the **Control Signals Pool** using the ✚ button.

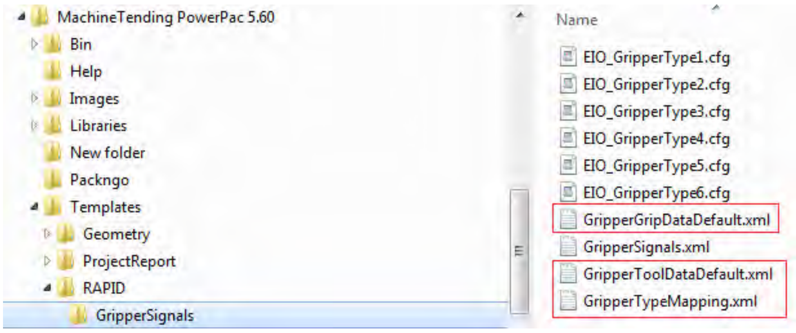
Continues on next page

MTPP gripper settings

For default grippers installed with Machine Tending PowerPac, each gripper is associated with

- default tooldata
- default gripper type mapping
- default gripdata

This association is defined in an *.xml file in the installation folder as shown.



MachineTending PowerPac 5.60
Bin
Help
Images
Libraries
New folder
Packngo
Templates
Geometry
ProjectReport
RAPID
GripperSignals

Name
EIO_GripperType1.cfg
EIO_GripperType2.cfg
EIO_GripperType3.cfg
EIO_GripperType4.cfg
EIO_GripperType5.cfg
EIO_GripperType6.cfg
GripperGripDataDefault.xml
GripperSignals.xml
GripperToolDataDefault.xml
GripperTypeMapping.xml

xx1300002633

The gripdata parameters are populated based on these files. You can modify the “GripperGripDataDefault.xml"” file and “GripperTypeMapping.xml” file for editing these default associations or adding associations for a new gripper.

4.2.2 Configure stations

4.2.2.1 Configuring a station

Introduction

In MTPP, a station represents the equipment which is used to process the part. The Machine Tending PowerPac categorizes the stations into the following three types.

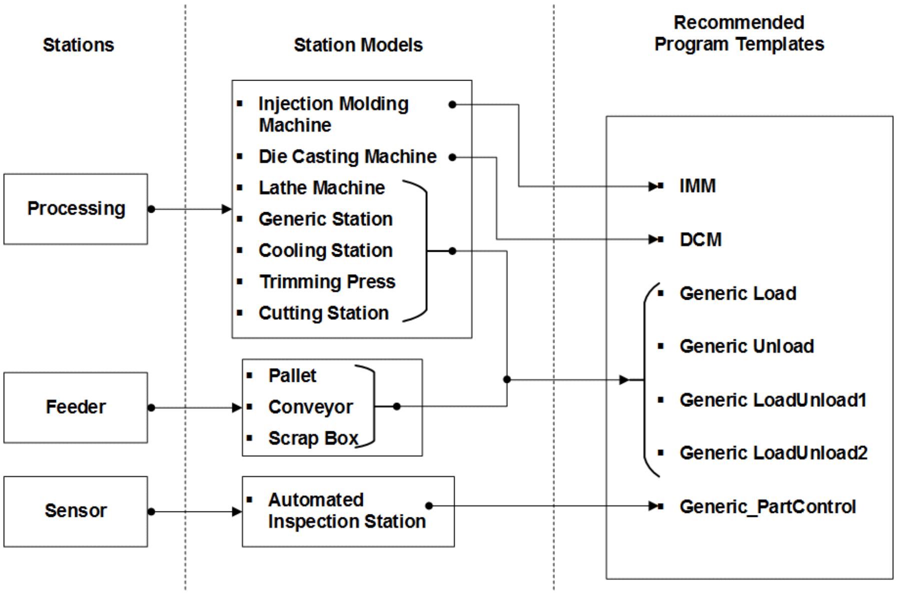
- **Processing** - The Processing station is any generic station which operates on a part and works on it, giving it a different physical form and contributes to the change of its appearance and properties. For example, Injection Molding Machine (IMM), Die Casting Machine (DCM), Lathe Machine, Cutting Station, Cooling Station, Trimming Press, and Generic station.
- **Sensor** - The Sensor station is a machinery which has sensors and inspects a part depending on the inspection criteria as defined, to consider the part or reject it. For example, Automated Inspection station.
- **Feeder** - The Feeder station is a machinery which is used to move or transport a part from one location in the cell to other. For example, Conveyor, Pallet, and Scrap Box.

The MTPP station component has the following associated files:

- Station.xml
- Station.lib
- Station.cfg

The details about these files are explained in the Templates section.

The process for adding stations is illustrated in the following figure.

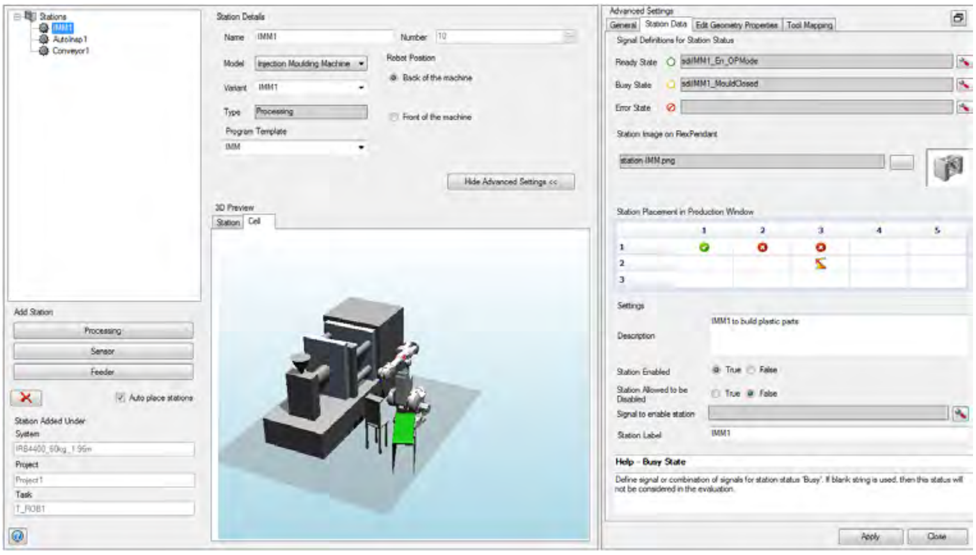


xx1300002629

Continues on next page

Adding a station

You can add a station to the cell using the **Configure Stations** window.



Stations
- IMM1
- AutoInsp1
- Conveyor1

Station Details

Name: IMM1
Number: 10
Model: Injection Moulding Machine
Variant: IMM1
Type: Processing
Program Template: IMM

Robot Position
● Back of the machine
○ Front of the machine

Add Station

☑ Auto place stations

Station Added Under
System: IRB4400_60kg_1.96m
Project: Project1
Task: T_ROB1

Advanced Settings

Signal Definitions for Station Status
Ready State: sdiIMM1_En_OPMode
Busy State: sdiIMM1_MouldClosed
Error State:

Station Image on FlexPendant: station IMM.png

Station Placement in Production Window

	1	2	3	4	5
1	✓	☒	☒		
2			⚠		
3					

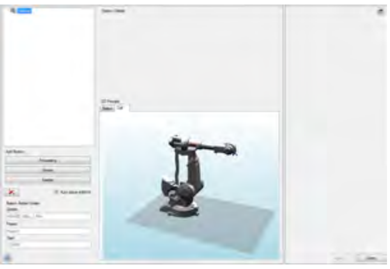
Settings
Description: IMM1 to build plastic parts
Station Enabled: ● True ○ False
Station Allowed to be Disabled: ○ True ● False
Signal to enable station:
Station Label: IMM1

Help - Busy State

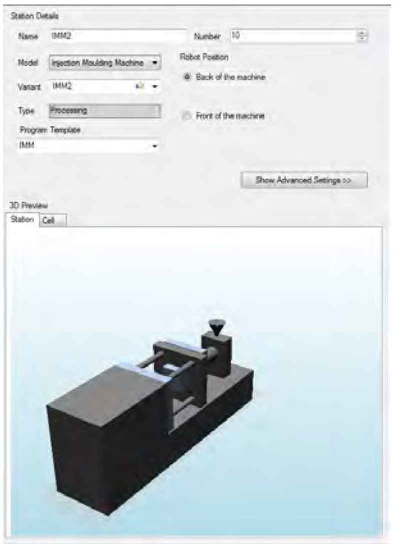
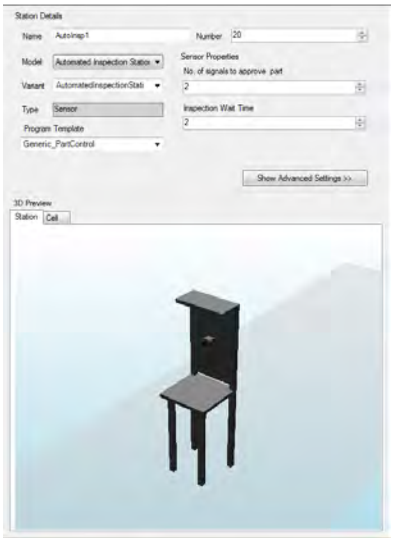
Define signal or combination of signals for station status 'Busy'. If blank string is used, then this status will not be considered in the evaluation.

xx1300000378

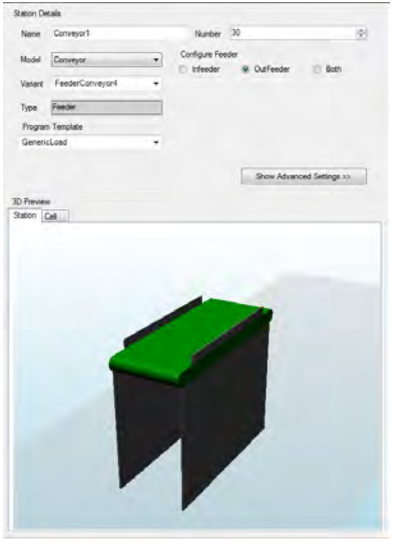
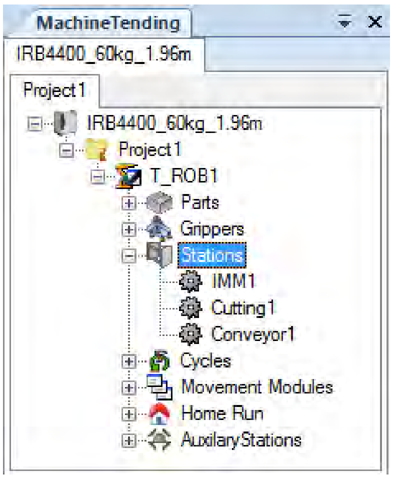
To add a station:

	Action	Description
1	Click **Configure Stations** > **Configure Stations** in the **Build Cell** group. The **Configure Stations** window is displayed.	 xx1300002617

Continues on next page

	Action	Description
2	Click **Processing**. The **Station Details** window is displayed.	By default, for a processing station type, a Generic Station model is added. For Processing stations IMM/DCM model, there are options to place the station either front or back of the machine.  Station Details Name: IMM2 Number: 10 Model: Injection Moulding Machine Variant: IMM2 Type: Processing Program Template: IMM Robot Position: ● Back of the machine ○ Front of the machine xx1300002618
3	Click **Sensor**. The **Station Details** window is displayed.	For Inspection stations, you can define the number of signals in the station and the time the robot waits at the station.  Station Details Name: AutoInsp1 Number: 20 Model: Automated Inspection Statio Variant: AutomatedInspectionStati Type: Sensor Program Template: Generic_PartControl Sensor Properties No. of signals to approve part: 2 Inspection Wait Time: 2 xx1300002619

Continues on next page

	Action	Description
4	Click **Feeder**. The **Station Details** window is displayed.	By default, for a feeder station type, a pallet station model is added. For Feeder stations, you can choose the type of feeder, that is, whether infeeder or outfeeder or both, based on which the corresponding templates are selected. Station Details Name: Conveyor1 Number: 30 Model: Conveyor Configure Feeder: ○ Infeeder ● OutFeeder ○ Both Variant: FeederConveyor4 Type: Feeder Program Template: GenericLoad xx1300002620
5	Click **Apply**. The station is included in the cell and listed under the browser tree.	MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Grippers Stations IMM1 Cutting1 Conveyor1 Cycles Movement Modules Home Run AuxilaryStations xx1300000496 To customize a station, select the station and edit the station settings in the **Station Details** section. More more details see, *Editing a station on page 43*.

Continues on next page

Editing a station

Editing a station

You can edit a station from the **Configure Stations** window.

To edit a station:

	Action	Description
1	Click **Configure Stations** in the **Build Cell** group. The **Configure Stations** window is displayed.	
2	Select a station to edit from the **Stations** browser. The details of the selected station is displayed in the **Station Details** section.	
3	Type a name for the station in the **Name** text box.	This name is used in the workobject, signals, and program module associated with the station. **Note** ABB recommends to use less than 10 characters for the station name, which is a valid RAPID name, as the template prefixes and suffixes this name in station signals.
4	Select a number for the station in the **Number** list.	Station numbers are displayed as multiples of 10. It is a unique number associated with a station, where the rob targets follow the number system pxx or pxxx. For example, the station number 10 has the targets p10, p11, p12 and so on.
5	Select a model for the station from the **Model** list.	The selected model is displayed in the 3D preview.
6	Select a variant for the station from the **Variant** list.	Every station model has some variants, which are in ascending order of the geometrical size. The most suitable variant for the chosen robot is represented with a 👍 thumps up symbol in the **Variant** list.
7	Select a template from the **Program Template** list.	For the selected model a template is pre selected. It is possible to change the template. For more information on templates refer to the *Program template on page 49* chapter.
8	Click **Show Advanced Settings** to edit more settings.	The following sections provide more details about editing the tabs in **Advanced Settings** section. • *Advanced settings - General on page 44* • *Advanced settings - Station data on page 45* • *Advanced settings - Edit geometry properties on page 47* • *Advanced settings - Tool mapping on page 48*

Continues on next page

	Action	Description
9	Click **Apply**. The changes are saved.	For some stations, additional options can be selected which simplify the station creation process.

Deleting a station

Click ✕ to delete a selected station from the **Stations** section.

Auto placement of stations

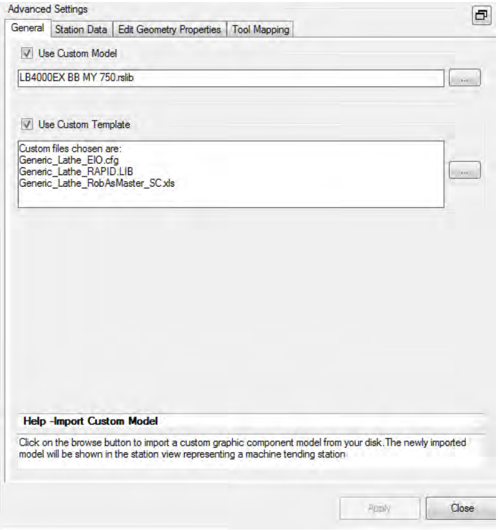
If you select the **Auto place stations** check box ☑ Auto place stations, all the stations placement are updated whenever a station is added, deleted, or edited. If the **Auto place stations** check box is not selected then a new station is placed in the base position.

> **Tip**
>
> While adding or deleting a station, it is suggested to clear the **Auto place stations** check box to avoid accidentally updating the placement of all the other stations in the cell.

Advanced settings - General

Use a customized model

The **General** tab provides settings for importing custom models for stations.



☑ Use Custom Model

LB4000EX BB MY 750.rslib

☑ Use Custom Template

Custom files chosen are:
Generic_Lathe_EIO.cfg
Generic_Lathe_RAPID.LIB
Generic_Lathe_RobAsMaster_SC.xls

Help -Import Custom Model

Click on the browse button to import a custom graphic component model from your disk.The newly imported model will be shown in the station view representing a machine tending station

xx1300000373

To import a custom model:

Action	Description
Select the **Use Customized model** check box.	The browse button is enabled.
Browse and navigate to the folder location and select and import the required file.	The file is imported into RobotStudio.

Continues on next page

To import custom templates:

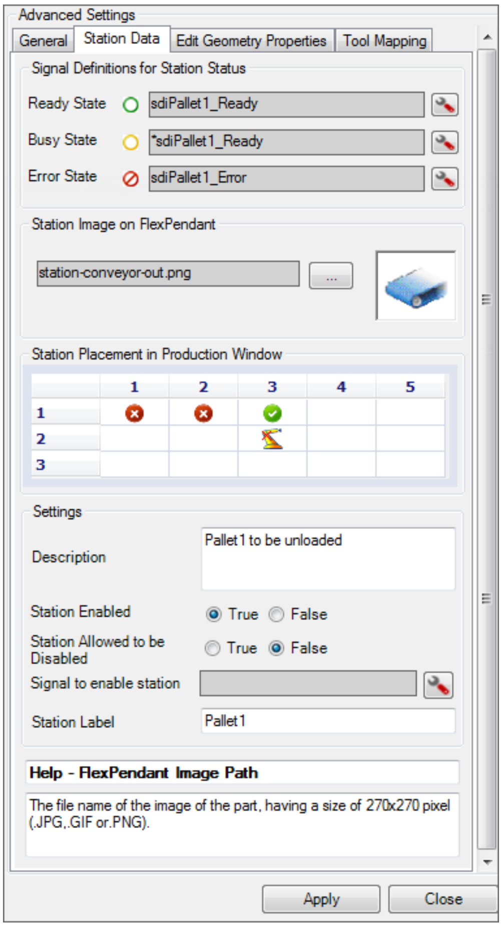
Action	Description
Select the **Use Custom Template** check box.	The browse button is enabled.
Browse and navigate to the folder location and select and import the required files.	The templates are imported into PowerPac.

For more information about creating custom stations, see *Custom models on page 56*.

Advanced settings - Station data

Station data

StationData is a data type associated with a station. The **StationData** tab allows you to edit the station data parameters. For more details about `stationdata`, see *Application manual - RobotWare Machine Tending*.



Advanced Settings

General | Station Data | Edit Geometry Properties | Tool Mapping

Signal Definitions for Station Status

Ready State	○	sdiPallet1_Ready
Busy State	○	*sdiPallet1_Ready
Error State	⊘	sdiPallet1_Error

Station Image on FlexPendant

station-conveyor-out.png

Station Placement in Production Window

	1	2	3	4	5
1	✗	✗	✓		
2					
3					

Settings

Description: Pallet1 to be unloaded

Station Enabled: ● True ○ False

Station Allowed to be Disabled: ○ True ● False

Signal to enable station:

Station Label: Pallet1

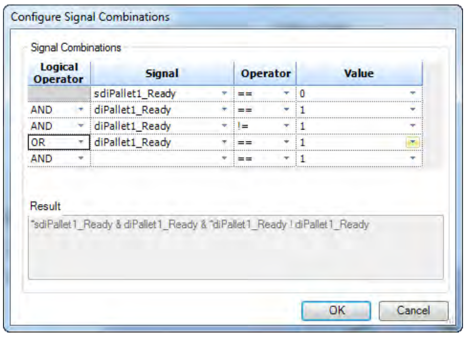
Help - FlexPendant Image Path

The file name of the image of the part, having a size of 270x270 pixel (.JPG,.GIF or.PNG).

xx1300000374

Continues on next page

To edit station data:

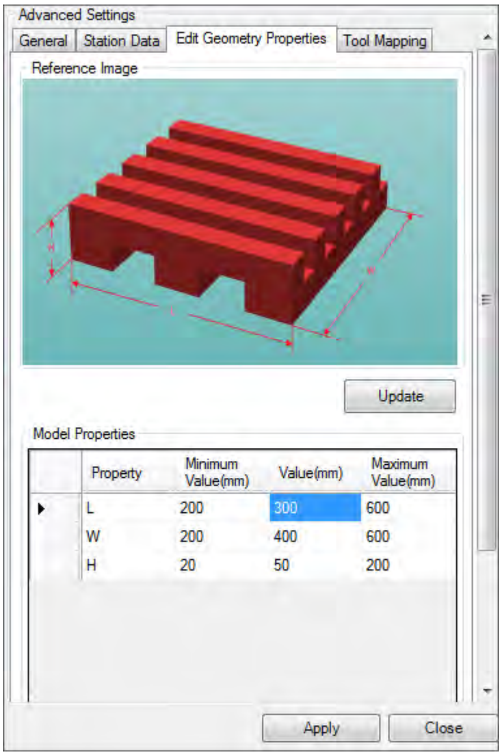
Action	Description
Click the configure icon of a state in the **Signal Definitions for Station Status** section. The **Configure Signal Combinations** window is displayed.	 Configure Signal Combinations Signal Combinations OK Cancel Logical Operator | Signal | Operator | Value sdiPallet1_Ready | == | 0 AND | diPallet1_Ready | == | 1 AND | diPallet1_Ready | != | 1 OR | diPallet1_Ready | == | 1 AND | | == | 1 Result *sdiPallet1_Ready & diPallet1_Ready & *diPallet1_Ready ! diPallet1_Ready xx1300000377 The **Configure Signal Combinations** window displays the stations signals that can be used to configure the ready, busy, or error state for a station.
Edit the information in the **Signal Combinations** section.	While defining a combination of signals, RobotWare Machine Tending uses the following notations in the **Result** section: • * - Invert the signal. • & - AND connection. • ! - OR connection. For more information, see *Application manual - RobotWare Machine Tending*.
Click **OK**. The editor closes and the new combination is displayed in the stationdata user interface.	The configure icon in the **Signal to enable station** is used to edit a single signal. Multiple signals or signal combinations are not allowed in this section.

Continues on next page

Advanced settings - Edit geometry properties

Edit geometry properties

The **Edit Geometry Properties** tab provides the settings to change the properties for a selected model and variant.



	Property	Minimum Value(mm)	Value(mm)	Maximum Value(mm)
▶	L	200	300	600
	W	200	400	600
	H	20	50	200

xx1300000375

To edit Geometry properties:

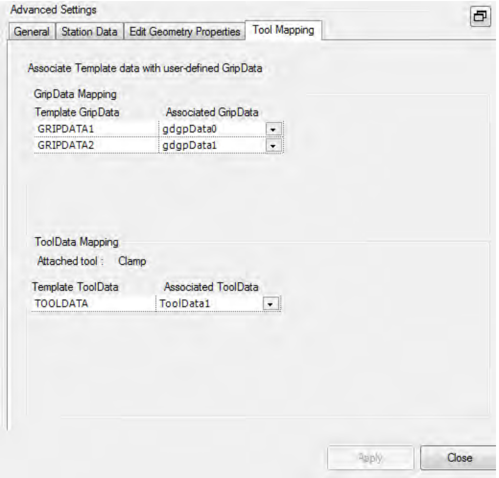
Action	Description
Update values in the **Value (mm)** column.	The new value should be within the pre-defined minimum and maximum value range. **Note** If the value you typed is not within the pre-defined range, an exclamation mark symbol is displayed.
Click **Update**. The updated geometry is displayed in the **3D Preview** section.	

Continues on next page

Advanced settings - Tool mapping

Tool mapping

The **Tool Mapping** tab provides the settings to choose `tooldata` and `gripdata`, to be associated with the corresponding station.



Associate Template data with user-defined GripData

GripData Mapping

Template GripData	Associated GripData
GRIPDATA1	gdgpData0
GRIPDATA2	gdgpData1

ToolData Mapping

Attached tool : Clamp

Template ToolData	Associated ToolData
TOOLDATA	ToolData1

xx1300000376

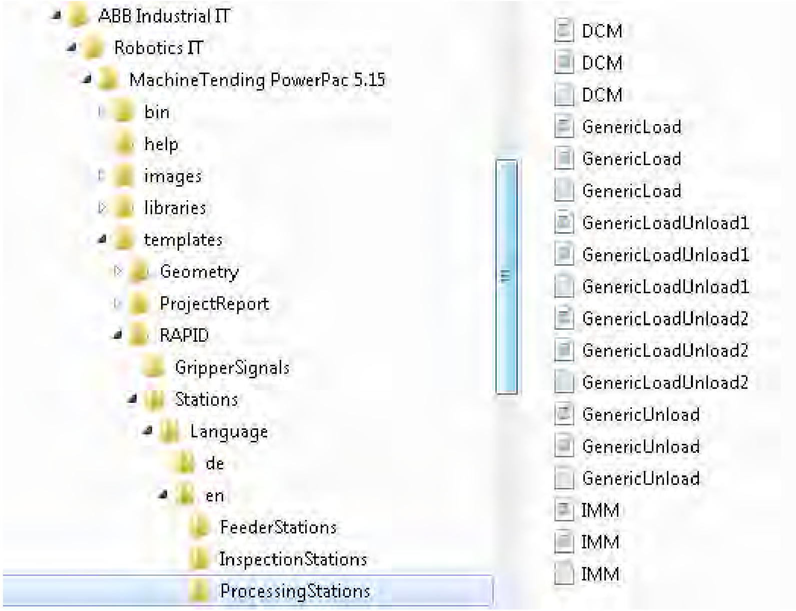
To edit tool mapping for a station:

Action	Description
Click on **Associated GripData** dropdown list.	Displays all the gripdata associated with the active tool attached to the robot.
Click on **Associated ToolData** dropdown list.	Displays all the tooldata associated with the active tool attached to the robot.
Click **Apply**.	The selected gripdata and tooldata information is stored and will be used in the station.mod template on synchronizing to VC. For more details refer the section *Program template on page 49*.

4.2.2.2 Program template

Template

Templates are pre-defined RAPID modules installed together with MTPP. In the Machine Tending PowerPac there are predefined RAPID templates for each station type. You can choose the template that closely represent the stations being added. Whenever a station is created using a template, the corresponding station module and station data are updated and the station signals are created.



xx1300000549

The RAPID program for a station is generated using a template. There is a default mapping between a station model and the program template.

Templates are categorized depending on the type of the station. Following are the different templates:

- **GenericLoad** - Contains data declarations, signals and routine content related to loading of a station only. Generally, an outfeeder will use a GenericLoad Template for its program.
- **GenericUnload** - Contains data declarations, signals and routine content related to unloading of a station only. Generally, an infeeder will use a GenericUnload Template for its program.
- **GenericLoadUnload1** - Contains data declarations, signals and routine content related to loading and unloading of a station. This is the most commonly used template in MTPP.
- **GenericLoadUnload2** - Contains data declarations, signals and routine content related to loading and unloading of a station, with some additional handshaking signals. In MTPP, a lathe station is associated with a GenericLoadUnload2 template.
- **IMM** - Customized template which caters to an IMM, containing signals (Euromap standards), data declarations, routines and functions.

Continues on next page

- **DCM** - Customized template which caters to a DCM, containing signals , data declarations, routines and functions.
- **Generic_PartControl** - Customized template for an inspection station with wait time and number of inspection signals as configurable parameters.
- **GenericPalletizing**: Customized template for a feeder station, containing configurable parameters for the robot to place the part in a palletizing pattern.
- **No Template** - When you select this option no targets are created for the station and no signals are associated with the station. because this option does not associate any template for a station.

The installation folder of MTPP includes the following files for each template:

File	Description
`<Template>.lib`	This is the RAPID file which describes the station handling. The RAPID module shall be created based on this file by replacing the `%xxxxx%` tags in the template with the station information.
`<Template>.cfg`	This the signal template which shall updated similar to the *.lib file and uploaded to the controller during Synchronize to VC.
`<Template>.xml`	The Station is a Smart Component. The `*.lib` file describes the relationship between the Station signals and the Smart component signals based on the logic of the station and the RAPID.

The Template library (`*.lib`) has several XML style tags that guide to add content as described in the following table:

```
<STATIONMODULE>
<DESCRIPTION>
<\DESCRIPTION>
<DISCLAIMER>
<\DISCLAIMER>
<RULES>
<\RULES>
<\STATIONMODULE>

<MOTION>
<POSITION>
<\POSITION>
<GRIPDATA>
<\GRIPDATA>
<STATIONDATADECLARATION>
<\STATIONDATADECLARATION>
<DECLARATIONS>
<\DECLARATIONS>
<CODE>
<\CODE>
<\MOTION>
<HOMERUN>
<CODE>
<\CODE>
</HOMERUN>
```

Continues on next page

Section	Description

`<STATIONMODULE>`
`<\STATIONMODULE>`
	Describes the RAPID module.

`<DESCRIPTION>`
`<\DESCRIPTION>`
	Describes the station information. This is displayed as a tool-tip in the ‘Configure Station’ feature.

`<DISCLAIMER>`
`<\DISCLAIMER>`
	This is the disclaimer section. The content below is retained after the *.mod file is created.

`<RULES>`
`<\RULES>`
	This section describes the rules about the content that are included and the sections that are removed or updated when the `*.mod` file is created.

`<MOTION>`
`<\MOTION>`
	This section declares all the content that will be used in the `<STATIONMODULE>` section and the content to be created in the Movement module.
	The content in this section is deleted when the `*.mod` file is created.

`<POSITION>`
`<\POSITION>`
	All the positions that are used in the template are declared in this section.

`<GRIPDATA>`
`<\GRIPDATA>`
	All the different GripData names used in the template are declared in this section.
	In MTPP, GripData can be created and mapped to the corresponding template declarations in the Tool mapping section of “Configure Stations”.
	The GripData declaration itself is created in a separate module from MTPP.

`<STATIONDATADECLARATION>`
`<\STATIONDATADECLARATION>`
	The StationData arguments are declared in this section.

`<DECLARATIONS>`
`<\DECLARATIONS>`
	All the RobTargets to be created in a separate Movement module are declared in this section.

`<CODE>`
`<\CODE>`
	All the movement routines to be defined in the Movement module are declared in this section.
	The content in this section is deleted when the *.mod file is created.

`<HOMERUN>`
`<\HOMERUN>`
	This section describes the content to be created in the Home run module (MT_HOME_User.mod).

`<CODE>`
`<\CODE>`
	The strategy for the positions to move to the Home position are declared in this section.

The Template XML file describes the behavior of the station smart component. It includes the:

- Initial state of the station (signal state), whether Open or Close, values. This represents a state of the station during simulation start.
- Description of station signals to be set or reset based on the working of the station.

To create a new template,

- Copy and modify an existing template (`*.lib`) file.
- Create the corresponding cfg file.
- Create XML file which describes the signal relationship between station SC and controller.

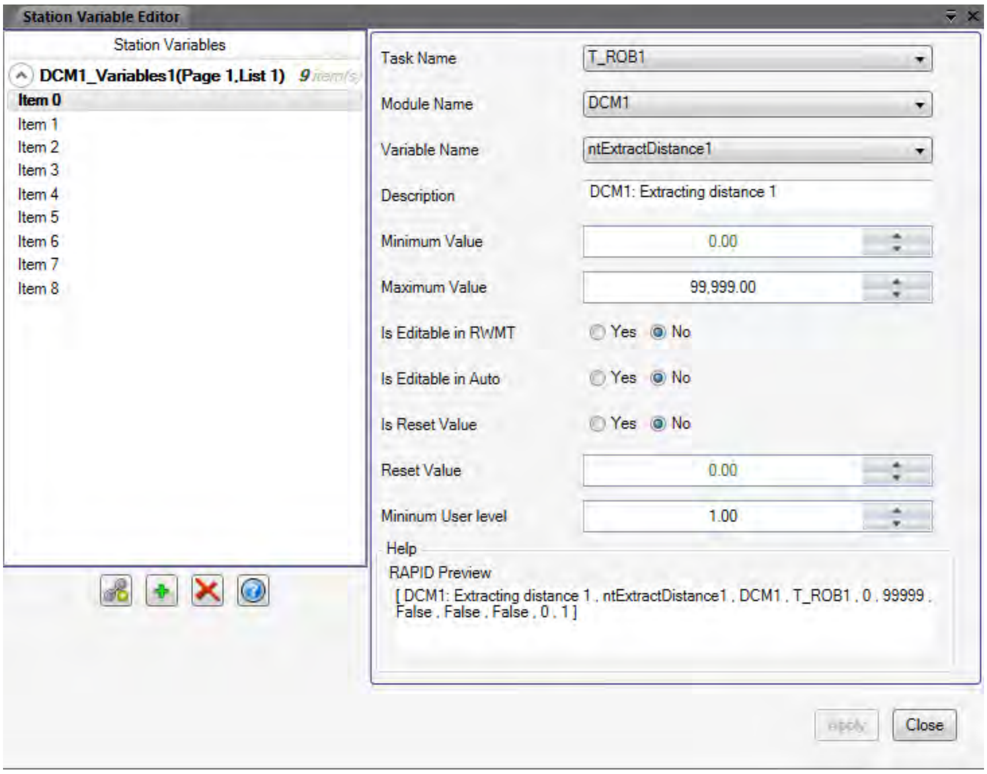
Continues on next page

- In MTPP, this template is listed in Configure Stations.

4.2.2.3 Data editors

Station variables editor

Right click on the station node and select **Configure** > **Station Variables**. The **Station Variable Editor** window is displayed.



Task Name: T_ROB1
Module Name: DCM1
Variable Name: ntExtractDistance1
Description: DCM1: Extracting distance 1
Minimum Value: 0.00
Maximum Value: 99,999.00
Is Editable in RWMT: ○ Yes ● No
Is Editable in Auto: ○ Yes ● No
Is Reset Value: ○ Yes ● No
Reset Value: 0.00
Mininum User level: 1.00

RAPID Preview
[DCM1: Extracting distance 1 , ntExtractDistance1 , DCM1 , T_ROB1 , 0 , 99999 , False , False , False , 0 , 1]

xx1400002486

The **Station Variable Editor** window displays the variable lists corresponding to the program template associated with the selected station. The variables associated with a variable list are diplayed below it. Click on a variable to display its corresponding parameters. You can edit the parameters according to your requirement.

Note

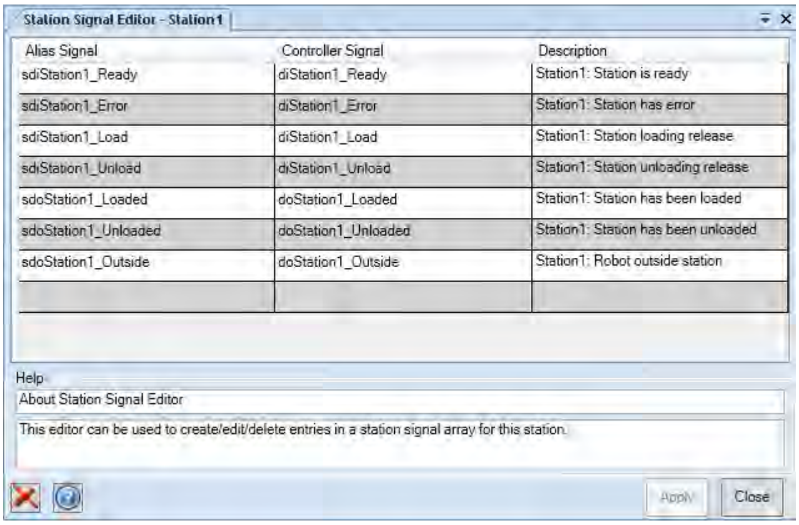
The variable name parameter displays the station variables that are declared and used in the selected RAPID module. The variable name you select in the **Variable Name** field should not be used for other variables.

Click to add a new variable list and click ✚ to add a new variable to a selected variable list. Click ✖ to delete a selected variable. If all the variables in a variable list are deleted, the variable list is automatically removed.

Continues on next page

Station signals editor

Right click on the station node and select **Configure** > **Station Signals**. The **Station Signal Editor** window is displayed.



Station Signal Editor - Station1

Alias Signal	Controller Signal	Description
sdiStation1_Ready	diStation1_Ready	Station1: Station is ready
sdiStation1_Error	diStation1_Error	Station1: Station has error
sdiStation1_Load	diStation1_Load	Station1: Station loading release
sdiStation1_Unload	diStation1_Unload	Station1: Station unloading release
sdoStation1_Loaded	doStation1_Loaded	Station1: Station has been loaded
sdoStation1_Unloaded	doStation1_Unloaded	Station1: Station has been unloaded
sdoStation1_Outside	doStation1_Outside	Station1: Robot outside station

Help

About Station Signal Editor

This editor can be used to create/edit/delete entries in a station signal array for this station.

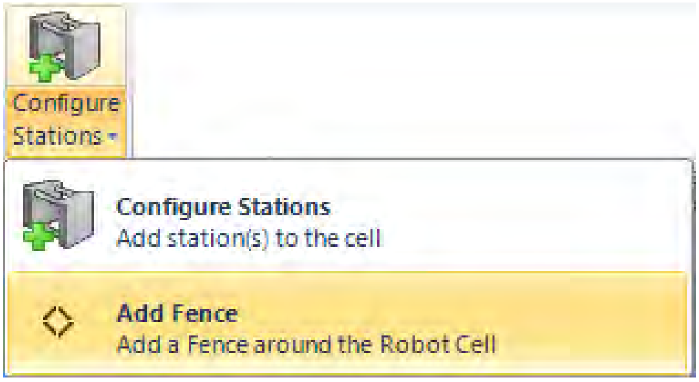
xx1400002487

The **Station Signal Editor** window displays a list of alias signals and the corresponding controller signals. You can double click on a **Alias Signal** cell or a **Controller Signal** cell to display a signal drop-down list. Select a signal according to your requirement to change the mapping. You can map an alias signal only with one controller signal and vice versa. Double click on the **Description** cell to edit and change the description of a mapping. When you click the **Apply** button the changes are saved.

Use the last blank row of the list to define a new mapping. When you click the **Apply** button the mapping is created and a new blank row is added as the last row.

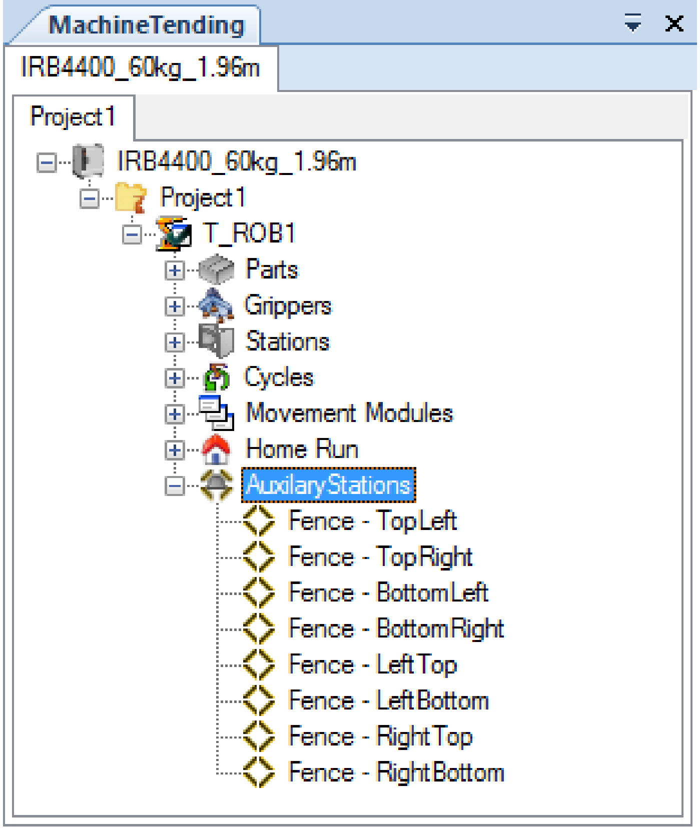
To delete a mapping, select the mapping and click the ☒ button.

4.2.2.4 Add fence



xx1300000499

Click **Configure Stations** > **Add Fence** in the **Build Cell** group to add a fence around the robot cell. The fence segments are listed under the project browser tree.



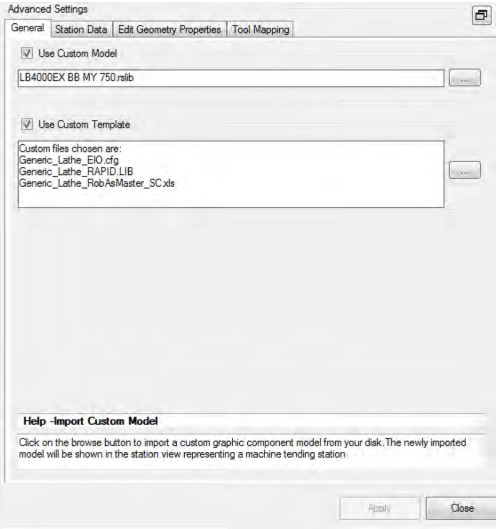
xx1300000550

The fence segments can be individually selected and deleted.

4.2.2.5 Custom models

Using custom models

You can select the custom station models and RAPID templates from the **Advanced Settings** tab in Configure Stations.



☑ Use Custom Model

LB4000EX BB MY 750.rslib

☑ Use Custom Template

Custom files chosen are:
Generic_Lathe_EIO.cfg
Generic_Lathe_RAPID.LIB
Generic_Lathe_RobAsMaster_SC.xls

Help -Import Custom Model

Click on the browse button to import a custom graphic component model from your disk.The newly imported model will be shown in the station view representing a machine tending station

xx1300000373

The following steps describe the process for selecting the custom station models and RAPID templates.

	Action	Description
1	Browse to select the custom Station library (`*.rslib`)	
2	Browse to select its corresponding templates.	**Note** All the three template files must be selected. The following are the available templates: • RAPID Template (`*.lib`): For handling the stations. • I/O configuration (`*.cfg`): For handling the station signals. • Signal mapping excel (`*.xls`): For creating the station smart component.
3	Click **Apply**.	

A custom station model can be used together with a default template. The station smart component is created based on the templates. You need to manually verify the station smart component for desired behavior.

Continues on next page

The added custom stations are auto-placed at an offset from the robot base frame and you should manually reposition this custom station to a desired position.

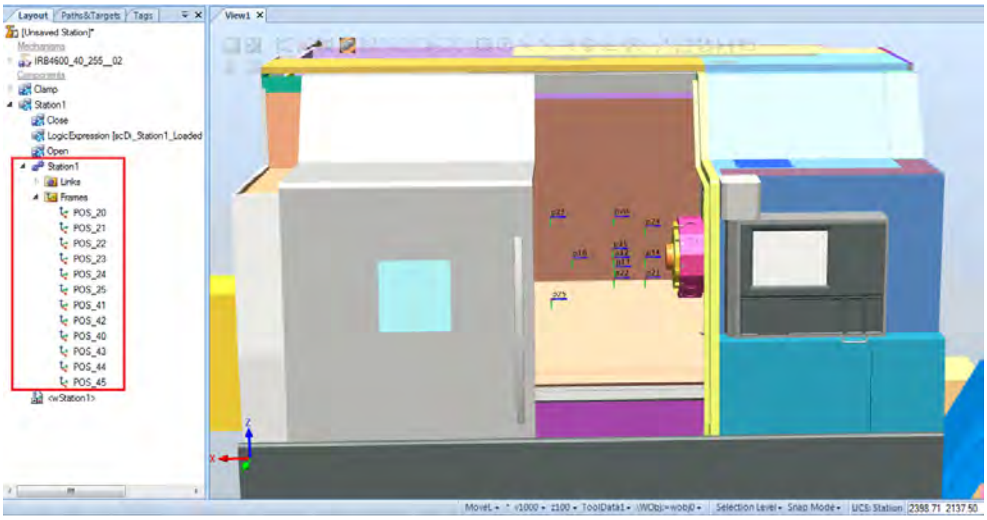
Tip

If a custom station model is selected to be used, then it is recommended to create its corresponding signal mapping excel (`*.xls`) to ensure that the smart component behaves as expected, based on the mechanism. Use the same file name for the custom template files of a particular station.

Following are some of the conditions for the selected custom model libraries:

- The model must be a mechanism with utmost one mechanism.
- It cannot be a smart component. The default behavior of the smart component is defined, based on the selected signal mapping excel. Hence it is not recommended to use a pre-defined smart component.
- The model is associated with its corresponding templates.
- The mechanism of the selected model must have Frames defined. The RobTargets are created at the Frame locations based on the RAPID templates.
- The frame naming convention:
  - The frames should be named as `POS_#` where # is an integer. For example, `POS_10`, `POS_11`, `POS_12`, and so on.
  - The sequence and count of frames should follow the target definitions in Station RAPID template. That is, there should be same number of frames in the model as the number of targets in the RAPID template.

Example, Okuma Lathe Model containing 12 frames and corresponding targets.



xx1400001098

Note

If the number of targets in a station template exceeds 10, then select the sation number of the next station so that there are no conflicts in the target numbering.

4.2.2.6 Excel approach for creating the smart component for stations

4.2.2.6.1 Creating smart component for stations

Introduction

Overview

In MTPP, the smart component of a station is created based on the mechanism and an Excel (*.xls) file which defines the following:

- Input and output signals exchanged between the station and the controller.
- Different States of the mechanism (for example, Machine Door Open, Machine Door Close)
- Relation between the inputs, outputs, and machine states.

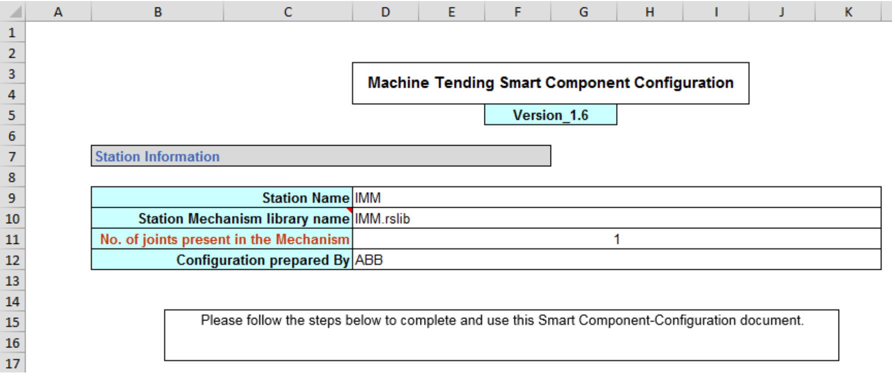
The excel file consists of the following worksheets:

- **Admin Sheet**
- **Step1_Define_Signals**
- **Step2_Define_Machine_States**
- **Step3_Define_Behavior**
- **Template revision History**

Admin sheet

Overview

The admin sheet contains the step-by-step information about using the excel file and the information about the associated machine model.



Machine Tending Smart Component Configuration

Version_1.6

Station Information

Station Name	IMM
Station Mechanism library name	IMM.rslib
No. of joints present in the Mechanism	1
Configuration prepared By	ABB

Please follow the steps below to complete and use this Smart Component-Configuration document.

xx1400001111

Labels	Description
Station Name	Displays the name of the station with which the excel template is used.
Station Mechanism library name	Displays the name of the mechanism for which the excel template is being prepared.
No. of Joints present in the Mechanism	Type the number of joints present in the library. If there are no joints then type ‘0’. This field should not be left empty.
Configuration prepared By	Displays the name of the administrator who prepared template.

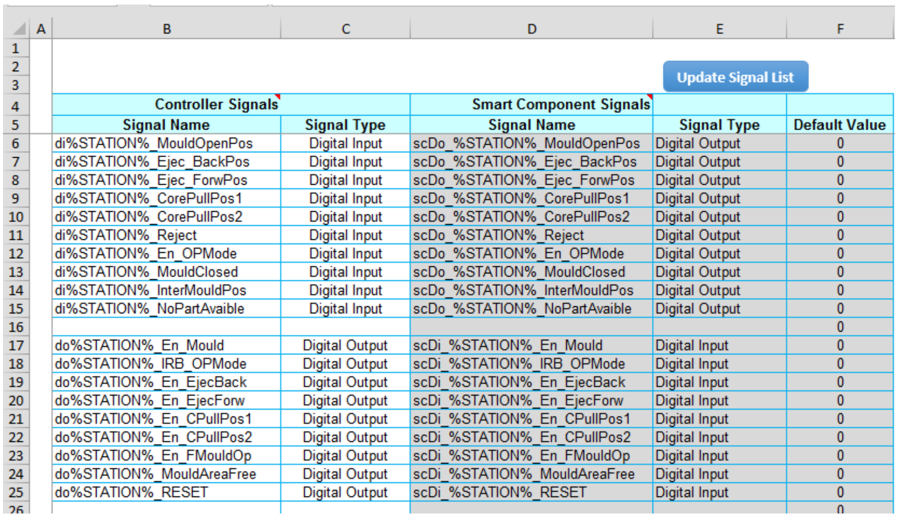
Continues on next page

Step1_Define_Signals

Overview

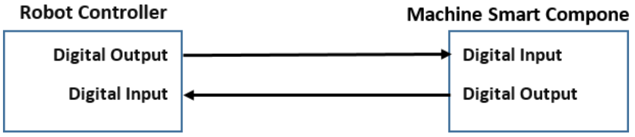
The **Step1_Define_Signals** worksheet contains a table consisting of mapping of controller signals and the respective smart component signals.

Enter the Controller Signal Name and the Controller Signal Type in the respective columns. The corresponding Smart Component signals are generated automatically.



	A	B	C	D	E	F
1						
2					Update Signal List	
3						
4		Controller Signals		Smart Component Signals		
5		Signal Name	Signal Type	Signal Name	Signal Type	Default Value
6		di%STATION%_MouldOpenPos	Digital Input	scDo_%STATION%_MouldOpenPos	Digital Output	0
7		di%STATION%_Ejec_BackPos	Digital Input	scDo_%STATION%_Ejec_BackPos	Digital Output	0
8		di%STATION%_Ejec_ForwPos	Digital Input	scDo_%STATION%_Ejec_ForwPos	Digital Output	0
9		di%STATION%_CorePullPos1	Digital Input	scDo_%STATION%_CorePullPos1	Digital Output	0
10		di%STATION%_CorePullPos2	Digital Input	scDo_%STATION%_CorePullPos2	Digital Output	0
11		di%STATION%_Reject	Digital Input	scDo_%STATION%_Reject	Digital Output	0
12		di%STATION%_En_OPMode	Digital Input	scDo_%STATION%_En_OPMode	Digital Output	0
13		di%STATION%_MouldClosed	Digital Input	scDo_%STATION%_MouldClosed	Digital Output	0
14		di%STATION%_InterMouldPos	Digital Input	scDo_%STATION%_InterMouldPos	Digital Output	0
15		di%STATION%_NoPartAvaible	Digital Input	scDo_%STATION%_NoPartAvaible	Digital Output	0
16						0
17		do%STATION%_En_Mould	Digital Output	scDi_%STATION%_En_Mould	Digital Input	0
18		do%STATION%_IRB_OPMode	Digital Output	scDi_%STATION%_IRB_OPMode	Digital Input	0
19		do%STATION%_En_EjecBack	Digital Output	scDi_%STATION%_En_EjecBack	Digital Input	0
20		do%STATION%_En_EjecForw	Digital Output	scDi_%STATION%_En_EjecForw	Digital Input	0
21		do%STATION%_En_CPullPos1	Digital Output	scDi_%STATION%_En_CPullPos1	Digital Input	0
22		do%STATION%_En_CPullPos2	Digital Output	scDi_%STATION%_En_CPullPos2	Digital Input	0
23		do%STATION%_En_FMouldOp	Digital Output	scDi_%STATION%_En_FMouldOp	Digital Input	0
24		do%STATION%_MouldAreaFree	Digital Output	scDi_%STATION%_MouldAreaFree	Digital Input	0
25		do%STATION%_RESET	Digital Output	scDi_%STATION%_RESET	Digital Input	0
26						0

xx1400001112

Labels	Description
Controller Signals	This column is for defining the controller signals pertaining to the station. Type the controller signal name in the **Signal Name** column and choose the corresponding signal type in the **Signal Type** column. See the section *Rules for filling the controller signal list on page 60* while entering the details.
Smart Component Signals	For the selected controller signals, the corresponding smart component signals are created automatically as indicated below.  Robot Controller Machine Smart Compone Digital Output Digital Input Digital Input Digital Output xx1400001113 The digital output from a controller is received by the machine smart component as a digital input. Similarly, the digital output from a machine smart component is received by the robot controller as a `Digital Input`. The Default Value column assigns the initial value to the smart component signal. The value can be either ‘0’ or ‘1’.
Update Signal List	After configuring all the desired signals, click this button to automatically update the list of signals.

Continues on next page

Note

Do not modify the greyed-out columns under the **Smart Component Signals** section. If the smart component signals are not automatically generated, click the **Update Signal List** button.

Rules for filling the controller signal list

Following are the rules for filling the controller signal list:

- The digital input signals must be defined first followed by the digital output signals separated by an empty row.
- The digital input signals must be preceded by **`di%STATION%_`** and the digital output signals must be preceded by **`do%STATION%_`**.
- The controller signal names must match those contained in the controller signal configuration file (`*.cfg`).
- After the digital output signals of the controller are defined, it is mandatory to configure the digital output signal **`do%STATION%_RESET`**. This is a RobotStudio station signal used during production simulation to set the machine to initial state.
- Do not manually insert new rows or delete the existing rows. You are allowed to delete the contents of a row but not the row itself.

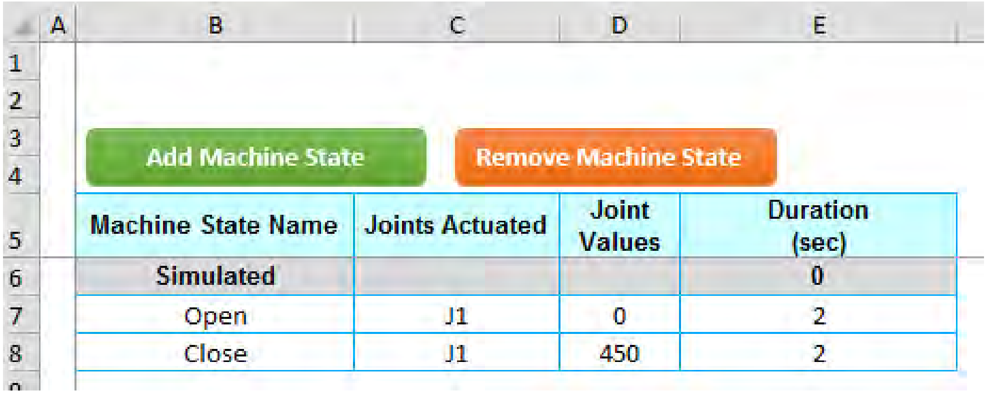
Step2_Define_Machine_States

Overview

A machine can have several states which are identified by joint positions. Example of machine states: Door Open, Door Close.

The **Step2_Define_Machine_States** worksheet allows to define Machine States with the corresponding values for the joints and the time taken to move the joint to the desired position.

The number of joints associated with a state is defined in the **Admin** worksheet by the **No. of Joints present in the Mechanism** field.



Machine State Name	Joints Actuated	Joint Values	Duration (sec)
Simulated			0
Open	J1	0	2
Close	J1	450	2

xx1400001102

Continues on next page

In the example shown in the figure, there are two states configured apart from the default **Simulated** state.

- For the **Open** state, the joint `J1` is actuated. Joint ‘J1’ moves to the ‘0'mm position in a duration of 2 seconds.
- For the **Close** state, the joint `J1`’ is actuated. Joint ‘J1’ moves to the‘450'mm position in a duration of 2 seconds.

To configure the machine states:

Action	Description
Click **Add Machine State** .	News rows are created. The number of rows created depends on the **No. of Joints present in the Mechanism** defined in the Admin worksheet.
In the **Machine State Name** column type the state name.	
For each joint in the machine state, type the joint values and the duration (in sec) in which it the joint has to move.	See the section *Rules for filling the table on page 61* while entering the details.

Rules for filling the table

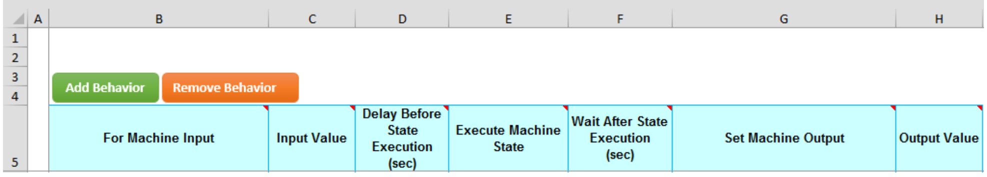
The following are the rules for filling this table:

- The default state **Simulated** should not be modified.
- The values in the **Joint Actuated** field are automatically filled and it should not be altered.
- All the joint values must be integers.
- To remove a machine state, select the Machine State Name cell and click the **Remove Machine State** button.
- Do not insert or delete rows manually. To add or remove a state, the respective buttons must be used.

Step3_Define_Behavior

Overview

The **Step3_Define_Behavior** worksheet allows to configure the machine response for inputs from the controller. The machine response can be a signal output, a joint movement or a combination of both.

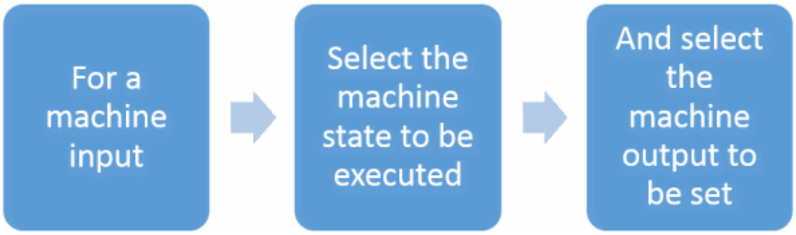


For Machine Input	Input Value	Delay Before State Execution (sec)	Execute Machine State	Wait After State Execution (sec)	Set Machine Output	Output Value

xx1400001103

Continues on next page

The following workflow should be considered while entering the details:


For a machine input

Select the machine state to be executed

And select the machine output to be set

xx1400001104

To configure the machine behavior:

	Action	Description
1	Click **Add Behavior**	Create a set of rows for a machine input.
2	Click on the newly added cell in the **For Machine Input** column and select the machine input signal	The machine behavior is to be configured for this input. A machine input can trigger several outputs. Each output is configured in a single row. The collection is called a **Behavior set** for the input signal.
3	For the selected machine input signal choose an input value.	
4	Select the machine state to be executed.	The machine states configured in the previous worksheet are available for selection. Choose a state from this drop-down list. If there is no joint movement configured/expected for the triggered input, select the machine state as **Simulated**.
5	Select the machine output signal to be set.	The machine output is triggered after the machine state is executed. Several output signals can be triggered simultaneously after a machine state is executed. In this scenario do not select the machine state for each output.
6	For the selected signal choose an output value.	
	Type a delay in the **Delay Before State Execution (sec)** column. Type a delay in the **Wait After State Execution (sec)** column.	The machine state is executed after this delay when an input is received. The machine output is set after this delay following the machine state execution.

If a machine output depends on a combination of machine inputs, then this can be configured by typing the logic expression. The logic expression input should be separted by a space after each element. For example, `( SingalA AND SignalB ) OR ( SignalC AND SignalD NOT )`

When using the above expression as the machine input, this input is 1 when SignalA and SignalB are high or when Signal C is high and Signal D is low. The 'AND', 'OR', 'NOT', and 'XOR' logic are used for framing the expression.

Rules for using the sheet

The following rules are to be followed while using this sheet:

- Every machine state or output to be set should start from the first row of the behavior set for a machine input.

Continues on next page

- Do not copy-paste entries in the **Execute Machine State**, **Set Machine Output**, and the **Input Value**, and **Output Value** columns. Choose only from the drop-down lists.
- The last behavior set in this worksheet should be the station RESET condition which defines the initial state of the machine. The input signal should be **`scDi_%STATION%_RESET`**. This signal is triggered from the Production dialog. Only the machine ouput signals which are to be set the value '1' are to be selected in the RESET behavior. All other machine output signals shall be set to then value '0' when the RESET is triggered.
- If a machine input requires more outputs to be triggered, then insert a row within the behavior set of the Machine input.
- When choosing a Machine input, it is recommended to choose from the drop-down list. The exception is when a signal combination has to be configured in which case you have to type the conditions.
- If a machine state execution triggers several output signals, then do not choose the machine state for each of these output signals. The state selected initially is applicable for all the following output signals (This rule is also applicable for the simulated state.)
- The delay before execution is only applicable for a row where Machine state is selected. (This rule is also applicable for the simulated state.)
- Empty cells in the columns **Delay Before State Execution** and **Wait After State Execution** are considered as ‘0’ seconds delay.

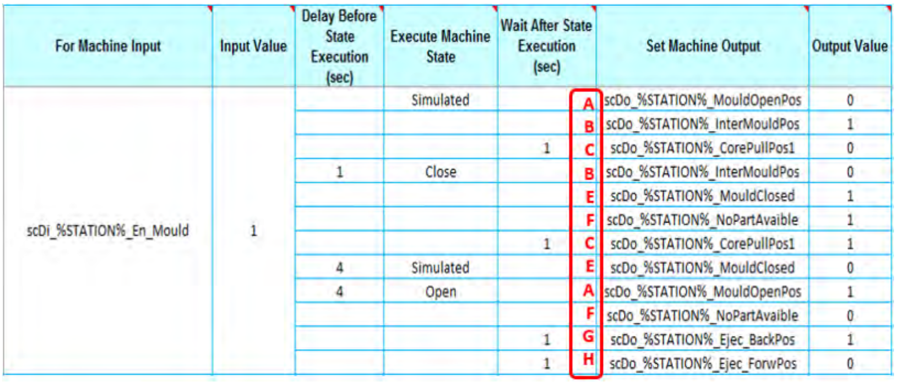
Steps for configuring the machine behavior

Consider the IMM station (Injection Molding Machine) which has only one joint for the door. It has two user-defined machine states namely ‘Open’ for the door open state and ‘Close’ for the door close state. The station also has the default machine state called ‘Simulated’.

The processing of the injection molding machine is started by setting the input signal `scDi_%STATION%_En_Mould` to the ‘High’ state. Then the mold closes and the molding process starts. The mold opens after the process. Certain actions takes place and signals are set or reset during this process. The behavior set for this process is shown in the following figure along with the pictorial representation. Following are the assumptions considered while preparing this behavior set:

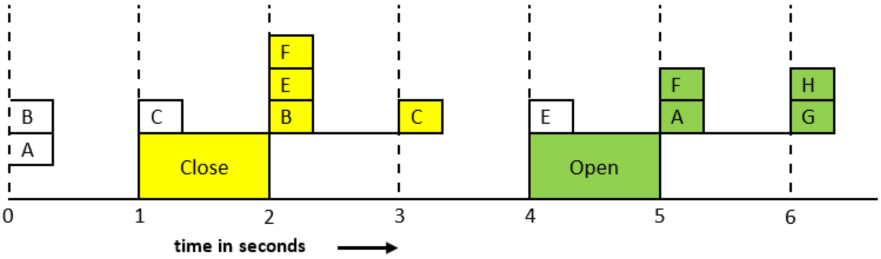
- Machine state **Close** takes 1 second
- Machine state **Open** takes 1 second
- Molding time is 2 seconds

Continues on next page



For Machine Input	Input Value	Delay Before State Execution (sec)	Execute Machine State	Wait After State Execution (sec)		Set Machine Output	Output Value
scDi_%STATION%_En_Mould	1		Simulated		A	scDo_%STATION%_MouldOpenPos	0
					B	scDo_%STATION%_InterMouldPos	1
				1	C	scDo_%STATION%_CorePullPos1	0
		1	Close		B	scDo_%STATION%_InterMouldPos	0
					E	scDo_%STATION%_MouldClosed	1
					F	scDo_%STATION%_NoPartAvaible	1
				1	C	scDo_%STATION%_CorePullPos1	1
		4	Simulated		E	scDo_%STATION%_MouldClosed	0
		4	Open		A	scDo_%STATION%_MouldOpenPos	1
					F	scDo_%STATION%_NoPartAvaible	0
				1	G	scDo_%STATION%_Ejec_BackPos	1
				1	H	scDo_%STATION%_Ejec_ForwPos	0

xx1400001110



xx1400001108

After the input trigger signal `scDi_%STATION%_En_Mould` is set to the value ‘1’, the following sequence of operations take place:

1. First Step – The Simulated machine state is executed immediately (0 seconds delay) after the input signal value becomes ‘1’. This state triggers the following output signals:
   - Signal **A** is set to the value ‘0’.
   - Signal **B** is set to the value ‘1’.
   - Signal **C** is set to the value ‘0’ after 1 second delay.
2. Second Step – After 1 second delay specified in the **Delay Before State Execution** column, the Close machine state is executed. The door closing time is 1 second. After the machine door is closed, the following output signals are triggered:
   - Signal **B** is set to the value ‘0’.
   - Signal **E** is set to the value ‘1’.
   - Signal **F** is set to the value ‘1’.
   - Signal **C** is set to the value ‘1’ after 1 second delay.
3. Third Step – The Close machine state is executed after a 1 second delay. The door closing time is 1 second and the processing time is 2 seconds. So from the instance of the input trigger signal being set ‘High’, the machine waits for 4 seconds before executing the next step. Hence, after 4 seconds

Continues on next page

(1+2+1) delay, the following machine states are executed simultaneously without any delay:

- The Simulated machine state is executed. This state triggers the following output signal:
    - Signal **E** is set to the value ‘0’.
- The Machine ‘Open’ state is executed. The door opening time is 1 second. After the door open state is executed, the following actions take place simultaneously:
    - Signal **A** is set to the value ‘1’.
    - Signal **F** is set to the value ‘0’.
    - Signal **G** is set to the value ‘1’ after 1 second delay.
    - Signal **H** is set to the value ‘0’ after 1 second delay.

Template revision history

Overview

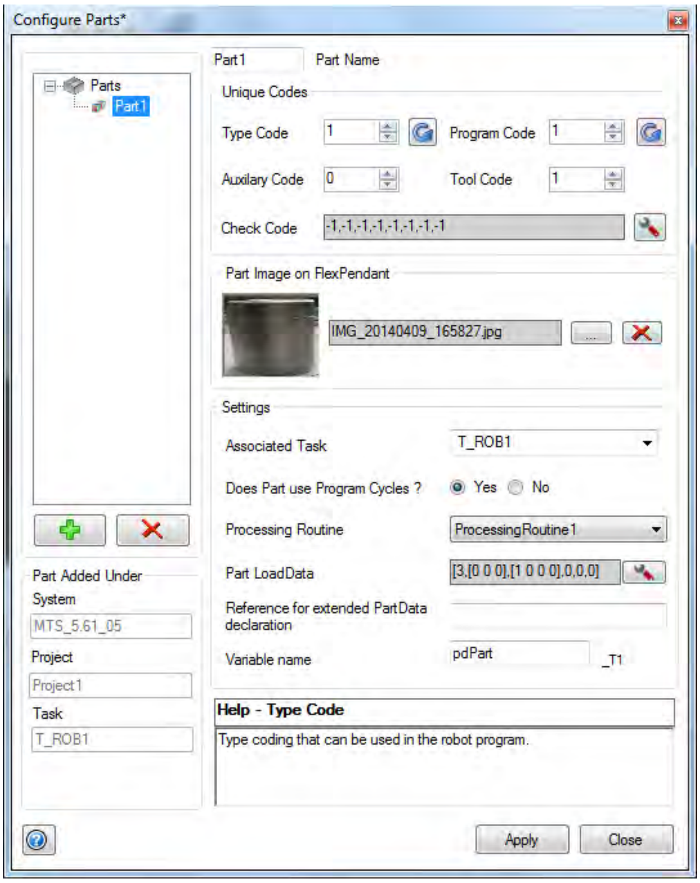
The **Template revision history** worksheet Contains a table for tracking the changes done in the excel file.

4.3 Part data

4.3.1 Configure parts

Introduction

The **Configure Parts** window allows you to configure part data.



Configure Parts*

Parts
Part1

Part1 Part Name

Unique Codes

Type Code: 1
Program Code: 1
Auxilary Code: 0
Tool Code: 1
Check Code: -1,-1,-1,-1,-1,-1,-1,-1

Part Image on FlexPendant: IMG_20140409_165827.jpg

Settings

Associated Task: T_ROB1
Does Part use Program Cycles ? ● Yes ○ No
Processing Routine: ProcessingRoutine1
Part LoadData: [3,[0 0 0],[1 0 0 0],0,0,0]
Reference for extended PartData declaration:
Variable name: pdPart _T1

Part Added Under
System
MTS_5.61_05
Project: Project1
Task: T_ROB1

Help - Type Code
Type coding that can be used in the robot program.

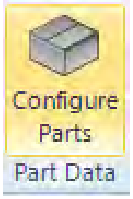
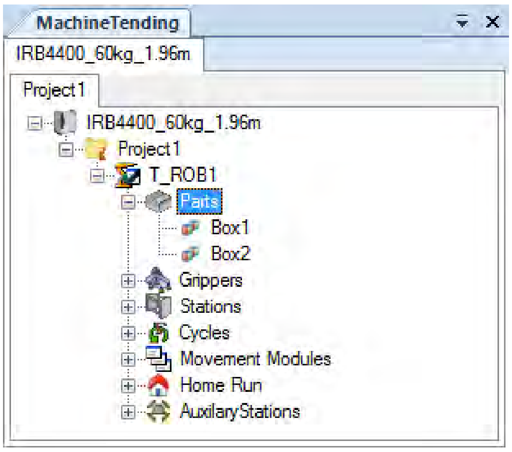
xx1300000363

A part is any finished or semi finished output of a production cycle in machine tending cell. There can be one or more parts produced by a cell (a set of stations and a robot). Configuring a part is primarily to configure partdata (a RobotWare Machine Tending data type), which conceptualizes a part in the real world by considering its properties like name, unique code, other codes, associated image, and so on. A part may have a processing routine associated with it, which in turn is used to execute a cycle (a series of operations to produce that part).

Continues on next page

Configuring partdata

To configure `partdata`:

	Action	Description
1	Click **Configure Parts**. The **Configure Parts** window is displayed.	 xx1300000498
2	Click + to add a part. The part is added to the **Parts** section.	Click ✕ to delete a selected part from the **Parts** section. To customize each part, select a part and edit the part settings.
3	Click **Apply**. The part is created and listed under the Browser tree.	 MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Box1 Box2 Grippers Stations Cycles Movement Modules Home Run AuxilaryStations xx1300000500

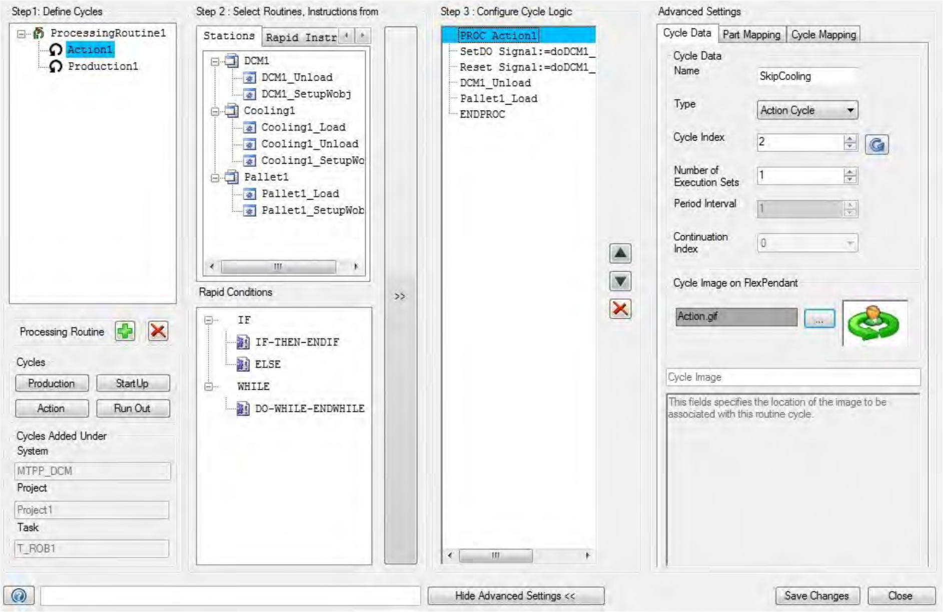
`partdata` is the RobotWare Machine Tending data type associated with a part. For more details about `partdata` see *Application manual - RobotWare Machine Tending*.

4.4 Programming

4.4.1 Define Cycles

Introduction

The **Define Cycle** window allows you to configure cycle sequences for serving different stations.



```
PROC Action1
SetDO Signal:=doDCM1_
Reset Signal:=doDCM1_
DCM1_Unload
Pallet1_Load
ENDPROC
```

System: MTPP_DCM
Project: Project1
Task: T_ROB1

Cycle Data Name: SkipCooling
Type: Action Cycle
Cycle Index: 2
Number of Execution Sets: 1
Period Interval: 1
Continuation Index: 0
Cycle Image on FlexPendant: Action.gif
Cycle Image: This fields specifies the location of the image to be associated with this routine cycle.

xx1300000425

The cycle routines are configured using the Load/Unload routines from the station templates. The routines are saved in a separate module within the project.

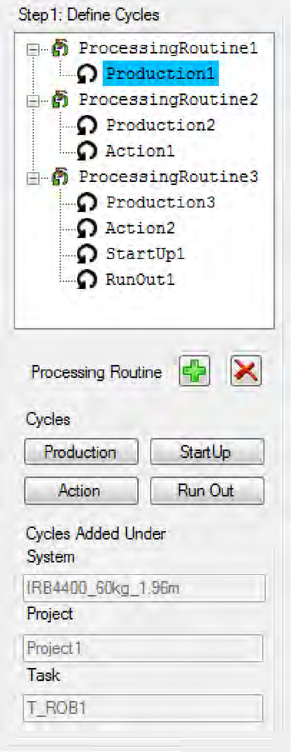
The necessary information needed to select and execute the cycle from the FlexPendant is also configured from this window.

Workflow

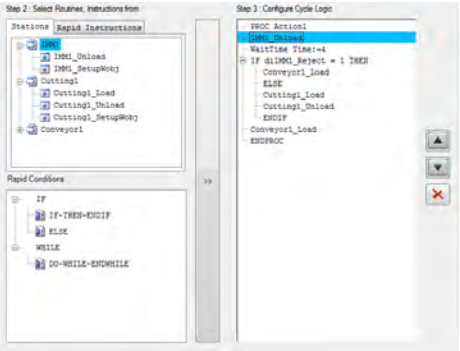
Use the following procedure to configure cycle sequences of serving different stations as RAPID cycle routines in a cycle module.

	Action	Description
1	Click **Define Cycles**. The **Define Cycles** page is displayed.	

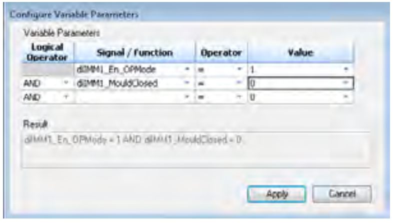
Continues on next page

	Action	Description
2	Click **Configure Define Cycles**. **The Step 1:** Define Cycles section is for creating the processing routines and the corresponding cycles routines. Prerequisite for this step is that at least 1 part is defined in the Project. To configure Define Cycles: 1 Click ✚. A processing routine is added to the list. **Note** The number of processing routines that can be added is limited to the number of parts defined in MTPP. 2 Click the required cycle routine types in the Cycles section. The selected cycle routines are added under the processing routine. The `cycledata` list is updated for each cycle routine added (from Advanced settings).	 Step 1: Define Cycles ProcessingRoutine1 Production1 ProcessingRoutine2 Production2 Action1 ProcessingRoutine3 Production3 Action2 StartUp1 RunOut1 Processing Routine ✚ ✖ Cycles Production StartUp Action Run Out Cycles Added Under System IRB4400_60kg_1.96m Project Project1 Task T_ROB1 xx1300000426 The different types of cycles are available for selection in this Step, guide you to plan the cycles. For more information on the types of cycles, see *Cycle data on page 75*. **Note** You can change the cycledata details for each of these cycle routines. Click ✖ to delete a cycle routine or a processing routine. Deleting a processing routine also deletes it associated cycle routines.

Continues on next page

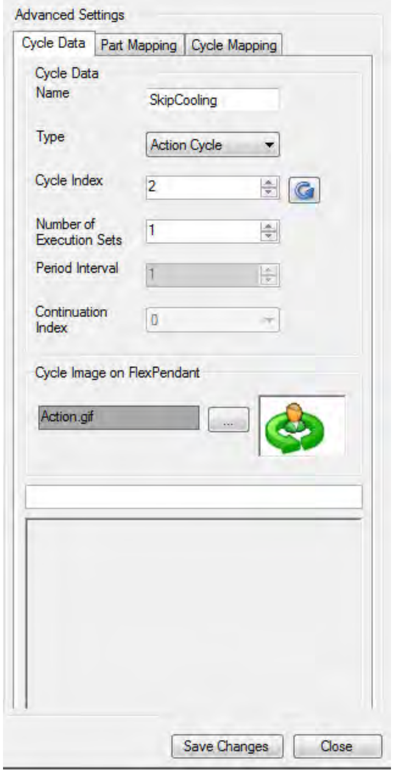
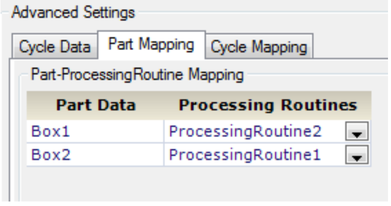
<table>
<tr><th></th><th>Action</th><th>Description</th></tr>
<tr>
<td>3</td>
<td>Configure Cycle Logic.
<b>Step 2 and Step 3:</b>
Presents the collection of:
• Station routines from the template Station modules
• Default RAPID instructions in RS
• RAPID conditions
To construct the cycle logic:
1 Select a routine / instruction / Conditions from the window
2 Click the arrow button.
The selected routine / instruction / condition is added to the <b>Configure Cycle Logic</b> section.
3 Repeat the preceding steps to build the cycle logic according to your requirement.</td>
<td>

Step 2 : Select Routines, Instructions from
Stations
Rapid Instructions
IMM1
IMM1_Unload
IMM1_SetupWobj
Cutting1
Cutting1_Load
Cutting1_Unload
Cutting1_SetupWobj
Conveyor1
Rapid Conditions
IF
IF-THEN-ENDIF
ELSE
WHILE
DO-WHILE-ENDWHILE
>>
Step 3 : Configure Cycle Logic

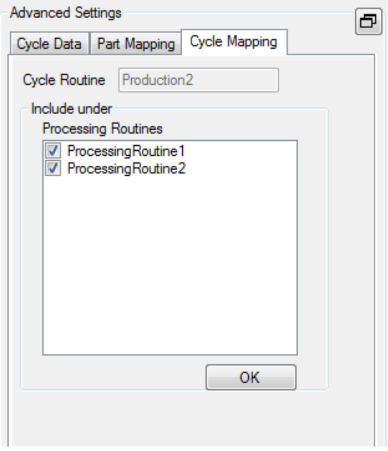
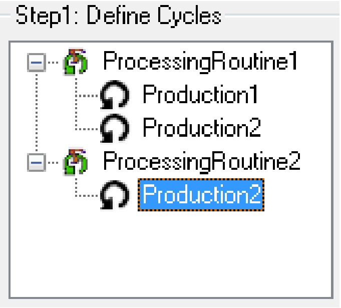
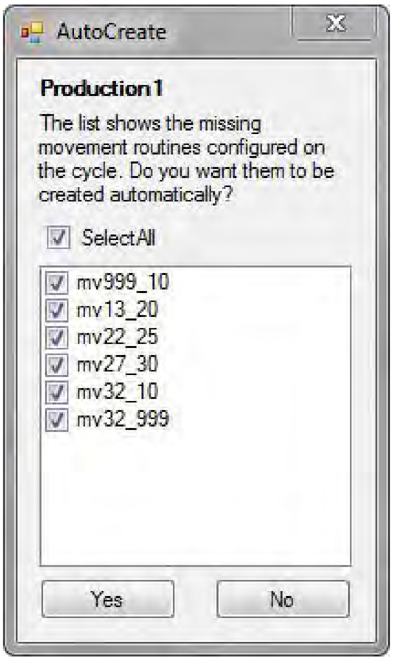
PROC Action1
IMM1_Unload
WaitTime Time:=4
IF diIMM1_Reject = 1 THEN
Conveyor1_load
ELSE
Cutting1_Load
Cutting1_Unload
ENDIF
Conveyor1_Load
ENDPROC
xx1300000432
The RAPID instruction selected can be configured in the same way as RobotStudio.
The IF / WHILE conditions can be configured as shown in the following image.

Configure Variable Parameters
Variable Parameters<table><tr><th>Logical Operator</th><th>Signal / Function</th><th>Operator</th><th>Value</th></tr><tr><td></td><td>diIMM1_En_OPMode</td><td>=</td><td>1</td></tr><tr><td>AND</td><td>diIMM1_MouldClosed</td><td>=</td><td>0</td></tr><tr><td>AND</td><td></td><td>=</td><td>0</td></tr></table>Result
diIMM1_En_OPMode = 1 AND diIMM1_MouldClosed = 0

xx1300002708</td>
</tr>
</table>

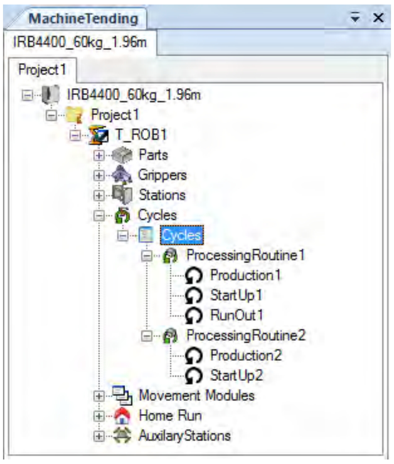
Continues on next page

<table>
<tr><th></th><th>Action</th><th>Description</th></tr>
<tr><td>4</td><td>Click <b>Show Advanced Settings</b>.
The <b>Advanced Settings</b> window is displayed in the following three tabs <b>Cycle Data</b>, <b>Part Mapping</b> and <b>Cycle Mapping</b>.</td><td>
Advanced Settings
Cycle Data | Part Mapping | Cycle Mapping
Cycle Data
Name SkipCooling
Type Action Cycle
Cycle Index 2
Number of Execution Sets 1
Period Interval 1
Continuation Index 0
Cycle Image on FlexPendant
Action.gif ...
Save Changes Close
xx1300000429
The <b>Cycle Data</b> tab allows you to configure the CycleData for the selected the cycle routine.
For more information on the parameters, refer to <i>Application manual - RobotWare Machine Tending</i> .

Advanced Settings
Cycle Data | Part Mapping | Cycle Mapping
Part-ProcessingRoutine Mapping
<table><tr><th>Part Data</th><th>Processing Routines</th></tr><tr><td>Box1</td><td>ProcessingRoutine2</td></tr><tr><td>Box2</td><td>ProcessingRoutine1</td></tr></table>
xx1300000521
The <b>Part Mapping</b> tab lists the part data and the corresponding processing routine. For more details, see <i>Configuring cycles on page 74</i>.
The processing routines mapping can be modified from this dialog.</td></tr>
</table>

Continues on next page

	Action	Description
5	To associate a cycle routine with other processing routines, select a cycle routine in the list and go to **Cycle Mapping** tab under Advanced Settings. Select the part processing routines from the list and click **OK**. **Note** Some cycle routines are common and can be executed with any of the selected part. Hence these routines can be called from any part processing routine. The **Cycle Mapping** tab allows you to configure this option.	 Advanced Settings Cycle Data Part Mapping Cycle Mapping OK Cycle Routine Production2 Include under Processing Routines ☑ ProcessingRoutine1 ☑ ProcessingRoutine2 xx1300000519 The list is updated as shown in the following image. Step1: Define Cycles ProcessingRoutine1 Production1 Production2 ProcessingRoutine2 Production2 xx1300000520 `Production2` routine is referenced in both Processing routines.
6	Click **Save Changes**. The **AutoCreate** window is displayed. It displays a set of movement routines based on the currently selected cycle logic that has been configured in **Step 3: Configure Cycle Logic**. **Note** The **AutoCreate** window generally lists only the missing movement routines that are needed for the robot to execute the cycle. In addition it also displays a movement routine from `HomePos(p999)` to the start point of the cycle logic. **Note** If the cycle type is `Continuous` or `Action`, the auto create window also displays the movement routine from the end point of the cycle logic to `HomePos(p999)`.	 **Production1** The list shows the missing movement routines configured on the cycle. Do you want them to be created automatically? ☑ SelectAll ☑ mv999_10 ☑ mv13_20 ☑ mv22_25 ☑ mv27_30 ☑ mv32_10 ☑ mv32_999 xx1500001666

Continues on next page

	Action	Description
7	Select the movement routines that need to be created automatically.	
8	Click **Yes**. The selected settings are saved and all the part processing routines and cycles are listed in the browser tree. The movement routines are also created. The RAPID module `cycle.mod` is created within the Project folder collection in the file system. **Note** There needs to be atleast one cycle routine below a processing routine while saving the cycle. **Note** The template for movement instructions used in the path is derived based on the configuration in the **Instruction Template** manager of the **Movements** window.	 MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Grippers Stations Cycles Cycles ProcessingRoutine1 Production1 StartUp1 RunOut1 ProcessingRoutine2 Production2 StartUp2 Movement Modules Home Run AuxilaryStations xx1300000502

Continues on next page

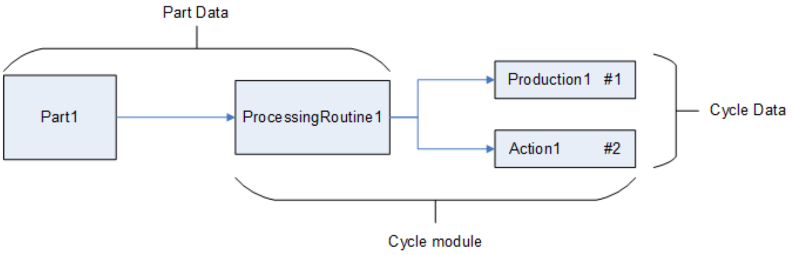
Define cycles - more information

Configuring cycles

An example for configuring cycles is given below:

```
CONST partdata
     part_T1:=["Part1","Pro-
     cessin-
     gRoutine1","",TRUE,1,4,3,1,[0,0,0,0,0,0,0,0],
     "Part1.GIF",[1,5,[0,0,0.001],[1,0,0,0],0,0,0],"pd-
     Part_T1"];
  PROC ProcessingRoutine1()
    !If startup cycle is reques-
         ted
    IF MT_CycleIndex=1 THEN
      Production1;
    !If a normal cycle is re-
         quested
    ELSEIF MT_CycleIndex=2 THEN
      NormalCycle;
    !If a runout cycle is re-
         quested
    ENDIF
  ENDPROC
  PROC Production1()
    IMM_Unload;
    AutoInsp1_Load;
    Cutter_Load;
    Pallet_Load;
  ENDPROC
  ...
ENDMODULE
```

```
!***********************************
! cycledata declaration
!***********************************
!Definition of the cycle list
TASK PERS cycledata
     MT_CycleList{20}:=
[
["Production
     cycle","",1,1,1,0,2,0],
["Normal
     cycle","",2,1,10,0,3,0],
["","",0,0,0,0,0,0],
["","",0,0,0,0,0,0],
...
...
]
```

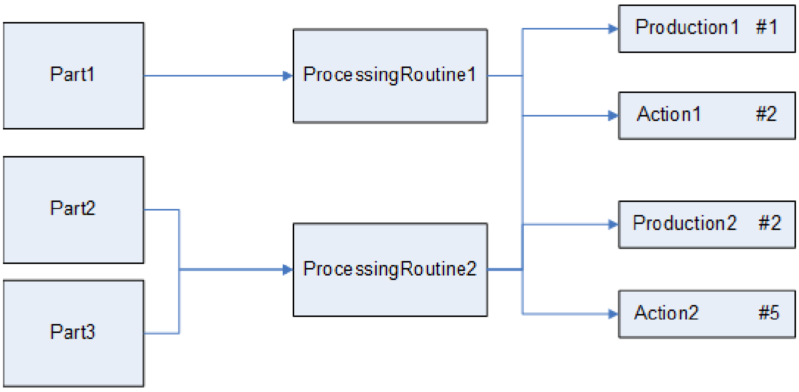


xx1300000424

When **Part1** is selected for production, the corresponding **ProcessingRoutine1** becomes active. The cycles **Production cycle** and **Action cycle** are available for selection from the FlexPendant. The selected cycle is executed by matching the cycle index defined in the **Cycle data**.

Continues on next page

There can be some extensions to this as illustrated in the following figure:



xx1300000402

Different parts can be associated with the same processing routine. Also a cycle routine can be associated with several ProcessingRoutines (**Production1** is called in both **ProcessingRoutine1** (previous image) and **ProcessingRoutine2**).

A machine tending cell can produce different parts. Each part can have different cycle routines. The cycle routines that can be executed for a part are collected through a Part Processing routine. A part is associated with its Part Processing routine through the datatype `PartData`.

Cycle data

The cycles can be selected from the RWMT interface on the FlexPendant through the datatype `CycleData` through the `MT_CycleList{20}`, which is present in the module `MT_Main`.

CycleData allows to configure different cycles types as described below.

Cycle type	Description
Continuous cycle	Continuous cycles are run without conditions. Normally, the execution is carried out till a **Halt after end of cycle** is requested. **Example**: Production with recurring process.
Counter cycle	Counter cycles are executed with the help of a counter, specifying the number of repetitions. **Example**: Batch finishing of 100 parts.
Action cycle	Action cycles are executed only on request from the user interface. The number of direct repetitions is specified. **Example**: Specific request for ejecting parts for manual quality control.
Periodic cycle	Periodic cycles are executed in a recurring (periodic) manner during the program execution. It is necessary for this type of cycle to specify the number of cycles of another cycle type after which the periodic cycle should be executed. The number of immediate repetitions is also specified. **Example**: Regular ejection of parts for manual quality control.

In MTPP, the cycles are mapped with the cycle types as shown below:

- Production - Continuous
- Action - Action
- StartUp - Counter
- RunOut - Counter

Continues on next page

For more details about cycledata refer *Application manual - RobotWare Machine Tending*.

4.4.2 Movements

4.4.2.1 Teach movements

Movements viewer

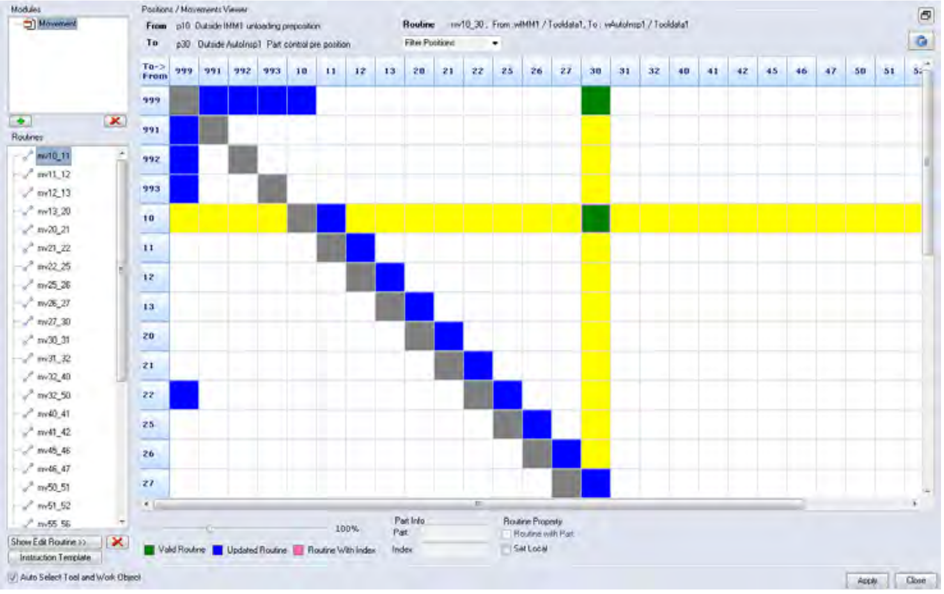
You can define movements between stations or within stations using the **Positions/Movement Viewer** window.

The movement between two points is defined as following:

```
PROC mv11_12()
!From: IMM Loading Position
!To : IMM Loading End Position
MT_MoveJ 11,p11,v50,z10,Tooldata1\WObj:=wGeneric1\NoMove;
MT_MoveLDO
     12,p12,v1500,z10,Tooldata1\WObj:=wGeneric1,sdoGeneric1_Outside,1;
ENDPROC
```

Where, `p11` and `p12` are the **From** and **To** positions. These procedures are defined in the `Movement.mod` module.

To configure a routine, select **Movements** > **Teach Movements**. The routines belonging to a selected movement module and the target positions in different stations and global positions are displayed in a grid as shown in the following image.



Positions / Movements Viewer

From p10 Outside IMM1 unloading preposition

To p30 Outside AutoInsp1 Part control pre position

Routine mv10_30 , From :wIMM1 / Tooldata1, To : wAutoInsp1 / Tooldata1

☑ Auto Select Tool and Work Object

xx1300000400

In RobotWare Machine Tending, positions for stations follow a convention. The positions `p10 – p19` are used to represent all the targets for StationA and `p20 – p29` are used for StationB. If more positions are required, they are defined as intermediate positions which lie between the main positions. For example, `p101101`, `p101102`, `p101103` are intermediate positions between `p10` and `p11`.

The corresponding procedure is illustrated in the following example:

```
PROC mv10_11()
```

Continues on next page

```
MT_MoveJ 10,p10,v50,z10,Tooldata1\WObj:=wIMM1\NoMove;
MT_MoveJ 101101,[[0,540,1200.015]…],v1000,z10,Tooldata1\WObj:=wIMM1;
MT_MoveJ 101102,[[0,360,1200.01]…],v1000,z10,Tooldata1\WObj:=wIMM1;
MT_MoveJ 101103,[[0,180,1200.005]…],v1000,z10,Tooldata1\WObj:=wIMM1;
MT_MoveL 11,p11,v500,fine,Tooldata1\WObj:=wIMM1;
ENDPROC
```

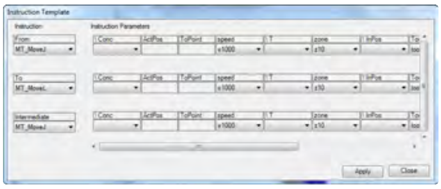
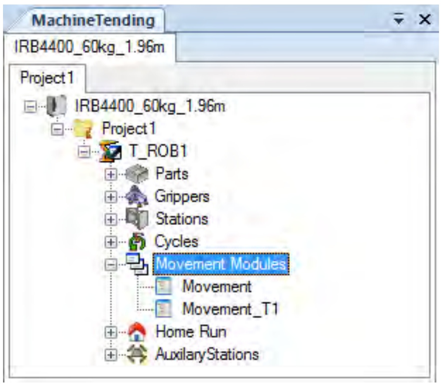
It is important to keep track of the order and sequence of these intermediate positions as they affect the **HomeRun** feature.

Add or delete movement routines

To configure a new movement routine:

	Action	Description
1	Click **Movements** and then click **Teach Movements**. The **Teach Movements** window is displayed.	
2	Select a movement module from the **Modules** list. The movement procedures belonging to the selected module are displayed.	The grid is populated with the positions defined for the selected module and global positions `p999` (**HomePos**), `p991- p993` (service positions) which are predefined when the PowerPac is activated. **Note** When you hover around the grid, the row and column in the grid is highlighted where the cursor is placed. The **From** and **To** field in the window is continuously updated with respect to the cursor position.
3	Double-click on a cell to select it.	The color of the cell changes from white to one of the following: • Green • Pink, if the **Routine with Part** option is selected, the part type number is suffixed to the routine (for example, mv11_12_T2). It is valid for the part type Movement modules. • Marked Local (L), if the **Set Local** option is selected. **Note** To delete existing routines, double-click on a routine. A cross mark ☒ is displayed on the selected routine. When you click the **Apply** button the selected routines are deleted.
4	Select the **Auto Select Tool and WorkObject**check box.	When this option is selected, the active Tool and the WObj of the RobTarget are assigned to the instruction. If this selection is cancelled, the values for `Tool` and `WObj` are referred from the **Instruction Template** as described in the previous step.

Continues on next page

	Action	Description
5	Click **Instruction Template**. The **Instruction Template** window is displayed.	 Instruction Template Instruction Instruction Parameters From MT_MoveJ To MT_MoveL Intermediate MT_MoveJ \Conc ActPos ToPoint speed v1000 \T zone z10 \InPos Apply Close xx1300000401
6	Select the instruction and parameters to be used for **From**, **To**, and **Intermediate** instructions.	
7	Click **Apply** and then click **Close**.	The selected routines are created and listed in the window and the cell color changes to blue or red depending on the **Routine with Part** option. **Note** When you double click on the configured or marked cell, if the routine is already created, that is, if the color of the cell is blue, then it is marked for deletion when you click **Apply**. If the cell is marked for creation, then it is deleted immediately.
8	Click **Apply**. All the movement modules belonging to the project are listed in the browser tree.	 MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Grippers Stations Cycles Movement Modules Movement Movement_T1 Home Run AuxilaryStations xx1300000503

Note

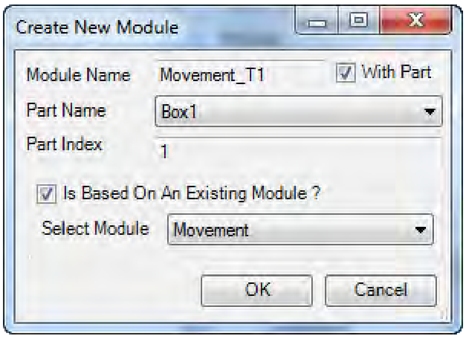
The station positions are usually declared as LOCAL. The routines mv991_999, mv992_999, mv993_999, mv999_991, mv999_992, and mv999_993 are created by default when the PowerPac is activated.

Continues on next page

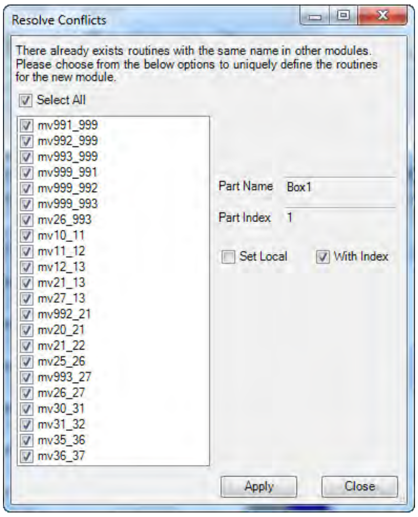
Add new movement module

When producing different parts, it is possible to use the same production cycle. In RobotWare Machine Tending, movement modules can be associated with a part as `Movement_T(X).mod`, where X is the part type index. Hence whenever a part is chosen for production, the corresponding movement module becomes active (loaded into the memory).

To create movement modules associated with part types:

	Action	Description
1	Click **Movements** and then click **Teach Movements**. The **Teach Movements** window is displayed.	
2	Click the + button under the **Modules** list. The **Create New Module** window is displayed.	 Create New Module Module Name Movement_T1 ☑ With Part Part Name Box1 Part Index 1 ☑ Is Based On An Existing Module ? Select Module Movement OK Cancel xx1300000403 The module name **Movement** is prefixed and is not editable. **Note** If the **Part Name** list is empty, then use the **Configure Parts** option to add parts.
3	Select the **With Part** check box. The **Part Name** and **Part Index** options are enabled. The parts added in the Project are listed in the **Part Name** drop down menu.	
4	Select a part from the **Part Name** list.	When a part is selected, the corresponding Index is appended to the Module Name (for example, **Movement_T1**).
5	Select the **Is Based On An Existing Module ?** check box and then select a Module from the **Select Module** drop-down list.	It is common for two movement modules to have the same routines and positions. In this option, existing movement modules are listed, user can choose to create a new module with similar routines. **Note** In this case, the targets are created by default and are declared as LOCAL to the module.

Continues on next page

	Action	Description
6	Click **OK**. The new module is listed in the **Modules** list.	 Resolve Conflicts There already exists routines with the same name in other modules. Please choose from the below options to uniquely define the routines for the new module. ☑ Select All ☑ mv991_999 ☑ mv992_999 ☑ mv993_999 ☑ mv999_991 ☑ mv999_992 ☑ mv999_993 ☑ mv26_993 ☑ mv10_11 ☑ mv11_12 ☑ mv12_13 ☑ mv21_13 ☑ mv27_13 ☑ mv992_21 ☑ mv20_21 ☑ mv21_22 ☑ mv25_26 ☑ mv993_27 ☑ mv26_27 ☑ mv30_31 ☑ mv31_32 ☑ mv35_36 ☑ mv36_37 Part Name Box1 Part Index 1 □ Set Local ☑ With Index Apply Close xx1300000404 There might be some naming conflicts in the new routines if they are created based on an existing module. In this case you have to resolve conflicts. The two options to resolve the conflicts to make the routines unique are: • **Set LOCAL** – Here the routine is set LOCAL to the new module. • **With Index** – If the module is based on a part, then the part index is appended to the routine. for example, `mv10_11_T1`. When the conflicts are resolved, the new module is listed in the **Modules** list.
7	Click **Apply**. The movement module is created.	The new routines are highlighted in the grid if they are not yet created. **Note** To delete a module, select the module in the **Modules** list and click ✕.

Note

When positions are edited using **Teach Position** window, click the refresh button ⟳ in the **Teach Movements** window to update the **Positions/ Movements Viewer** section.

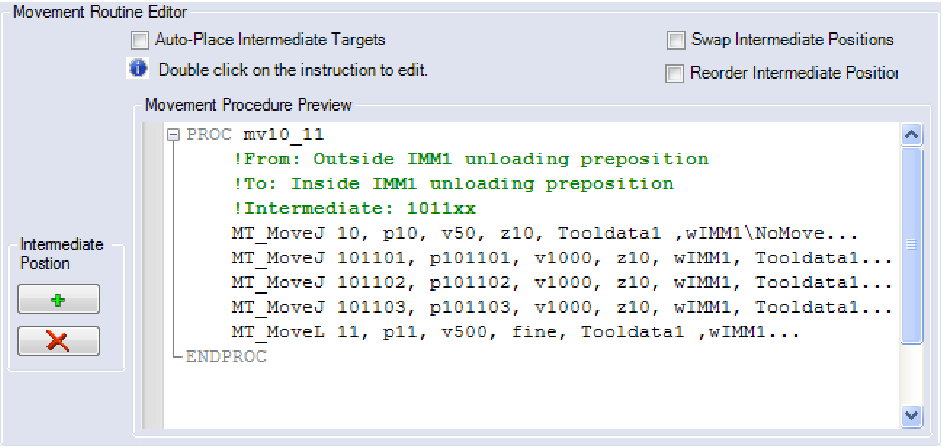
Edit routines

The **Show Edit Routine** option allows you to modify the movement routines by:

- Editing the routines to change the instruction parameter values.
- Adding or removing intermediate instructions to the routines.

Continues on next page

Click the **Show Edit Routine** >> button Show Edit Routine >> to display the **Movement Routine Editor** window.


Movement Routine Editor
□ Auto-Place Intermediate Targets
Double click on the instruction to edit.
□ Swap Intermediate Positions
□ Reorder Intermediate Positio
Movement Procedure Preview
Intermediate Postion
+
✕

```
PROC mv10_11
    !From: Outside IMM1 unloading preposition
    !To: Inside IMM1 unloading preposition
    !Intermediate: 1011xx
    MT_MoveJ 10, p10, v50, z10, Tooldata1 ,wIMM1\NoMove...
    MT_MoveJ 101101, p101101, v1000, z10, wIMM1, Tooldata1...
    MT_MoveJ 101102, p101102, v1000, z10, wIMM1, Tooldata1...
    MT_MoveJ 101103, p101103, v1000, z10, wIMM1, Tooldata1...
    MT_MoveL 11, p11, v500, fine, Tooldata1 ,wIMM1...
ENDPROC
```

xx1300000405

Note

The **Movement Routine Editor** is not a RAPID editor. You can modify an instruction by double clicking it which launches the **Modify Instruction** window.

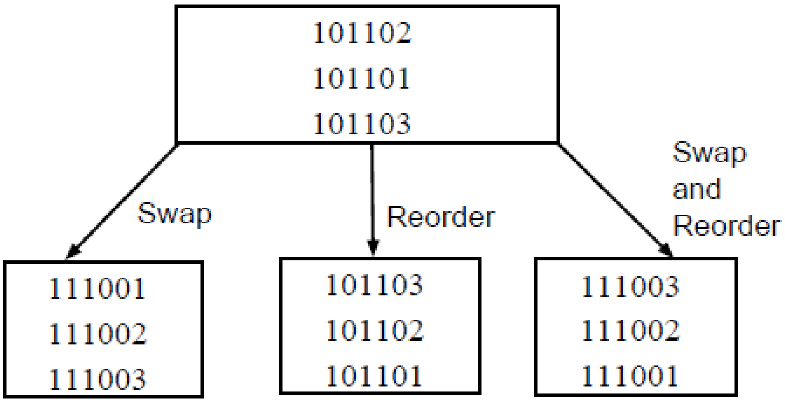
The following options in the **Movement Routine Editor** window allows you to manage the intermediate positions:

- Double-click an intermediate instruction displays the **Modify Instruction** window from where you can edit the selected instruction.
- The **Delete** button ✕ deletes a selected intermediate instruction.
- The **Auto-Place Intermediate Targets** check box places the new intermediate positions between the start and end instructions.

  Note

  The **Auto-Place Intermediate Targets** check box affects the positioning of the intermediate instructions on the path `mvxx_yy()`.
- The **Swap Intermediate Positions** check box swaps the start and end target position for a selected intermediate instruction.
- The **Reorder Intermediate Position** check box arranges the intermediate instructions in ascending or descending order.

Continues on next page



xx1300000564

To create an intermediate instruction:

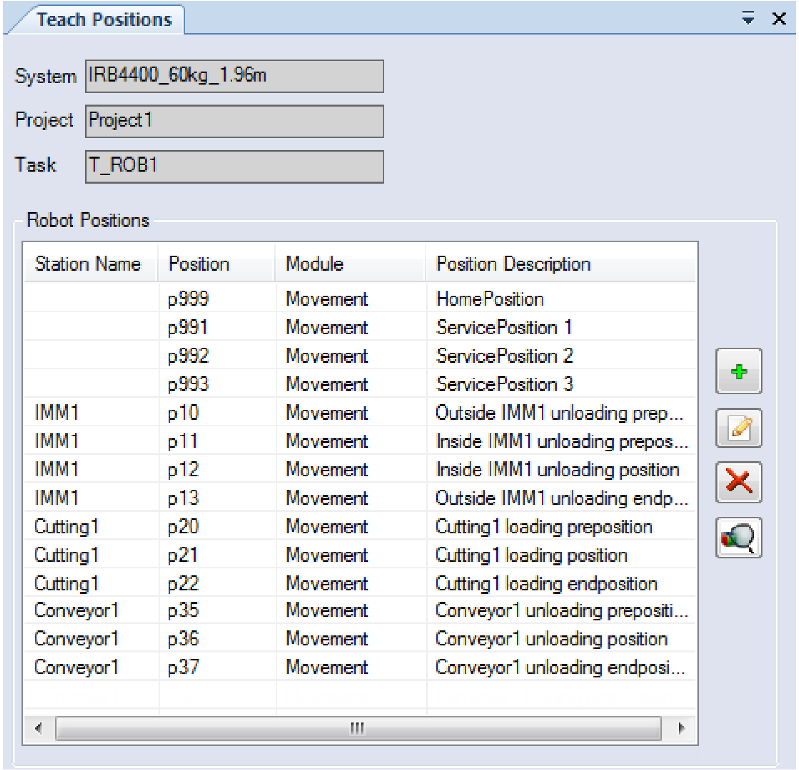
	Action
1	Click **Movements** and then click **Teach Movements**. The **Teach Movements** window is displayed.
2	Select a module from the **Module** list. The corresponding routines are listed in the **Routines** list.
3	Select a routine from the **Routines** list and click the **Show Edit Routine** >> button. The **Movement Routine Editor** window is displayed.
4	Click + in the **Intermediate Position** section. An intermediate instruction is inserted in the routine between the main instructions.
5	Click **Apply**. The changes to the routine are saved.

4.4.2.2 Teach positions

Introduction

The **Teach Positions** option is used for managing the robot positions.

To display robot positions, click **Movements** and then click **Teach Positions**. The **Teach Positions** window is displayed.



System IRB4400_60kg_1.96m

Project Project1

Task T_ROB1

Robot Positions

Station Name	Position	Module	Position Description
	p999	Movement	HomePosition
	p991	Movement	ServicePosition 1
	p992	Movement	ServicePosition 2
	p993	Movement	ServicePosition 3
IMM1	p10	Movement	Outside IMM1 unloading prep...
IMM1	p11	Movement	Inside IMM1 unloading prepos...
IMM1	p12	Movement	Inside IMM1 unloading position
IMM1	p13	Movement	Outside IMM1 unloading endp...
Cutting1	p20	Movement	Cutting1 loading preposition
Cutting1	p21	Movement	Cutting1 loading position
Cutting1	p22	Movement	Cutting1 loading endposition
Conveyor1	p35	Movement	Conveyor1 unloading prepositi...
Conveyor1	p36	Movement	Conveyor1 unloading position
Conveyor1	p37	Movement	Conveyor1 unloading endposi...

xx1300000415

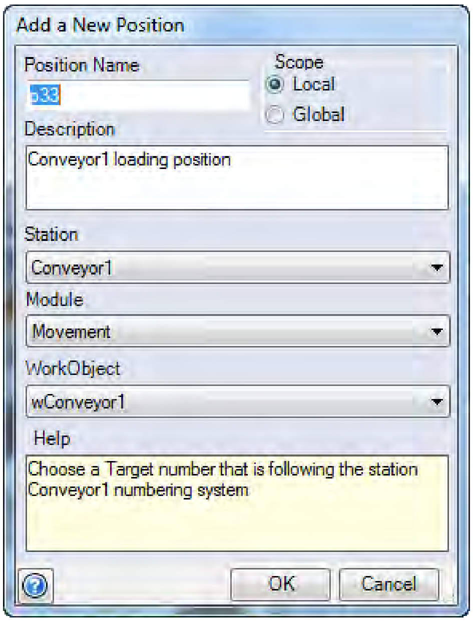
The **Teach Positions** window allows you to perform the following tasks:

- Create a new position
- Edit an existing position
- Delete a position
- Zoom in to a selected position

Continues on next page

Add a new position

To add a new position:

	Action	Description
1	Click +. The **Add a New Position** window is displayed.	 Add a New Position Position Name: p33 Scope: ● Local ○ Global Description: Conveyor1 loading position Station: Conveyor1 Module: Movement WorkObject: wConveyor1 Help: Choose a Target number that is following the station Conveyor1 numbering system xx1300000417 If you select an existing position and click +, the next available position number for the station is suggested. The position number follows the numbering convention of the station.
2	Type a name in the**Position Name** text box.	
3	Type a description in the **Description** text box.	
4	Select a station from the **Station** list.	
5	Select a module from the **Module** list.	
6	Select a work object from the **WorkObject** list.	

Continues on next page

	Action	Description

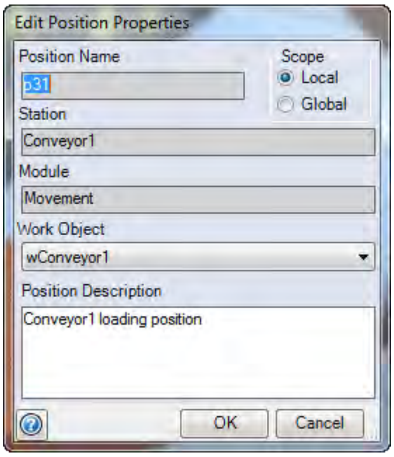
7	Click **OK**.
	The new position is added to the **Robot Positions** list.
		On synchronizing to virtual controller, the positions and their corresponding description are defined in the data type `posname`which is present in module `MT_Main`. Following is an example of the posname datatype.
		`!`
		`!***********************************`
		`! posname declarations`
		`!***********************************`
		`!Assigning position description for HomeRun`
		`CONST posname pnPositions{20}:=[`
		`[10,"Outside IMM1 unloading preposition"],`
		`[11,"Inside IMM1 unloading preposition"],`
		`[12,"Inside IMM1 unloading position"],`
		`[13,"Outside IMM1 unloading endposition"],`
		`[20,"Outside AutoInsp1 Part control pre position"],`
		`[21,"Inside AutoInsp1 Part control check position"],`
		`[22,"Outside AutoInsp1 Part control end position"],`
		`[30,"Conveyor1 loading preposition"],`
		`[31,"Conveyor1 loading position"],`
		`[32,"Conveyor1 loading endposition"],`
		`[991,"ServicePosition 1"],`
		`[992,"ServicePosition 2"],`
		`[993,"ServicePosition 3"],`
		`[999,"HomePosition"],`
		`[0,""],`
		`[0,""],`
		`[0,""],`
		`[0,""],`
		`[0,""],`
		`[0,""]`
		`];`

Continues on next page

Edit position

You can change the work object for a selected position.

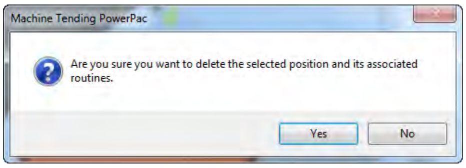
To edit an existing position:

	Action	Description
1	Select a position and click ✎. The **Edit Position Properties** window is displayed.	 Edit Position Properties Position Name: p31 Scope: ● Local ○ Global Station: Conveyor1 Module: Movement Work Object: wConveyor1 Position Description: Conveyor1 loading position xx1300000418
2	Change the work object in the **Work Object** list.	
3	Click **OK**. The changes are updated in the **Robot Positions** list.	

Delete position

You can delete a position from the Robot Positions list.

To delete a position:

	Action	Description
1	Select a position and click ✖. A confirmation window is displayed.	 Machine Tending PowerPac Are you sure you want to delete the selected position and its associated routines. xx1300000420
2	Click **Yes**. The selected position is deleted.	

Zoom in

You can zoom into a selected position.

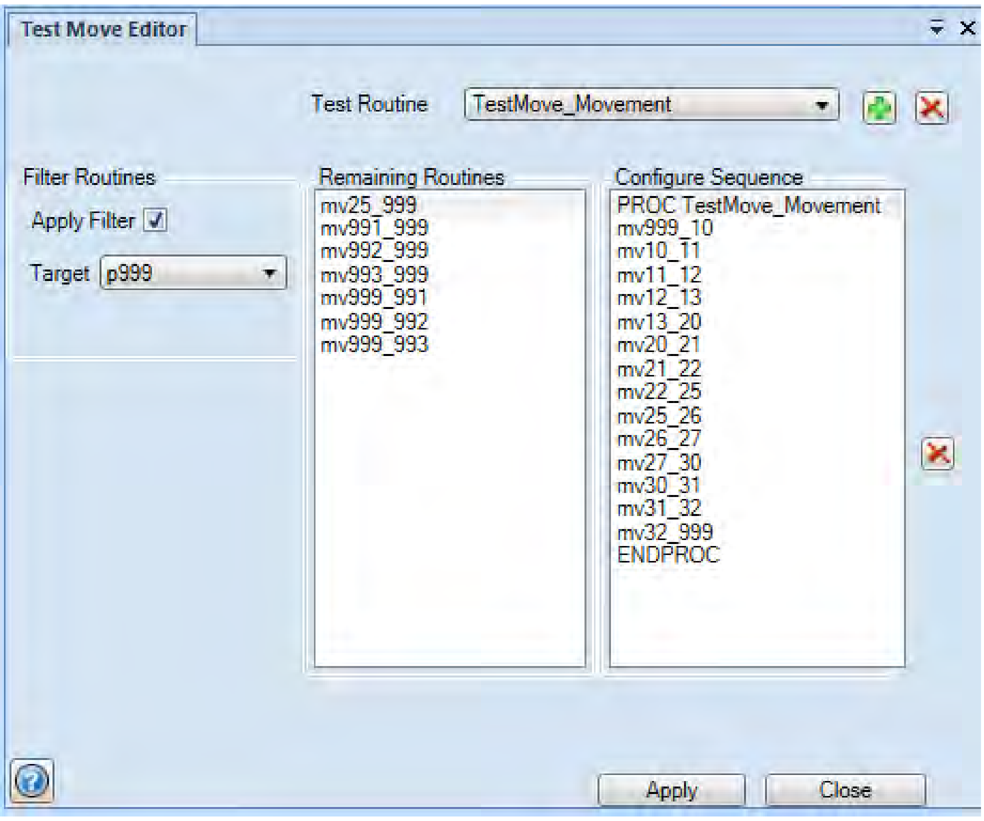
To zoom in to a selected position, select a position and click 🔍. The view is zoomed to the selected position.

4.4.2.3 Test Move Editor

Test Move Editor

Test Move Editor allows you to create, rearrange, and edit the movement routines.

To access Test Move Editor, select **Movements** > **Test Move Editor**. The following figure and table provide more information about **Test Move Editor** window.



Test Routine: TestMove_Movement

Filter Routines

Apply Filter ☑

Target: p999

Remaining Routines

mv25_999
mv991_999
mv992_999
mv993_999
mv999_991
mv999_992
mv999_993

Configure Sequence

PROC TestMove_Movement
mv999_10
mv10_11
mv11_12
mv12_13
mv13_20
mv20_21
mv21_22
mv22_25
mv25_26
mv26_27
mv27_30
mv30_31
mv31_32
mv32_999
ENDPROC

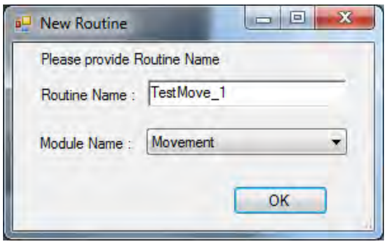
xx1600001658

Label	Description
Test Routine list	Displays all the available test move routines.
+ button	Creates a new test routine.
✕ button.	Deletes the selected test routine.
Configure Sequence list	List all the routines available in the selected test routine.
Remaining Routines list	Displays the routines which are available in the module but not used in the selected routine.
Filter Routines section	Click the **Apply Filter** check box and select a filter from the **Target** list to display the routines in the **Remaining Routines** list based on the selected target filter.
Apply button	Save the changes to the selected routine.
Close button	Closes the **Test Move Editor** window.

Continues on next page

Create a test routine

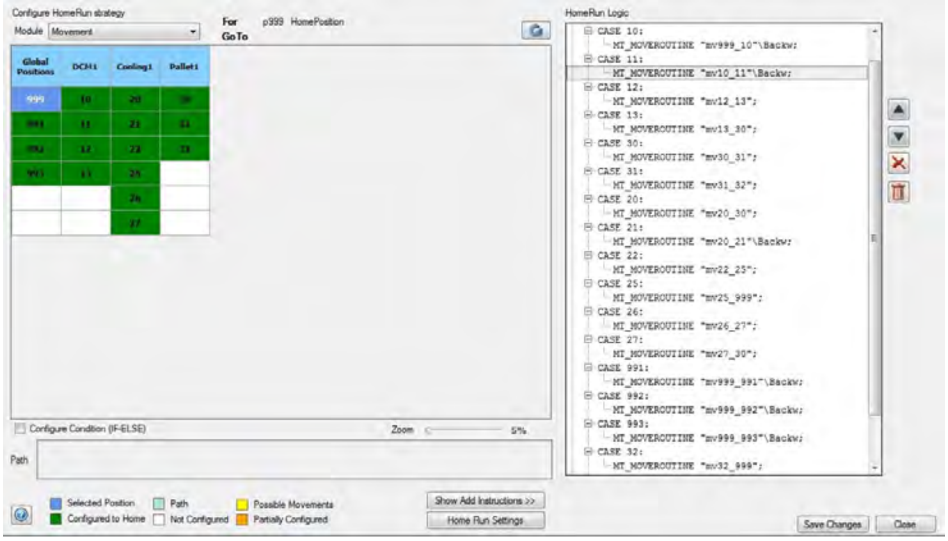
To create a test routine:

Step	Action	Description
1	Click the ➕ button The **New Routine** window is displayed.	 Please provide Routine Name Routine Name : TestMove_1 Module Name : Movement xx1600001659
2	In the **Routine Name** field type a name for the new routine.	**Note** The prefix `TextMove_` is not editable. You can type new characters after this prefix.
3	Select a module from the **Module Name** list.	
4	Click OK. The new routine is added to the **Test Routine** list.	
5	From the **Remaining Routine** list double-click the required routines and add it to the **Configure Sequence** list.	**Note** The **Filter Routines** section allows you to filter and display the required routines in the **Remaining Routine** list. **Note** Use the ✕ button to remove a routine from the **Configure Sequence** list.
6	Click **Apply**. The new routine is saved.	

4.4.3 HomeRun

Configure HomeRun

The **HomeRun** option in RobotWare Machine Tending ensures that the robot reaches the Home position from any position in the cell. The **HomeRun** can be requested by the operator from the FlexPendant or through other interfaces. The strategy for each position in the station can be configured from this user interface.



Global Positions	DCM1	Cooling1	Pallet1
999	10	20	30
991	11	21	31
992	12	22	32
993	13	25	
		26	
		27	

```
CASE 10:
  MT_MOVEROUTINE "mv999_10"\Backw;
CASE 11:
  MT_MOVEROUTINE "mv10_11"\Backw;
CASE 12:
  MT_MOVEROUTINE "mv12_13";
CASE 13:
  MT_MOVEROUTINE "mv13_30";
CASE 30:
  MT_MOVEROUTINE "mv30_31";
CASE 31:
  MT_MOVEROUTINE "mv31_32";
CASE 20:
  MT_MOVEROUTINE "mv20_30";
CASE 21:
  MT_MOVEROUTINE "mv20_21"\Backw;
CASE 22:
  MT_MOVEROUTINE "mv22_25";
CASE 25:
  MT_MOVEROUTINE "mv25_999";
CASE 26:
  MT_MOVEROUTINE "mv26_27";
CASE 27:
  MT_MOVEROUTINE "mv27_30";
CASE 991:
  MT_MOVEROUTINE "mv999_991"\Backw;
CASE 992:
  MT_MOVEROUTINE "mv999_992"\Backw;
CASE 993:
  MT_MOVEROUTINE "mv999_993"\Backw;
CASE 32:
  MT_MOVEROUTINE "mv32_999";
```

xx1300000433

From programming perspective:

- The HomeRun is always numbered as p999.
- The HomeRun is a RAPID module (`MT_Home_User.mod`).
- For each position, the HomeRun describes the action to be taken in `TEST-ENDTEST` format.
- The strategy is based on the status of the robot at the position and possible movements between From and To positions.

Example:

```
MODULE MT_HOME_USER
!***********************************************************
!*
!* Module name: MT_HOME_USER
!* User module for safe home run
!*
!***********************************************************


PROC MT_HomeRun(num Position)
  TEST Position
    CASE 11:
      MT_MOVEROUTINE "mv10_11"\Backw;
    CASE 10:
      MT_MOVEROUTINE "mv999_10"\Backw;
```

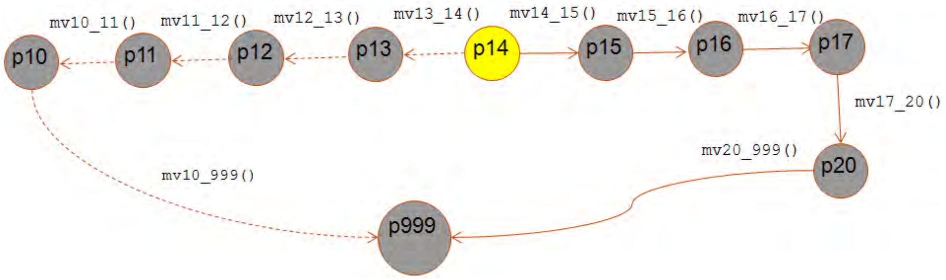
Continues on next page

```
        CASE 12:
          MT_MOVEROUTINE "mv11_12"\Backw;
        CASE 13:
          MT_MOVEROUTINE "mv12_13"\Backw;
        CASE 14:
          IF diPartinGripper THEN
            MT_MOVEROUTINE "mv14_15";
          ELSE
            MT_MOVEROUTINE "mv13_14"\Backw;
          ENDIF
        CASE 15:
          MT_MOVEROUTINE "mv15_16";
        CASE 16:
          MT_MOVEROUTINE "mv16_17";
        CASE 17:
          MT_MOVEROUTINE "mv17_20";
          MT_GripSet gsClose, gdGRP1_STG11;
        CASE 20:
          MT_MOVEROUTINE "mv20_999";
        DEFAULT;
          MT_ContHomeRun Position;
      ENDTEST
    ENDPROC
    ENDMODULE
```

If the HomeRun request is made when robot is at p14 then the path it takes is decided based on whether the part is gripped or not.

If part is not at the gripper, then the strategy is: 14 --> 13 --> 12 --> 11 --> 10 --> 999. That is, the robot moves back along the path. The routines mv13_14, mv12_13, mv11_12, mv10_11, mv10_999 are executed in the reverse order.

If part is in the gripper, then the strategy is: 14 --> 15 --> 16 --> 17 --> 20 --> 999. That is, the robot moves in the forward direction. The strategy is to release the part at p20 and then move to p999.

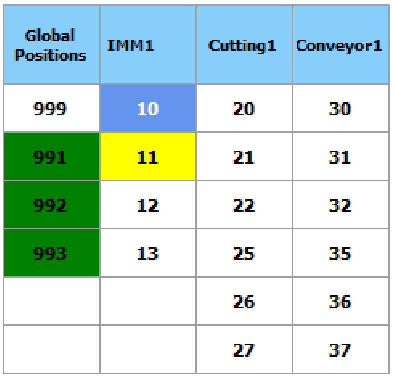
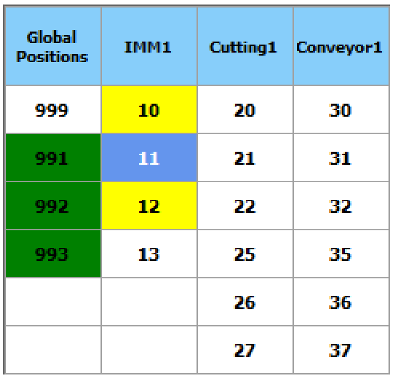


xx1300000565

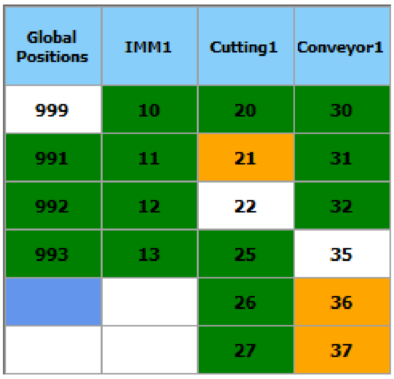
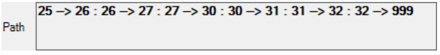
The HomeRun iteratively executes the strategy for each position until the robot reaches the home position.

Continues on next page

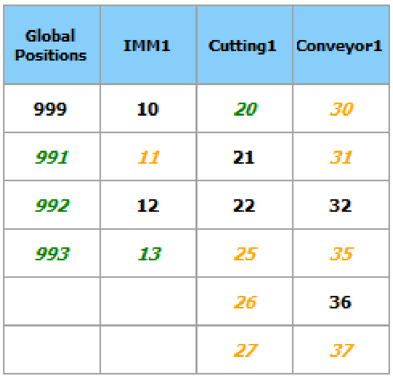
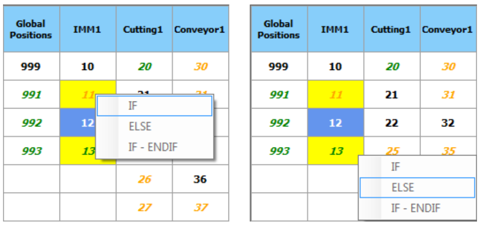
To configure RAPID for HomeRun:

<table>
<tr><th></th><th>Action</th><th>Description</th></tr>
<tr><td>1</td><td>Click <b>HomeRun</b>.
The <b>Home run strategy</b> window is displayed with the <b>Movement</b> module preselected in the <b>Module</b> list.</td><td>The <b>Home run strategy</b> window displays all the station positions belonging to the selected movement module in a grid together with the global positions p999 (HomePos), p991-p993 (Service positions). The strategy can be configured from this grid.</td></tr>
<tr><td>2</td><td>Select a position cell in the grid.
The selected cell becomes active and the color changes to blue. The strategy is to be configured for this selected position.</td><td>
<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td>21</td><td>31</td></tr>
<tr><td>992</td><td>12</td><td>22</td><td>32</td></tr>
<tr><td>993</td><td>13</td><td>25</td><td>35</td></tr>
<tr><td></td><td></td><td>26</td><td>36</td></tr>
<tr><td></td><td></td><td>27</td><td>37</td></tr>
</table>
xx1300000435
The position cells which have the movement routines configured either To or From the selected position are highlighted in yellow.</td></tr>
<tr><td>3</td><td>Select one of the highlighted cells to configure the strategy.
The strategy for the selected position in the previous step is configured. In the RAPID section, the <code>TEST-CASE</code> for the position is created with the movement instruction <code>MT_MOVEROUTINE</code>.</td><td>
<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td>21</td><td>31</td></tr>
<tr><td>992</td><td>12</td><td>22</td><td>32</td></tr>
<tr><td>993</td><td>13</td><td>25</td><td>35</td></tr>
<tr><td></td><td></td><td>26</td><td>36</td></tr>
<tr><td></td><td></td><td>27</td><td>37</td></tr>
</table>
xx1300000457
Subsequently the currently selected position becomes the next active position cell.
<b>Note</b>
Pressing the ESC key while configuring a cell clears the selection of the selected cell (blue color) and the highlighted (yellow color) cells, and displays the actual configured cells (green and orange).</td></tr>
</table>

Continues on next page

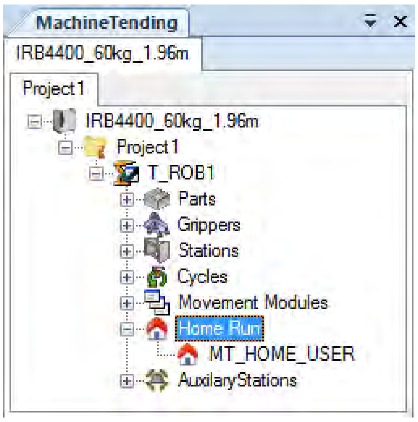
<table>
<tr><th></th><th>Action</th><th>Description</th></tr>
<tr><td>4</td><td>Repeat the preceding 2 steps for all the positions in the grid.
The color of the cells change when they are configured.</td><td>

<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td>21</td><td>31</td></tr>
<tr><td>992</td><td>12</td><td>22</td><td>32</td></tr>
<tr><td>993</td><td>13</td><td>25</td><td>35</td></tr>
<tr><td></td><td></td><td>26</td><td>36</td></tr>
<tr><td></td><td></td><td>27</td><td>37</td></tr>
</table>
xx1300000438
The cell color is either green or orange depending on whether the path reaches the Home position (p999). In a path, if the last movement is configured to reach Home position then the cell color of all the child paths are green, else they are orange.
For a position being configured, the cells of possible positions are highlighted in yellow. If a highlighted position is already configured, the position text is colored either green or yellow. Select any configured cell to view its path highlighted in light blue. Double-clicking a configured cell removes the position from the path.
When configuring the strategy for a position (for example, p10), then all the positions (for example, p11, p37, p36, and p35) in the chain are affected. This is indicated by the cell color change.
For a configured position, the complete path chain is displayed as shown in the following image.

Path 25 –> 26 : 26 –> 27 : 27 –> 30 : 30 –> 31 : 31 –> 32 : 32 –> 999
xx1300000436</td></tr>
</table>

Continues on next page

<table>
<tr><th></th><th>Action</th><th>Description</th></tr>
<tr><td>5</td><td>Select the <b>Configure Condition (IF-ELSE)</b> check box.
The text color is changed to the color of the cell and cell highlight is removed.</td><td>
<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td>21</td><td>31</td></tr>
<tr><td>992</td><td>12</td><td>22</td><td>32</td></tr>
<tr><td>993</td><td>13</td><td>25</td><td>35</td></tr>
<tr><td></td><td></td><td>26</td><td>36</td></tr>
<tr><td></td><td></td><td>27</td><td>37</td></tr>
</table>
xx1300000437
The path the robot takes during HomeRun depends on various conditions. For example, if part is in gripper then it has to be released at another position before moving to Home position. Else robot can move directly to the Home position. By selecting the <b>Configure Condition (IF-ELSE)</b> check box, these possibilities can be configured.</td></tr>
<tr><td>6</td><td>Select a position cell to configure the IF-ELSE strategy.
The position cells which have movement routines configured either To or From the selected position are highlighted in yellow.</td><td><b>Note</b>
If a path is configured using IF else condition and all the paths reaches Home position. If a path is not reaching Home position then it is displayed as orange or else in green.</td></tr>
<tr><td>7</td><td>Right-click and choose the IF condition and then select another highlighted cell, right-click, and choose either the ELSE or IF-ENDIF condition.</td><td>
<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td></td><td></td></tr>
<tr><td>992</td><td>12</td><td></td><td></td></tr>
<tr><td>993</td><td>13</td><td></td><td></td></tr>
<tr><td></td><td></td><td>26</td><td>36</td></tr>
<tr><td></td><td></td><td>27</td><td>37</td></tr>
</table>
<table>
<tr><th>Global Positions</th><th>IMM1</th><th>Cutting1</th><th>Conveyor1</th></tr>
<tr><td>999</td><td>10</td><td>20</td><td>30</td></tr>
<tr><td>991</td><td>11</td><td>21</td><td>31</td></tr>
<tr><td>992</td><td>12</td><td>22</td><td>32</td></tr>
<tr><td>993</td><td>13</td><td>25</td><td>35</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

xx1300000458
The position is configured and indicated by the text color change. The text color depends on the path reaching the Home Position similar to the earlier steps.
The RAPID section is configured as below:
<pre>
CASE 12:
IF <Exp> THEN
  MT_MOVEROUTINE
        "mv11_12"\Backw;
ELSE
  MT_MOVEROUTINE "mv12_13";
ENDIF
</pre>
<b>Note</b>
If you select a path having the IF-ELSE condition, then the <b>Path</b> field displays all the possible paths as separate rows.</td></tr>
</table>

Continues on next page

	Action	Description
8	Click **Show Add Instructions**. The **Add Instructions** list is displayed.	
9	Click **Apply**. The RAPID is saved under the HomeRun node in MTPP.	
10	Click **Save Changes**. The **Save As** window is displayed.	The HomeRun strategy can be saved as a module within the project folder. The name of the module is pre-configured as `MT_HOME_User.mod`.
11	Click **Save**. The HomeRun is configured and listed in the Browser tree and the RAPID is saved under the HomeRun node in MTPP.	 MachineTending IRB4400_60kg_1.96m Project1 IRB4400_60kg_1.96m Project1 T_ROB1 Parts Grippers Stations Cycles Movement Modules Home Run MT_HOME_USER AuxilaryStations xx1300000504

Note

In a project there can be several movement modules for different part types, but there is only one HomeRun module.

The grid and the RAPID are configured for the positions defined in the selected movement module. If the other movement module is selected and it does not contain the configured position or movement routine, then the corresponding CASE condition is removed and the grid is updated accordingly.

Ensure that the HomeRun strategy is valid for all the movement modules in the project.

Tip

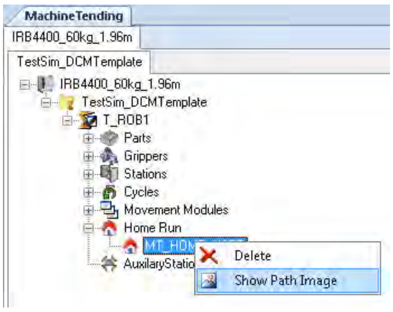
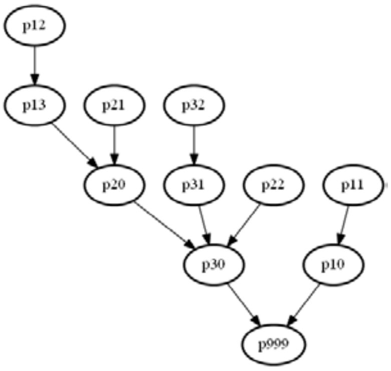
Select the movement module that contains the maximum number of positions and routines configured.

Continues on next page

Show path image

You can view a pictorial representation of all the paths covered with respect to various positions for the Home Run. home run, user can right click on the Home Run tree node in the browser and do so.

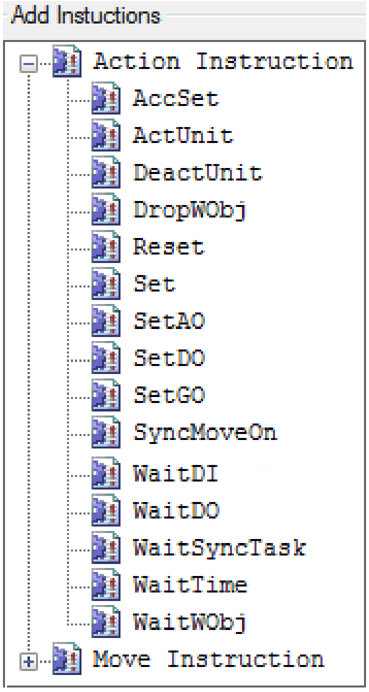
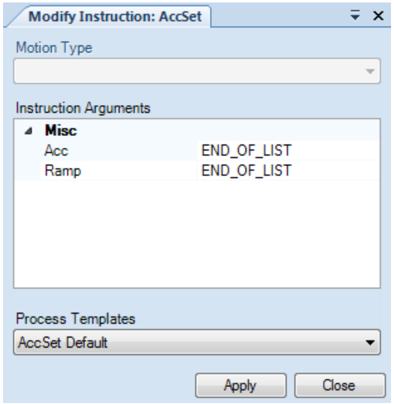
To view path image:

	Action	Description
1	Right-click on the HomeRun node (**MT_HOME_USER**) in the Browser tree.	 MachineTending IRB4400_60kg_1.96m TestSim_DCMTemplate IRB4400_60kg_1.96m TestSim_DCMTemplate T_ROB1 Parts Grippers Stations Cycles Movement Modules Home Run MT_HOM AuxilaryStatio Delete Show Path Image xx1300000563
2	Click **Show Path Image** The path image is displayed in the default image viewer.	 p12 p13 p21 p32 p20 p31 p22 p11 p30 p10 p999 xx1300000573

Continues on next page

Configuring instructions

To configure instructions:

Action	Description
Click **Show Add Instructions**. The **Add Instructions** list is displayed.	 Add Instuctions Action Instruction AccSet ActUnit DeactUnit DropWObj Reset Set SetAO SetDO SetGO SyncMoveOn WaitDI WaitDO WaitSyncTask WaitTime WaitWObj Move Instruction xx1300000459 The displayed instructions can be included to the **HomeRun Logic** section.
Select an instruction in the **Home Run Logic** section.	
Select an instruction in the **Add Instructions** list and click the arrow button. The **Modify Instruction** window is displayed.	 Modify Instruction: AccSet Motion Type Instruction Arguments Misc Apply Close Acc END_OF_LIST Ramp END_OF_LIST Process Templates: AccSet Default xx1300000461
Select the required instruction arguments and click the **Apply** button. The selected instruction is added below the selected instruction in the **HomeRun Logic** section.	

Continues on next page

Editing HomeRun Logic section

You can edit and rearrange instructions in the **HomeRun Logic** section.

- To edit an instruction that is added from the **Add Instruction** section, double-click the instruction. The **Modify Instruction** window is displayed from where you can modify the instructions.
- To rearrange and to delete the instructions, use the Up arrow ▲, Down arrow ▼, or Delete ✕ buttons.
- To clear all the configured RAPID and the grid, click the Clear button 🗑.

4.5 Validate

4.5.1 Path View

4.5.1.1 Overview

The **Path View** window displays the targets in a path by representing them with instruction icons. When a program contains several paths, the first instruction in the path view is the last instruction of the preceding path. The large arrow head indicates the start of a path. This enables you to verify that the robot can bridge the paths.

4.5.1.2 Instruction icons

Overview

An instruction icon shows the type (shape), motion (arrow), and status (color) of each target for a movement routine.

Instruction icons

The following table describes the icon shapes:

Icon	Description
○	Target

The following table describes the arrows:

Icon	Description
→	Linear move
⤳	Joint move

The following table describes the color coding. The goal is to turn all the status fields green before synchronizing the paths to the virtual controller.

Icon	Description
	White indicates an unknown status
🟩	Green indicates that the target is verified
🟨	Yellow indicates that solution is found, but not verified
🟥	Red indicates no solution or target is out of reach

The following table describes examples of the above:

Icon	Description
⤳🟢	Joint move, target verified
→○	Linear move, unknown status

4.5.1.3 The Path View tool bar

Overview

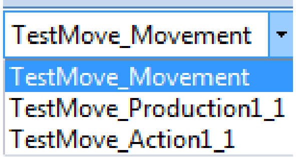
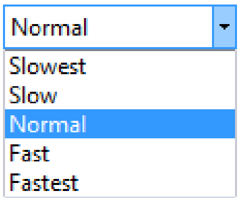
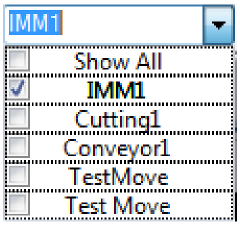
The **PathView** tool bar has buttons for the most frequently used commands.

Path View tool bar

Icon	Description
	The **Set View Center** button sets the center view automatically to the selected target in the path view or the active TCP (if no target is selected).
	The **Mode Selection** button enables the editing of the instruction.
	The **View Tool at Target** button displays the tool configuration at the target.
	The **Check Reachability** button tries to move the robot to the selected target to check whether the target is reachable.
	The **Arm** button displays the **Robot Configuration** dialog box. The **Arm** button indicates the active arm configuration of the target.
	The **Wrist** button displays the **Robot Configuration** dialog box. The**Wrist** button indicates the active wrist configuration of the target.
	The **Tool** button displays the **Robot Configuration** dialog box. The**Tool** button indicates the active tool configuration of the target.
	The **Jump to Target** button jumps the robot with active TCP to the selected target. It gives you a chance to view torch angles and detect the possible collisions with the robot. • A successful result turns the target yellow and moves the robot one discrete step towards the target. • An unsuccessful result turns the target red and leaves the robot in its current position.
	The **Move to Target** button moves the robot to the selected target from the previous target in the target list, checks for reach and sets the robot configuration. You can also select a range of targets and move the robot in sequential order down the target list. • A successful result turns the target green and moves the robot continuously towards the target. • An unsuccessful result turns the target red and leaves the robot in its current position. **Note** The virtual controller is not running while executing this command.
	The **Simulate** button synchronizes the opened paths to the virtual controller and executes the paths in the virtual controller.

Continues on next page

4.5.1.3 The Path View tool bar
Continued

Icon	Description
■	The **Stop** button stops the current execution.
 TestMove_Movement TestMove_Movement TestMove_Production1_1 TestMove_Action1_1 xx1600001660	Displays the available test move routines. The path view graphics display is updated based on the selected test move routine.
	The **Speed Control** list applies only to **Jump to** and **Move to** target. Simulation speed is determined by the RAPID program.
 IMM1 Show All IMM1 Cutting1 Conveyor1 TestMove Test Move	This list allows you to select the routines associated with specific station. The Test Move option in this list to view the path configured using test move.

4.5.1.4 The configuration menu

Check reach

This command checks whether the robot can reach a target. A successful result turns the target yellow, while an unsuccessful result turns it red.

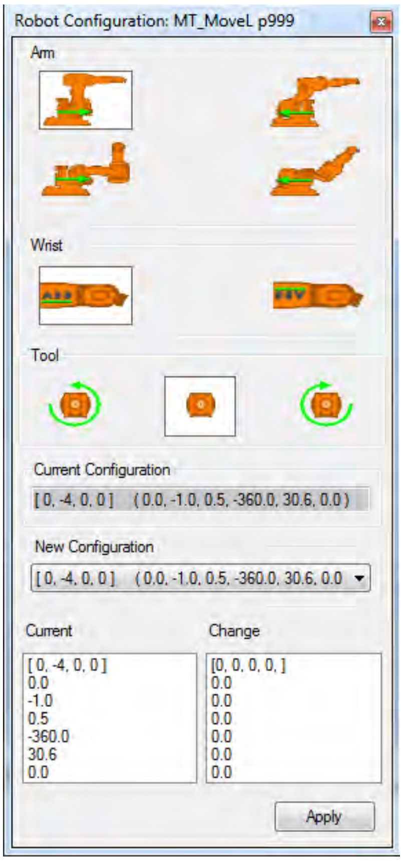
Set configuration

Overview

The same target position and orientation can be attained in different ways, using different sets of axis angles. This settings is known as robot configurations. In Machine Tending PowerPac, robot configuration can be set for individual targets. The system calculates a configuration based on the selections, after which the values are displayed.

The Robot configuration window

The **Robot Configuration** window requires that you select three positions, one each for the robot arm, wrist, and tool, before clicking **Apply**.



Robot Configuration: MT_MoveL p999

Current Configuration: [0, -4, 0, 0] (0.0, -1.0, 0.5, -360.0, 30.6, 0.0)

New Configuration: [0, -4, 0, 0] (0.0, -1.0, 0.5, -360.0, 30.6, 0.0

Current	Change
[0, -4, 0, 0]	[0, 0, 0, 0,]
0.0	0.0
-1.0	0.0
0.5	0.0
-360.0	0.0
30.6	0.0
0.0	0.0

xx1300000408

The following table describes the elements of the **Set Configuration** window:

Object	Description
Arm	Specifies whether the robot wrist is in front of or behind axis 1, and whether the elbow is up or down.

Continues on next page

Object	Description
Wrist	Specifies whether the axis 4 can be turned positive or negative.
Tool	Specifies whether axis 6 can be turned positive or negative. If neutral is selected, the configuration of axis 6 is determined by the arm and wrist selections.
New	Displays the configuration and joint values for a selected configuration, as well as alternatives.
Current	Displays the configuration currently stored in the selected target.
Change	Displays the difference in joint values between the currently selected target and the preceding target in the path.

Tip

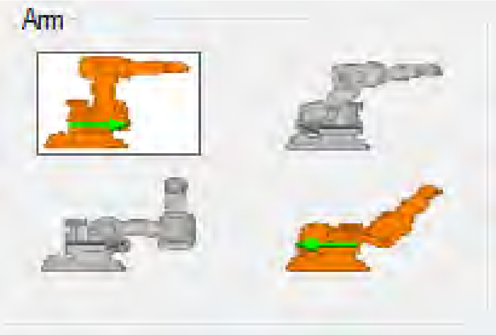
If the **Robot Configuration** window and the **Modify Target** dialog box are open at the same time the configuration is automatically recalculated when the target is modified, which helps in determining how much you can adjust the target while keeping it within reach.

Setting Robot configuration

The Machine Tending PowerPac provides two ways to set Robot configuration for a target.

- Specify a configuration setting by selecting a combination of arm, wrist, and tool configuration among the symbols on the control. The settings is used to find the most suitable robot configuration.
- Manually select a robot configuration among all the possibilities given. This is done by selecting one of the robot configurations in the drop-down **New**. The configuration setting controls is then be updated accordingly.

When selecting a configuration, the selected symbol gets a frame around it, and the selection process makes the background white for the valid configuration that is selected. If the two (the frame and the white background) does not coincide, the selection could not be fulfilled.



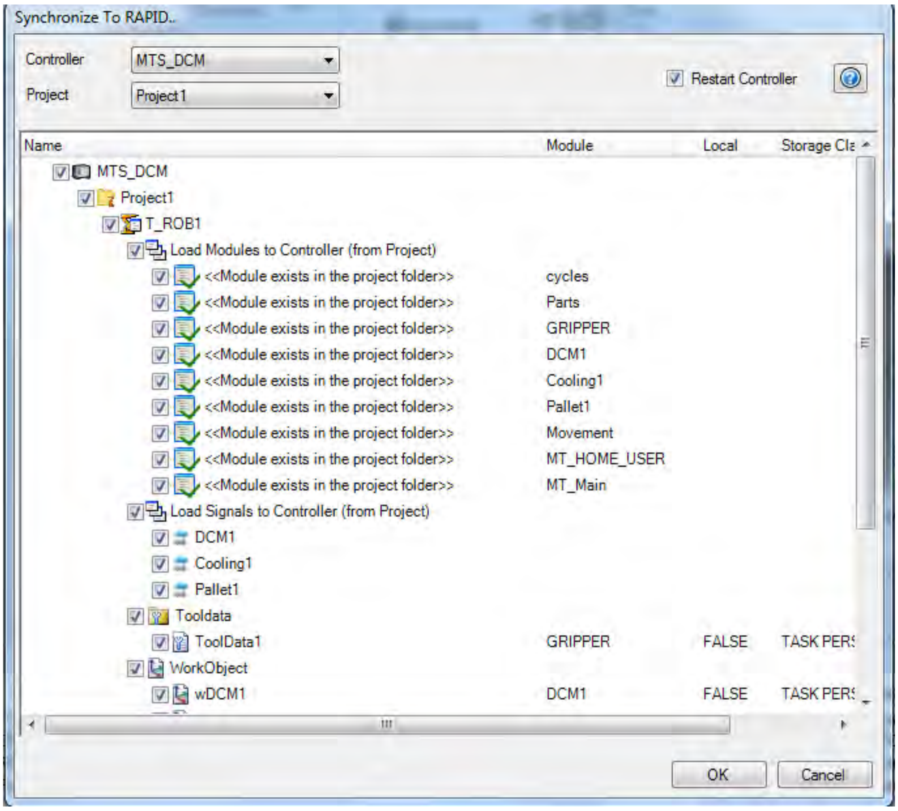
xx1300000410

4.6 Controller options

4.6.1 Synchronize to RAPID

Introduction

The **Synchronize to RAPID** option enables the creation of RAPID modules on the Virtual Controller (VC) for a selected project.



Synchronize To RAPID..

Controller: MTS_DCM
Project: Project1
☑ Restart Controller

Name	Module	Local	Storage Cl
☑ MTS_DCM			
☑ Project1			
☑ T_ROB1			
☑ Load Modules to Controller (from Project)			
☑ <<Module exists in the project folder>>	cycles		
☑ <<Module exists in the project folder>>	Parts		
☑ <<Module exists in the project folder>>	GRIPPER		
☑ <<Module exists in the project folder>>	DCM1		
☑ <<Module exists in the project folder>>	Cooling1		
☑ <<Module exists in the project folder>>	Pallet1		
☑ <<Module exists in the project folder>>	Movement		
☑ <<Module exists in the project folder>>	MT_HOME_USER		
☑ <<Module exists in the project folder>>	MT_Main		
☑ Load Signals to Controller (from Project)			
☑ DCM1			
☑ Cooling1			
☑ Pallet1			
☑ Tooldata			
☑ ToolData1	GRIPPER	FALSE	TASK PER:
☑ WorkObject			
☑ wDCM1	DCM1	FALSE	TASK PER:

xx1300000473

The **Synchronize to RAPID** option offers the following possibilities:

- Load project modules to VC.
- Synchronize the following data types apart from the regular `WObj` and `ToolData`.
  - `GrpData`
  - `PartData`
  - `CycleData`
  - `StationData`
  - `ProjInfo`
  - `posname`
- Synchronize the target and the routines defined in the RobotStudio.
- Load signals to the virtual controller for the station signals and gripper signals.

Continues on next page

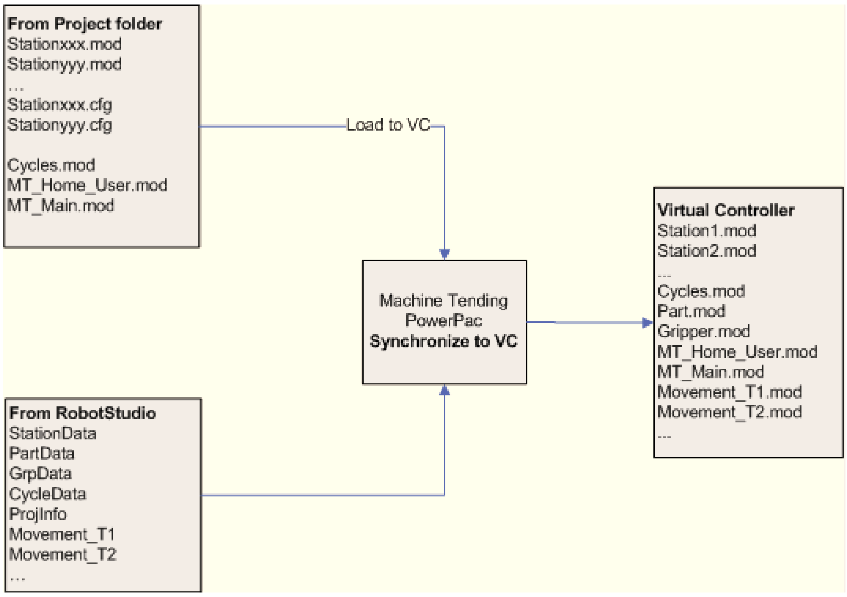
To create the project related information on the virtual controller.

Action	Description
Define tool, stations, parts, cycles, HomeRun, movement in the Project.	All the items are created and listed in the browser tree.
Click **Synchronize to RAPID**. The **Synchronize to RAPID** window is displayed.	
Select the required items for synchronizing including options to load the station signals and restart the controller.	The project properties lists all the modules that are part of the Project. • All the module name entries belonging to Project are listed. These modules are loaded to controller from the file system. If there is no corresponding module in the file system for an entry, then it is highlighted as **Module does not exist**. • All the data types and routines created from MTPP.
Click **OK**.	The selected project modules are loaded to controller from the file system. The selected items are synchronized to the Virtual controller.

Note

The message **Module does not exist in Project folder** is not an error. It is indicating that the file has not yet been saved. Synchronize to VC and select **Save Project** option from the MTPP ribbon. This saves the modules in the project from the VC into the folder.

The process of different modules (also signals) getting included into the virtual controller is illustrated in the following diagram.



xx1300000454

Continues on next page

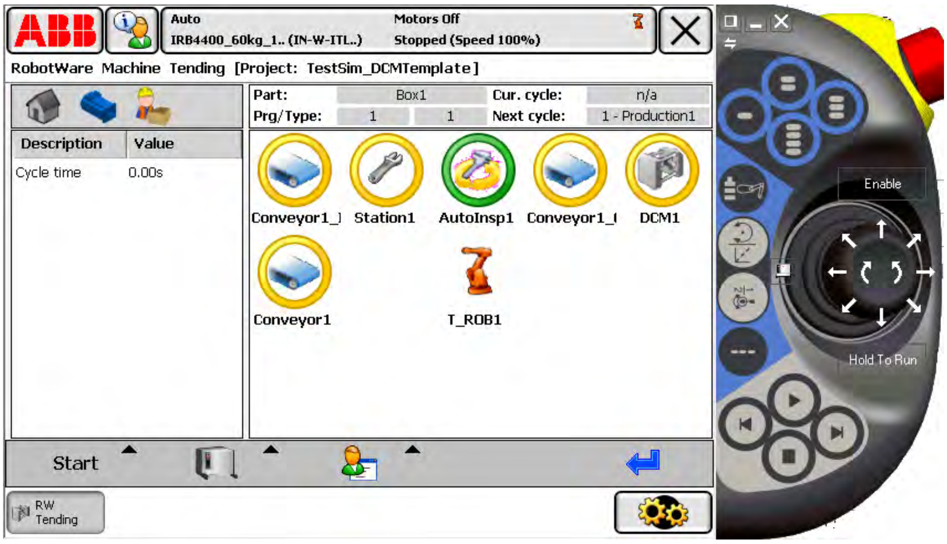
After synchronization, all the modules are included in the VC. The sequence of synchronization is in the same order as in the dialog, that is, first modules are loaded to controller followed by synchronizing the data types and then the routines. For the selected station modules which are loaded to the controller, the corresponding station signals are also added to the controller. The signals are included within the corresponding station units. When data type `grpdata` is selected for synchronization, the corresponding gripper signals are added to the controller. The controller has to be restarted when there are updates to the configuration. The **Restart** option restarts the controller after synchronization.

The status modules associated with the project indicates whether the module exists in the project folder to load to the controller.

4.6.2 Virtual FlexPendant

Introduction

The **Virtual FlexPendant** option allows you to open the virtual FlexPendant. On opening the FlexPendant the RobotWare Machine Tending application can be viewed.



Auto
IRB4400_60kg_1.. (IN-W-ITL..)
Motors Off
Stopped (Speed 100%)

RobotWare Machine Tending [Project: TestSim_DCMTemplate]

Part:	Box1	Cur. cycle:	n/a
Prg/Type:	1 1	Next cycle:	1 - Production1

Description	Value
Cycle time	0.00s

xx1300000551

You can open the Virtual FlexPendant in one of the following ways:

- In the **Controller Tools** group, click **Virtual FlexPendant**.
- Press the keyboard shortcut, CTRL + F5.

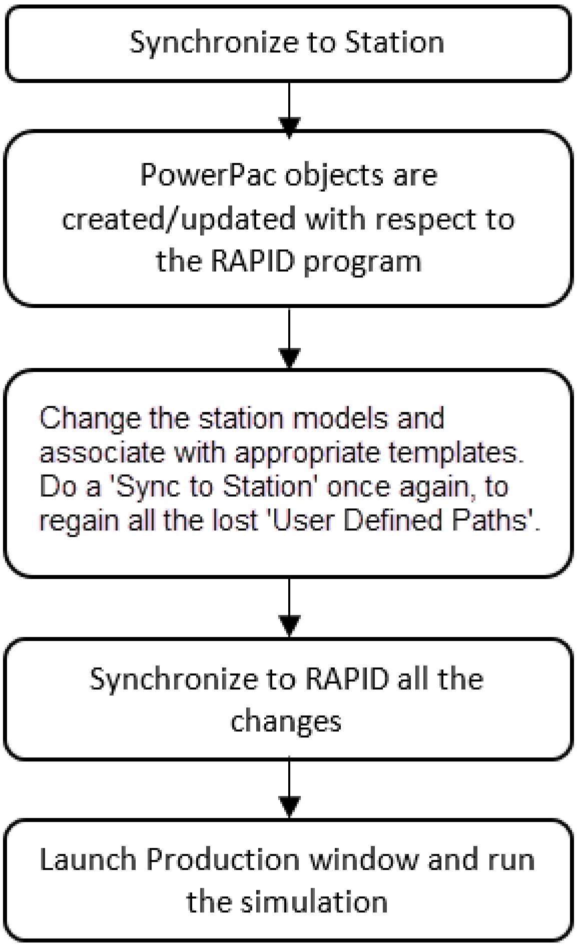
Note

The Virtual FlexPendant is applicable while running a virtual controller.

4.6.3 Synchronize to Station

Introduction

The **Synchronize to Station** option generates the powerpac objects in MTPP based on the configured RAPID program. With the help of this feature, you can generate most of the cell layout in virtual environment of RobotStudio for modeling, offline programming, and verification of the complete simulation of the machine tending station with minimal configuration efforts.


Synchronize to Station
PowerPac objects are created/updated with respect to the RAPID program
Change the station models and associate with appropriate templates. Do a 'Sync to Station' once again, to regain all the lost 'User Defined Paths'.
Synchronize to RAPID all the changes
Launch Production window and run the simulation

xx1400002485

The Synchronize to station feature works in any one of the following situations:

- There is an existing RobotStudio station (Pack and Go) with the Machine Tending project configured and synchronized to VC.
- There is a new RobotStudio station with a VC containing Machine Tending modules and synchronize to station operation is performed to show the content in MTPP.

Synchronize to station of MTPP synchronizes the following powerpac objects apart from the regular WorkObject and ToolData.

- `GrpData`
- `PartData`

Continues on next page

- `CycleData`
- `StationData`
- `ProjInfo`
- `Posname`
- Targets and the movement routines
- `HomeRun logic`
- `Cycle logic`

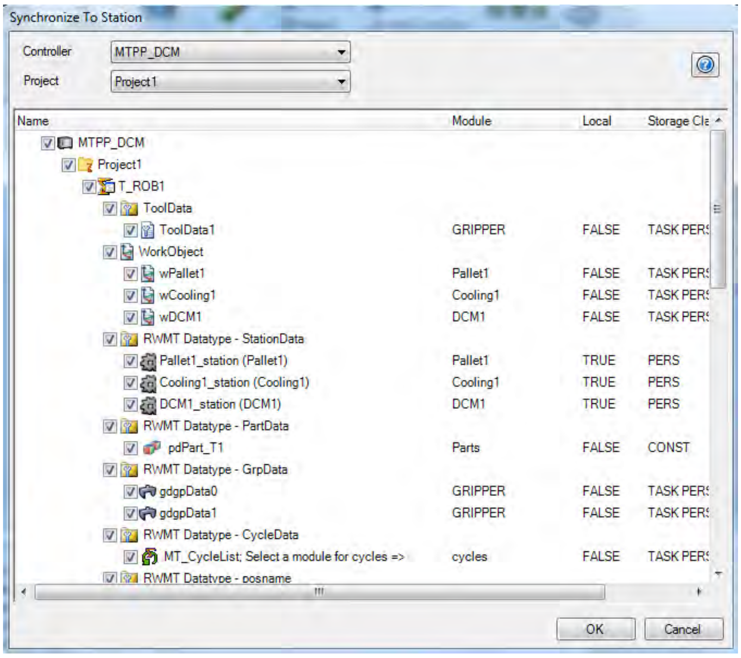
The following conditions must be satisfied before performing the synchronize to station:

- RAPID program contains Machine Tending modules – That is, RAPID programming must be in adherence to the programs created using the PowerPac.
- A tool must be attached to the robot.

To Synchronize only the station objects:

1 Click **Synchronize** > **Synchronize to Station**.

  The **Synchronize to Station** window is displayed.



Controller: MTPP_DCM

Project: Project1

Name	Module	Local	Storage Cla
☑ MTPP_DCM			
☑ Project1			
☑ T_ROB1			
☑ ToolData			
☑ ToolData1	GRIPPER	FALSE	TASK PERS
☑ WorkObject			
☑ wPallet1	Pallet1	FALSE	TASK PERS
☑ wCooling1	Cooling1	FALSE	TASK PERS
☑ wDCM1	DCM1	FALSE	TASK PERS
☑ RWMT Datatype - StationData			
☑ Pallet1_station (Pallet1)	Pallet1	TRUE	PERS
☑ Cooling1_station (Cooling1)	Cooling1	TRUE	PERS
☑ DCM1_station (DCM1)	DCM1	TRUE	PERS
☑ RWMT Datatype - PartData			
☑ pdPart_T1	Parts	FALSE	CONST
☑ RWMT Datatype - GrpData			
☑ gdgpData0	GRIPPER	FALSE	TASK PERS
☑ gdgpData1	GRIPPER	FALSE	TASK PERS
☑ RWMT Datatype - CycleData			
☑ MT_CycleList; Select a module for cycles =>	cycles	FALSE	TASK PERS
☑ RWMT Datatype - posname			

xx1400002483

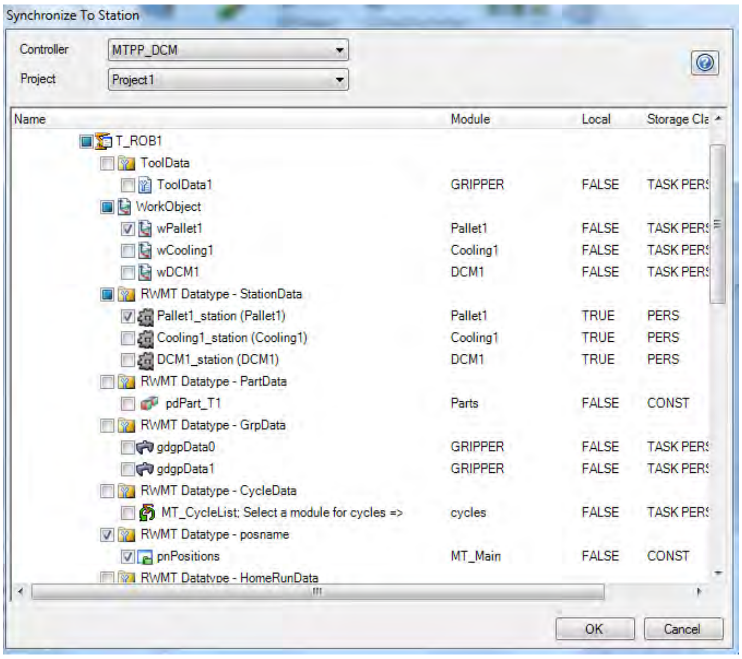
2 Clear all the default selections and select only the following:

  - WorkObject
  - RWMT Datatype – StationData
  - RWMT Datatype – posname
  - Paths & Targets

Continues on next page

Note

If a single station object is to be synchronized then select the data types for that particular station.



Synchronize To Station

Controller: MTPP_DCM

Project: Project1

Name	Module	Local	Storage Cla
■ T_ROB1			
□ ToolData			
□ ToolData1	GRIPPER	FALSE	TASK PERS
■ WorkObject			
☑ wPallet1	Pallet1	FALSE	TASK PERS
□ wCooling1	Cooling1	FALSE	TASK PERS
□ wDCM1	DCM1	FALSE	TASK PERS
■ RWMT Datatype - StationData			
☑ Pallet1_station (Pallet1)	Pallet1	TRUE	PERS
□ Cooling1_station (Cooling1)	Cooling1	TRUE	PERS
□ DCM1_station (DCM1)	DCM1	TRUE	PERS
□ RWMT Datatype - PartData			
□ pdPart_T1	Parts	FALSE	CONST
□ RWMT Datatype - GrpData			
□ gdgpData0	GRIPPER	FALSE	TASK PERS
□ gdgpData1	GRIPPER	FALSE	TASK PERS
□ RWMT Datatype - CycleData			
□ MT_CycleList; Select a module for cycles =>	cycles	FALSE	TASK PERS
☑ RWMT Datatype - posname			
☑ pnPositions	MT_Main	FALSE	CONST
□ RWMT Datatype - HomeRunData			

xx1400002484

3 Click **OK**.

The station is created with the following properties:

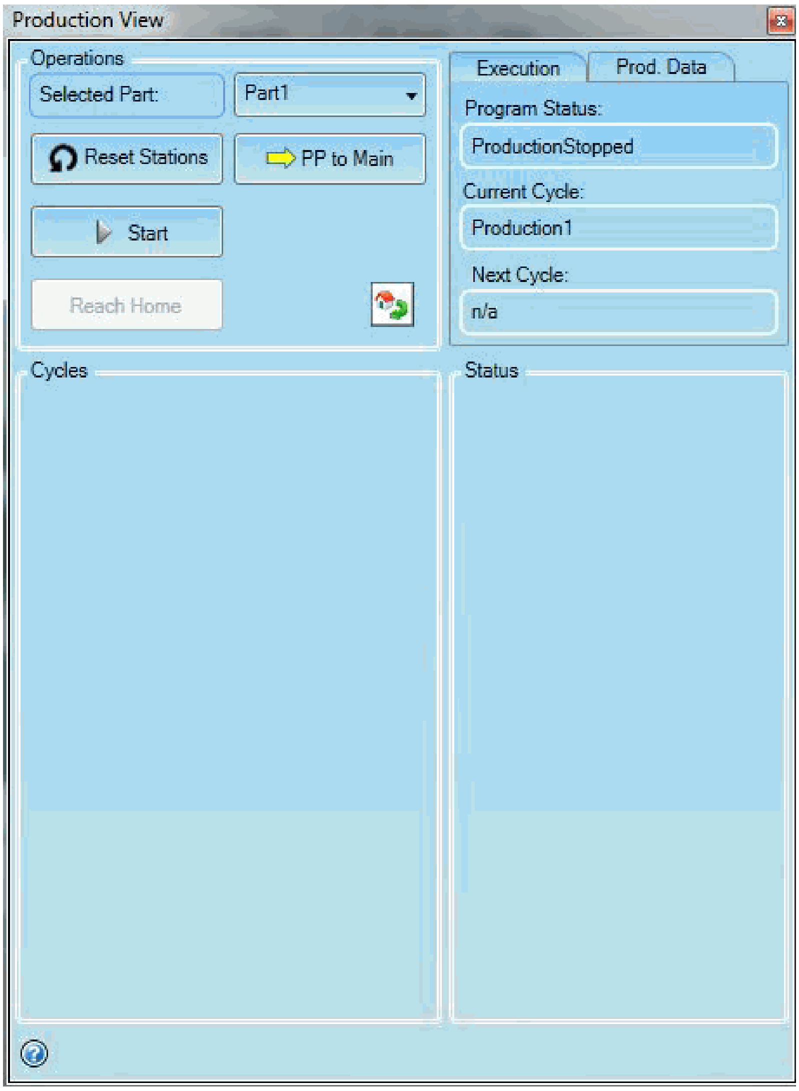
- Station model by default is Generic table.
- Station is associated with “No Template”
- Station is placed according to the workobject.
- Station number is with respect to the target numbering of the station.

4.7 Simulation

4.7.1 Production view

Introduction

The **Production View** feature enables simulation of cycles in RobotStudio.



Selected Part: Part1

Program Status: ProductionStopped

Current Cycle: Production1

Next Cycle: n/a

xx1300000525

Prerequisites

Following are the prerequisites for running the simulation:

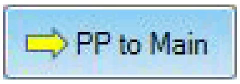
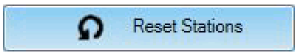
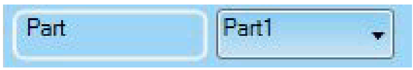
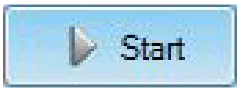
- The Project is configured completely with information about the part, stations, cycles, and so on.
- It is synchronized to VC and is error free.
- All the required movement routines for different cycles and HomeRun are configured and validated using Path view.

Continues on next page

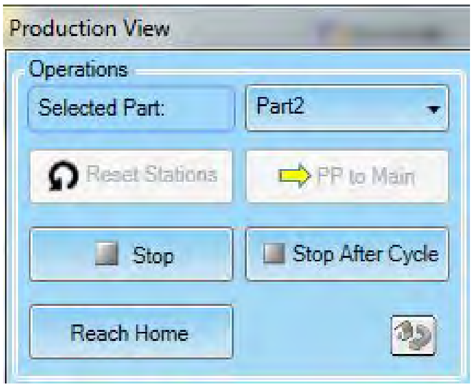
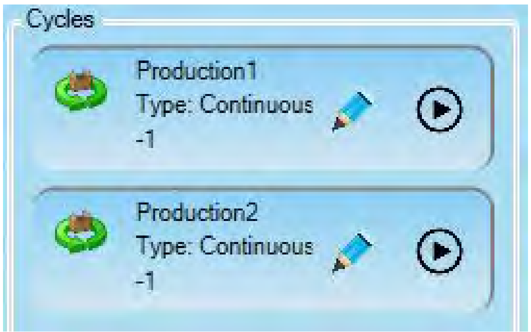
- The PartData is configured to run with cycles and the name of the processing routine is defined.
- The processing routine is defined to call the corresponding cycle routine.
- The corresponding **cycledata** is updated in the `MT_Main( )` module.
- Ensure that the entry position in simulation setup is set to `main( )` routine.

Simulation procedure

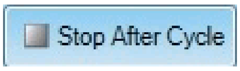
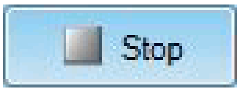
The following procedure describes the process of simulating the cycles:

	Action	Description
1	Select **Production** from the MTPP ribbon. The **Production View** window is displayed.	
2	Click **PP to Main**. The program pointer is set to `MT_Execute( )` in the `MT_Main( )` module.	 xx1300000526
3	Click **Reset Stations**. Resets all the station smart components to their initial state.	 xx1300000527 Each Station SmartComponent (SC) has an initial state. For example, for the Injection Moulding Machine, the initial state of the machine is open with the signals for core pullers, ejectors being set to the respective values. **Note** This is required to be done only once during a simulation. But if there are some situations where the station smart component is not behaving as expected they can be reset again.
4	Select a part from the **Part** drop-down list. The part is selected.	 xx1300000524 This is the first step while executing the cycles. For the selected part, the corresponding cycles are shown in the Cycles List based on the part processing routine. Click **Deselect** to clear the selection
5	Click **Start**. The robot automatically moves to the home position.	 xx1300000528 For selecting any cycle for production, the robot must be in the home position. As the robot moves to the home position, the **Start** button is enabled again. **Note** It might be required to select the **Start** and **Reach Home** more than once to move the Robot. This behavior is the same as on the RobotWare Machine Tending user interface on the FlexPendant where there is a confirmation for every operation.

Continues on next page

	Action	Description
6	Click **Start** again. The simulation is started and a list of available cycles is displayed.	Once you start a cycle the options to stop and stop after cycle are available as shown in the following figure.  Production View Operations Selected Part: Part2 Reset Stations PP to Main Stop Stop After Cycle Reach Home xx1500002115
7	Select a cycle from the **Cycles** list and click on the play button. The cycle is executed together with the station components as defined in the cycledata. That is, either continuous, counter, periodic, and so on.	 Cycles Production1 Type: Continuous -1 Production2 Type: Continuous -1 xx1300000570 When the selected cycle is executing, the next possible cycles are listed for selection. This behavior is same as that on the RobotWare Machine Tending FlexPendant user interface. For example, when a continuous cycle is executing, the action cycles are listed and vice-versa. If the robot is stopped, the list of cycles is not displayed. **Note** If the selected cycle does not execute or the cycle is blinking, then there might be missing or incorrect information configured in the RAPID modules. Verify the RAPID for the following common issues. • Whether the part data and cycle routines are properly mapped. • Whether the cycle data is configured with the correct cycle information. For more details on starting individual station smart components, see *Starting simulation for station smart components on page 116*.
8	While simulation is executing click **Reach Home** The simulation stops.	 Reach Home xx1500002110

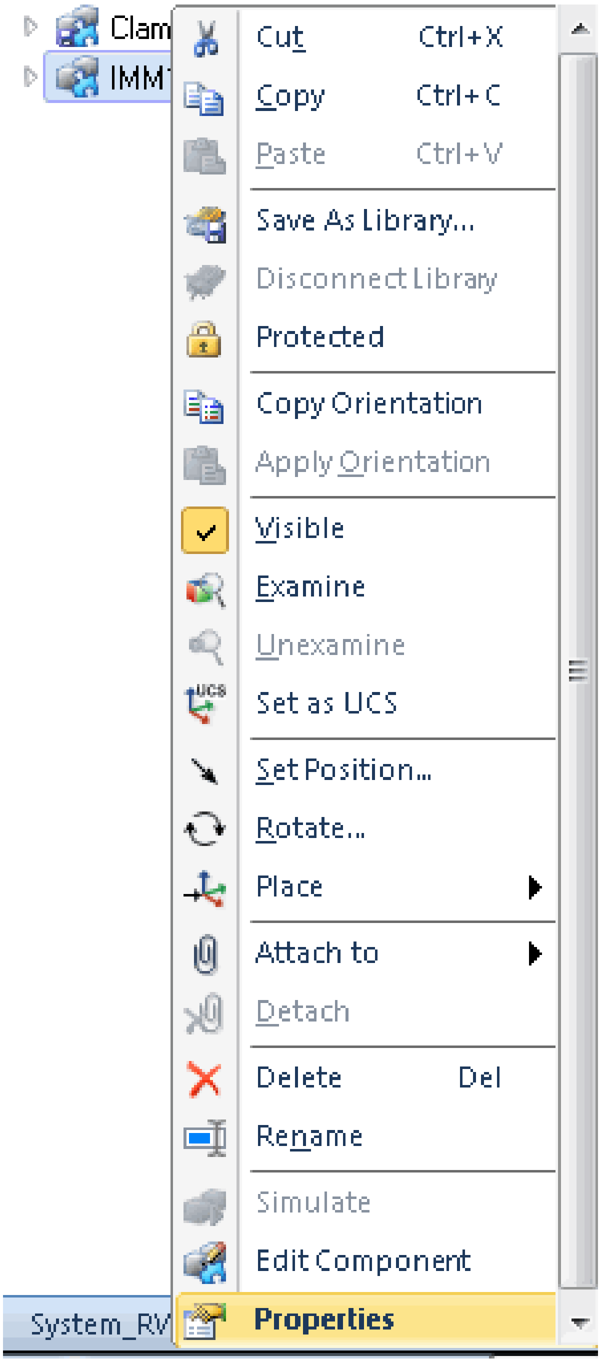
Continues on next page

	Action	Description
9	Click **Start**. Depending on the current position of the robot and whether there is a home run strategy configured, the robot either follows the strategy or directly moves to the home position.	**Note** It might be required to select the **Start** and **Reach Home** more than once to move the Robot. This behavior is the same as on the RobotWare Machine Tending user interface on the FlexPendant where there is a confirmation for every operation.
10	Repeat the steps 6 and 7 to start simulating the cycles.	
11	Click **Stop After Cycle**. The currently running cycle is completed and moves the robot to the home position.	 xx1300000533
12	Click **Stop**. The robot stops immediately and the program can be restarted again.	 xx1300000535

Continues on next page

Starting simulation for station smart components

While simulating some station smart components for the first time, it might be needed to set some signals to trigger the operation. There are unique signals for each smart component which can be accessed from the context menu, Properties sections for each smart component as shown in the following image.



xx1300000536

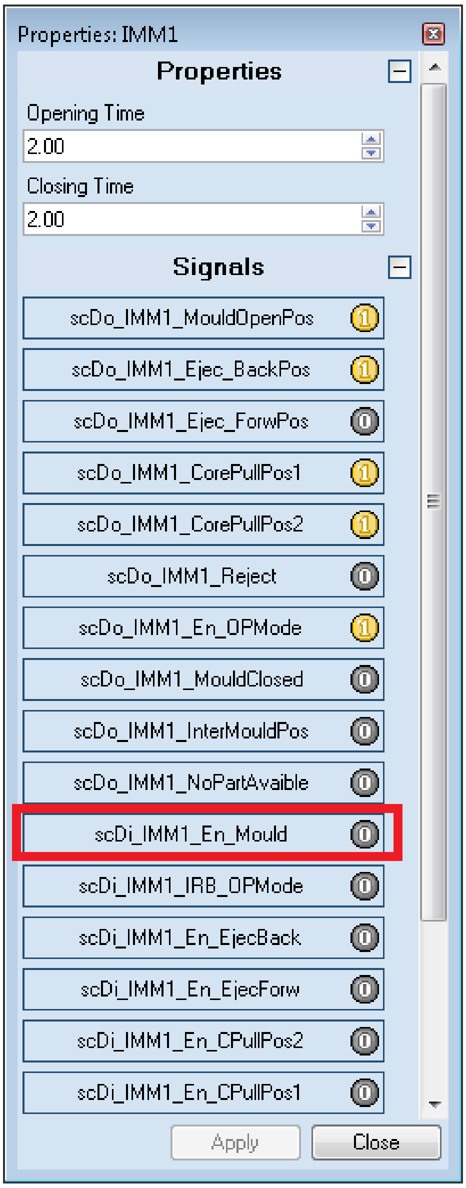
The different signals to be selected for smart components are:

- For Injection Moulding Machine

Continues on next page

The IMM starts the operations to close the mould by selecting the signal `scDi_%STATION%_En_Mould`. This signal is to be set or reset for the first time the IMM simulation is executed. Subsequently the machine is handled by the cycles.

For example, `scDi_IMM1_En_Mould`



Properties: IMM1

Properties

Opening Time: 2.00

Closing Time: 2.00

Signals

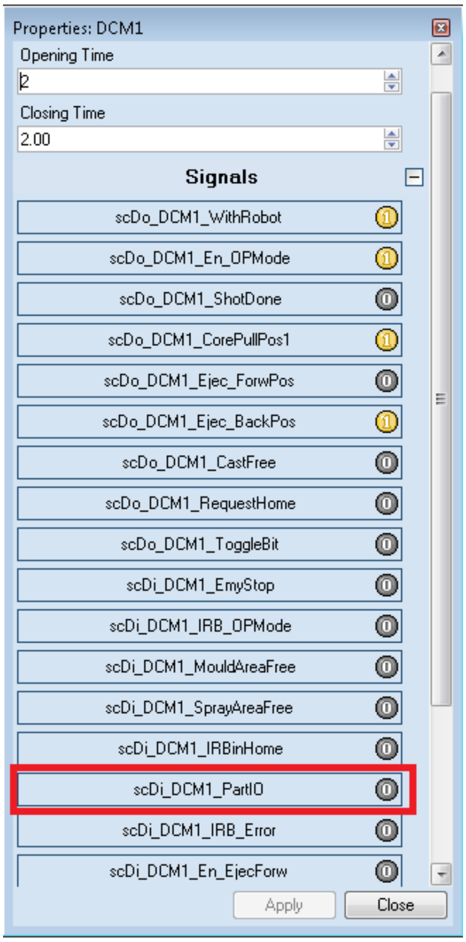
scDo_IMM1_MouldOpenPos	1
scDo_IMM1_Ejec_BackPos	1
scDo_IMM1_Ejec_ForwPos	0
scDo_IMM1_CorePullPos1	1
scDo_IMM1_CorePullPos2	1
scDo_IMM1_Reject	0
scDo_IMM1_En_OPMode	1
scDo_IMM1_MouldClosed	0
scDo_IMM1_InterMouldPos	0
scDo_IMM1_NoPartAvaible	0
scDi_IMM1_En_Mould	0
scDi_IMM1_IRB_OPMode	0
scDi_IMM1_En_EjecBack	0
scDi_IMM1_En_EjecForw	0
scDi_IMM1_En_CPullPos2	0
scDi_IMM1_En_CPullPos1	0

xx1300000537

- For Die Casting Machines

Continues on next page

The signal `scDi_%xxxxxx%_PartIO` is used to start the next cycle for the Die casting machine. This signal is usually set by the cycle.



Properties: DCM1

Opening Time
2

Closing Time
2.00

Signals

scDo_DCM1_WithRobot	1
scDo_DCM1_En_OPMode	1
scDo_DCM1_ShotDone	0
scDo_DCM1_CorePullPos1	1
scDo_DCM1_Ejec_ForwPos	0
scDo_DCM1_Ejec_BackPos	1
scDo_DCM1_CastFree	0
scDo_DCM1_RequestHome	0
scDo_DCM1_ToggleBit	0
scDi_DCM1_EmyStop	0
scDi_DCM1_IRB_OPMode	0
scDi_DCM1_MouldAreaFree	0
scDi_DCM1_SprayAreaFree	0
scDi_DCM1_IRBinHome	0
scDi_DCM1_PartIO	0
scDi_DCM1_IRB_Error	0
scDi_DCM1_En_EjecForw	0

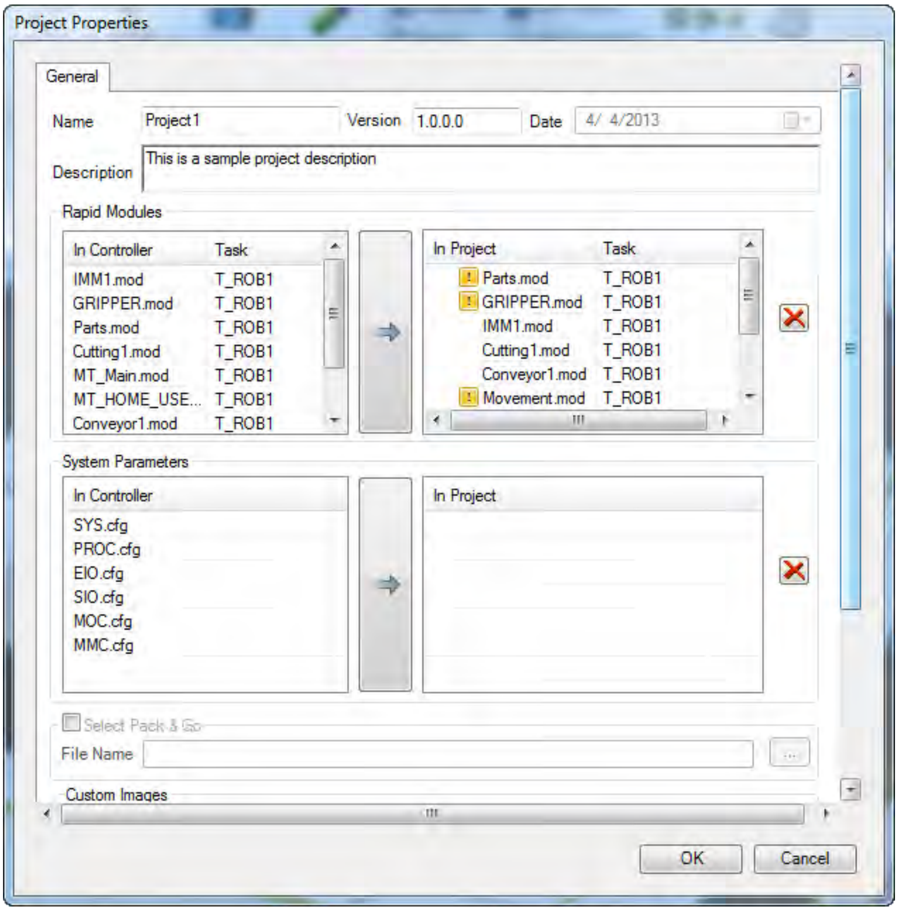
xx1300000572

4.8 Project

4.8.1 Project properties

The **Project Properties** option allows you to manage the project. For more details refer, *Project concept on page 14*.

Click the **Properties** button Properties. The **Project Properties** window is displayed.



Project Properties

General

Name: Project1
Version: 1.0.0.0
Date: 4/ 4/2013
Description: This is a sample project description

Rapid Modules

In Controller	Task
IMM1.mod	T_ROB1
GRIPPER.mod	T_ROB1
Parts.mod	T_ROB1
Cutting1.mod	T_ROB1
MT_Main.mod	T_ROB1
MT_HOME_USE...	T_ROB1
Conveyor1.mod	T_ROB1

In Project	Task
Parts.mod	T_ROB1
GRIPPER.mod	T_ROB1
IMM1.mod	T_ROB1
Cutting1.mod	T_ROB1
Conveyor1.mod	T_ROB1
Movement.mod	T_ROB1

System Parameters

In Controller
SYS.cfg
PROC.cfg
EIO.cfg
SIO.cfg
MOC.cfg
MMC.cfg

In Project

□ Select Pack & Go

File Name:

xx1300000452

You can perform the following tasks to manage a selected project.

- To rename the project, type the new name in the **Name** text box.

  On clicking **OK**, the project name in the browser tree, folder name, and `*.mtp` file is updated.
- To change the version number of the project , type the new version number in the **Version** text box.

Continues on next page

- To change the project description, type new description in the **Description** text box.

> **Note**
>
> Project description is limited to 78 characters.

- To add RAPID modules from the controller to the selected project, in the **Rapid Modules** section, select the module in the **In Controller** column and click the arrow button.

  The selected module is added to the **In project** list box.

- To delete a RAPID modules from the selected project, in the **Rapid Modules** section, select the module in the **In Project** column and click ✖. It is only possible to delete the modules added from the controller list, in the project properties dialog.

> **Note**
>
> To delete a module entry which is also listed in the browser tree of the Project requires it to be deleted from the browser tree. This will automatically update the Project properties.

- To add configuration parameters from the controller to the selected project, in the **System Parameters** section, select the parameter in the **In Controller** column and click the arrow button.

  The selected configuration parameter is added to the **In project** list box.

- To delete a configuration parameter from the selected project, in the **System Parameters** section, select the parameter in the **In Project** column and click ✖.

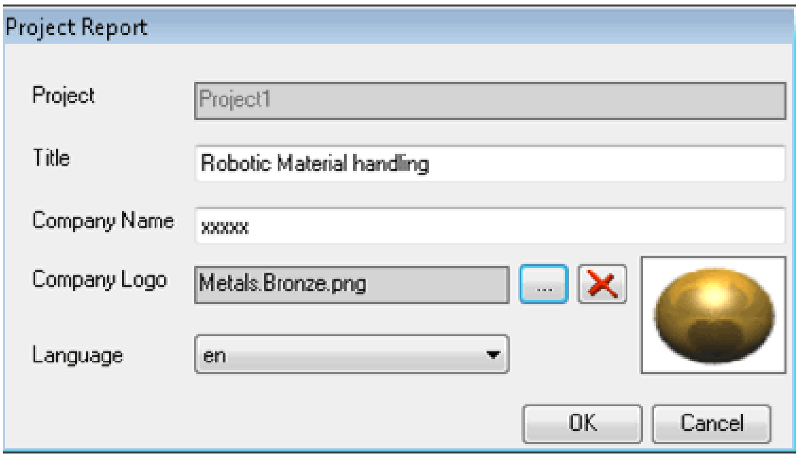
- It is possible to select the name of the Pack & Go file associated with the Project.

- All the images corresponding to the Project are saved in the Images folder under the Project. The names of all the images are listed in the `Project.mtp` file. From the Properties dialog, it is possible to add or delete images to the folder and update the `Project.mtp` file.

4.8.2 Project report

Generate reports

This feature generates Project reports in word format for reference purpose. It contains the information configured from MTPP and gives a guideline to include the additional details about the Project.



Project Report

Project: Project1

Title: Robotic Material handling

Company Name: xxxxx

Company Logo: Metals.Bronze.png

Language: en

xx1300000538

Some information presented in the Report:

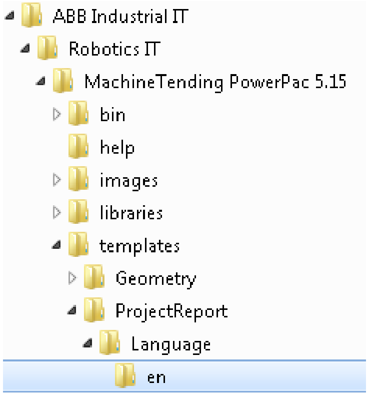
- Project Description and details about different Modules
- Position Description
- RobotWare Machine Tending Data type declarations
  - StationData, PartData, GripData, CycleData
- HomeRun strategy described in an image.
- Examples on
  - How to pictorically describe the stations
  - Textual description of different cycles

This information is expected to be filled by the user for the current application.

To generate a Report, follow the steps listed below:

Step	Operation	Result
1	Click **Reports**. The **Project Report** window is displayed.	Reports dialog opens.
2	Type a title for the report in the **Title** text box.	The title is shown in the first page.
3	Type the name of the company in the **Company Name** text box.	
4	Select a logo from the **Company Logo** list.	The name of the company and the logo are used in the footer section of the report.
5	Select a language from the **Language** drop-down list.	

Continues on next page

Step	Operation	Result
6	Click **OK**.	The report is generated from a sample template document (`sample.doc`) placed in the language specific folder in the installation folder of the Machine Tending PowerPac.  ABB Industrial IT Robotics IT MachineTending PowerPac 5.15 bin help images libraries templates Geometry ProjectReport Language en xx1300000539 You can translate the existing template to a different language and place it in the corresponding language folder and select the same from the dialog.
7	Select to view the report	The report is displayed.

4.9 Transfer

4.9.1 Save project

In the **Transfer** group, click **Save Project**.

When you click the **Save Project** button the following changes are saved in the project folder:

- All the Program Modules listed in Project properties (`Project.mtp`) are saved from Controller into the file system under: `~\<System Name>\HOME\RobotWare Machine Tending\Projects\<Project Name>\RAPID\<Active Task Name>\`
- Custom images, added from Project Properties user interface, are saved under: `~\<System Name>\HOME\RobotWare Machine Tending\Projects\<Project Name>\Images\`
- Pack & Go, added from the Project Properties user interface, are saved under: `~\<System Name>\HOME\RobotWare Machine Tending\Projects\<Project Name>\PackNGo`

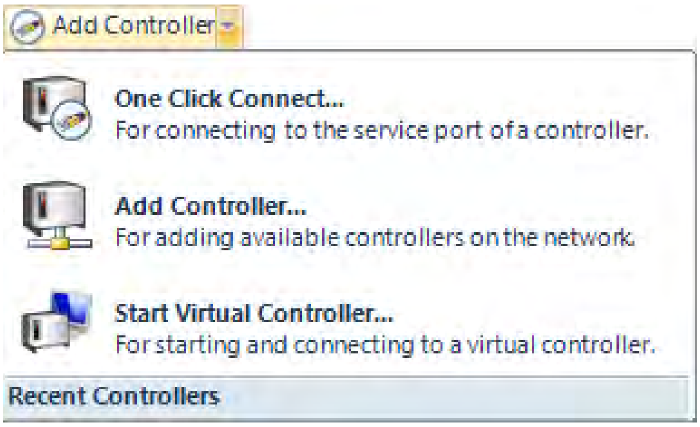
Note

To save configuration files, you have to manually save them from the controller.

Saving the project prepares the project folder with content which can be downloaded to the real controller.

4.9.2 Add controller

In the **Transfer** group, click **Add Controller**. The **Add Controller** list is displayed.



Add Controller

One Click Connect...
For connecting to the service port of a controller.

Add Controller...
For adding available controllers on the network.

Start Virtual Controller...
For starting and connecting to a virtual controller.

Recent Controllers

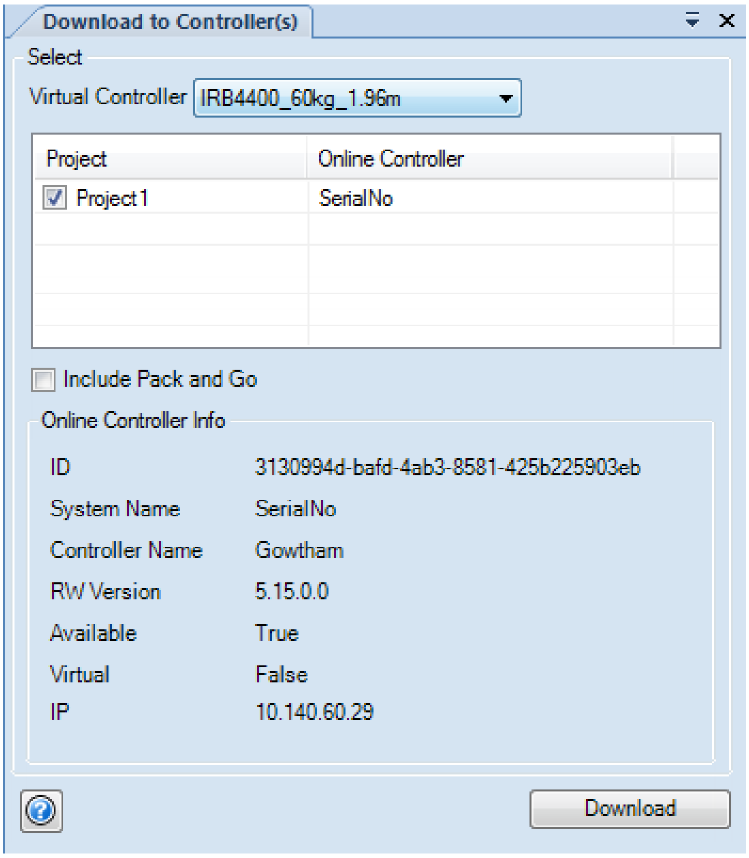
xx1300000411

The **Add Controller** list has the following options:

- **Add Controller** - For adding available controllers to the network.
- **One Click Connect** - For connecting to the service port of the controller.
- **Start Virtual Controller** - For deploying a project to the VC.

4.9.3 Download

In the **Transfer** group, click the **Download** button Download. The **Download to Controller(s)** window is displayed. The available virtual controllers and online controllers are listed in the table.



Virtual Controller: IRB4400_60kg_1.96m

Project	Online Controller
☑ Project1	SerialNo

□ Include Pack and Go

Online Controller Info

ID	3130994d-bafd-4ab3-8581-425b225903eb
System Name	SerialNo
Controller Name	Gowtham
RW Version	5.15.0.0
Available	True
Virtual	False
IP	10.140.60.29

xx1300000412

The download includes:

- Transferring the Project folder and its contents to the selected controller. The project is copied under the system folder.
- If the **Include Pack and Go** check box is selected, then the `*.rspag` is copied under the `.../hd0a/MTPP/` folder. It is not included in the system folder to avoid issues while taking system backup.

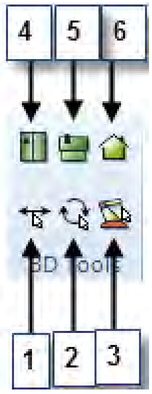
4.10 3D tools

4.10.1 Using the options in the 3D tools group

Introduction

The **3D Tools** group allows you to manage the movement of the robot using freehand tools and manage the view of the robot system.

The following figure and table provides information regarding different elements in the **3D Tools** group.



xx1300000511

Label	Button name	Description
1	**Move**	Allows you to drag an item relative to the active reference coordinate system.
2	**Rotate**	Allows you to enable rotation around the various axes of an object determined by the reference coordinate system.
3	**Jog Joint**	Allows you to jog the different axes of a robot.
4	**Top**	Displays the top view of the station with reference to the selected coordinate system.
5	**Front**	Displays the front view of the station with reference to the selected coordinate system.
6	**Right**	Displays the right view of the station with reference to the selected coordinate system.

The following sections provide information regarding using the tools in the 3D tools group.

Moving an item

1. In the **Layout** browser, select the item you want to move.
2. Click **Move**.
3. In the graphics window, click one of the axes and drag the item into position.

Rotating an item

1. In the **Layout** browser, select the item you want to rotate.
2. Click **Rotate**.

Continues on next page

3 In the graphics window, click one of the rotational rings and drag the item into position.

If you press the **ALT** key while rotating, the item will snap 10 degrees at a time.

Jogging the joints of a robot

1 In the **Layout** browser, select the robot you want to move.

2 Click **Jog Joint**.

3 Click the joint you want to move and drag it to the preferred position.

If you press the **ALT** key when jogging the joints of the robot, the robot will move 10 degrees at a time. If you press the **f** key, the robot will move 0.1 degree at a time.

This page is intentionally left blank

Index

Contact us

ABB AB, Robotics
Robotics and Motion
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics
Robotics and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics and Motion
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics and Motion
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

www.abb.com/robotics