

---

**Autodesk® PowerMill®**

# **Robot**

**Robot Postprocessor 3.0**



# PowerMill Robot Postprocessor 3.0

This document gives an overview of "PowerMill Robot Postprocessor", usually called "PRIPost" and lists the variables available in the postprocessors.

## 1. What is PRIPost?

PRIPost 3 is the third major version of the PowerMill Robot PostProcessor engine.

This version allows PowerMill Robot to create an NC Program for robot by creating main programs and sub programs.

Tool changes, large toolpath programs splitting (including retraction and plunge moves), spindle speed change... are available.

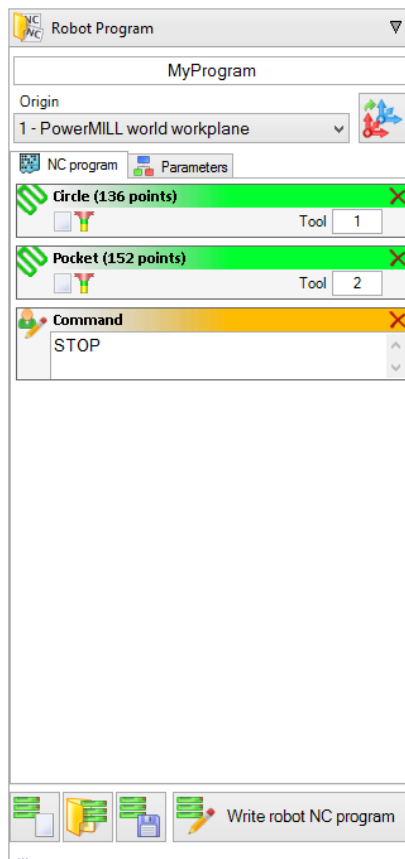
PRIPost 3 also allows the user to add user defined actions and transitions between toolpaths. It can also evaluate functions (using PowerMill Macro language or VB.NET)

The file extension of these postprocessors is *\*.PRIPost*

Since PRIPost 2, the *\*.RobConfig* file is now only used to store the settings of the robot cell (spindle calibrations, tools, cell setups...)

**Warning:** PRIPost 3 does not support Imperial units (Inches) yet. Ensure PowerMill is running in mm.

Below is a preview of PowerMill Robot (v8.5.02) program tab:



## 2. PostProcessor Structure

### 2.1 File format

PRIPost postprocessors use XML file format.

```
<?xml version="1.0" encoding="utf-8"?>
```

### 2.2 File version and robot information

The header contains the PRIPost engine version, the robot information as well as the spindle/part setting.

If the spindle is on the robot and the part is external then `Part="TABLE"`.

If the part is on the robot and the spindle is external then `Part="HEAD"`.

The "Robot Manufacturer" is requested in order to use the correct calculation methods during post-processing.

```
<Config>
  <PRIPost Version="2.2" />
  <Robot Manufacturer="<Robot Manufacturer>" Name="<Robot Name>" Option="" Part="TABLE" />
```

The `Option=""` attribute might be filled with some specific robot controller option to adjust some of the calculation – check the notes within the postprocessor file, if available.

*For example: KUKA Absolute Accuracy "AA" robots have some specific configuration calculation. In such case `Option="AA"` is required.*

### 2.3 Postprocessor compatibility

The postprocessor compatibility needs to be defined. Optionally, E1 to E12 can be used if needed to define the limitations of this compatibility (in this example, E1 must be a linear and E2 must be a rotary axis). If the not used, the `En` attribute must remain empty.

```
<!-- Compatibility of the Postprocessor -->
<!-- ExternalAxesCompatibility="0|1|2|3|4|5|6" : number of external axes supported by this postprocessor -->
<Compatibility ExternalAxesCompatibility="0|1|2" E1="Linear" E2="Rotary" E3="" E4="" E5="" E6="" />
```

This is used to filter the postprocessor compatible with the current robot kinematic.

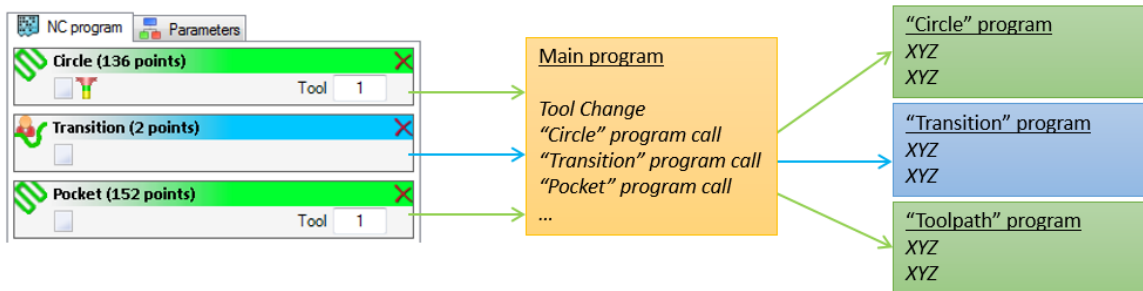
## 2.4 Output program types

There are currently two robot output program types available. This requires compatible postprocessors.

### 2.4.1 “Robot” program output type (default)

If `<Output Type="Robot"/>` (or if this is missing), the generated robot program is based on a main program calling sub programs (usually containing the points only). The main program also calls the tool change, fixture offset and spindle speed change programs for instance.

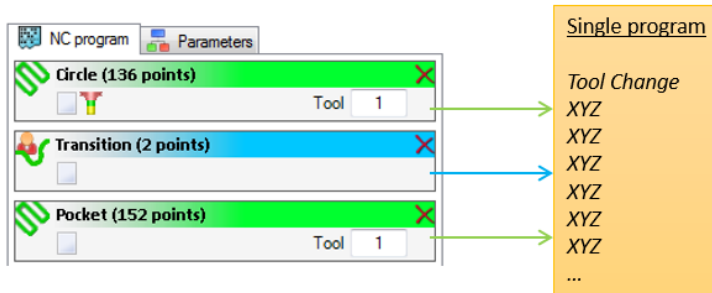
The structure looks like:



### 2.4.2 “CNC” program output type

If `<Output Type="CNC"/>`, the generated robot program is a single output file (like for CNC machines), which contains the points as well as the tool change, fixture offset and spindle speed change for instance.

The structure looks like:



This output is mainly based on Toolpath" file type (`<File Type="Toolpath">`)

#### Limitations:

- File splitting cannot be used so ensure the program size is below the robot memory limits or that the robot can run large programs (using a drip feed mechanism for instance)
- The feature `<Files ConciderTransitionAsToolpath="False">` is disabled: transitions are always considered as toolpath.
- “User defined command”, in NCProgram, are not supported.

## 2.5 User parameters

User defined variables can be created and then used in the postprocessor. These variables can be sorted by groups to make the reading easier.

If `Visible="True"`, the variable value can be change directly from the user interface. To display a list of items instead of a simple input textbox, the `Items="25|50|75|100|?"` attribute needs to be created and filled with the possible values. If one of the item is "?", the user can select this item to get an input textbox and write a non-listed value.

```
<UserParameters>
  <!--This parameter will be displayed as a list of items (Combobox) -->
  <UserParameter Name="JVelocity" Group="Robot" Visible="True" Variable="udp_JVel" Value="25" Items="25|50|75|100|?" Unit="%" Tag="" />
  <!--This parameter will be displayed as an input box (textbox) -->
  <UserParameter Name="Acceleration" Group="Robot" Visible="True" Variable="udp_Acceleration" Value="10" Unit="%" Tag="" />
  ...
</UserParameters>
```

To use a user defined variable in the postprocessor, it must use the prefix and suffix: "!"  
For instance, `Variable="udp_Acceleration"` can be used in the Postprocessor as `!udp_Acceleration!`  
This is true for all variable (system & user defined) used in PRIPost.

These user parameters variable format can be defined in the format area if required (valid for numerical variables only).

## 2.6 Point parameters

Point parameters are set to toolpath points through PowerMill. They are written into the \*.RobSim file during simulation.

These point parameters can then be written in the robot output program for further use...

If point parameters needs to be printed on the same line than the points coordinates, or anywhere else in the robot program, then a variable like `!udp_<MyPointParameter>!` can be used in the appropriate section. The `PrintPosition` must be set to "None" and the parameter can be initialize using the `InitValue` attribute, until the it's actually found in the simulation file (if not set on the first point). The content of the `CDATA[]` section is ignored in this case.

If the point parameter must be written "Before" or "After" the move, the `PrintPosition` attribute must be set accordingly and the `CDATA[]` section completed with the required text.

The point parameter can have multiple modes:

- `Mode="Always"`: The content of the `CDATA[]` section is written for each point.
- `Mode="IfChanged"`: The content of the `CDATA[]` section is written only when the value changes.
- `Mode="IfAvailable"`: The content of the `CDATA[]` section is written only when the point parameter is available, even if its value is the same. When the parameter is not available, its value is the `InitValue`.

```
<PointParameters>
  <PointParameter Name="PRESSURE" Enabled="True" PrintPosition="After" Variable="udp_PRESSURE"
    Mode="IfChanged" InitValue="999" PowerMillParameter="" FilesID="">
    <![CDATA[Pressure = !udp_PRESSURE!]]> <!-- This will be printed after the point, if the value changes -->
  </PointParameter>
</PointParameters>
```

If a **PowerMillParameter** attribute is filled with a PowerMill parameter, then the parameter value is extracted from PowerMill and patched into the simulation file (as new point parameters of the first point, when the simulation is saved). **PrintPosition="None"** is usually used in this case.

```
<PointParameters>
  <PointParameter Name="TOOL_GAUGE_LENGTH" Enabled="True" PrintPosition="None" Variable="udp_GAUGE_LENGTH"
    Mode="Always" InitValue="0" PowerMillParameter="gauge_length(Tool)" FilesID="">
    <![CDATA[]]> <!-- Nothing will be printed but the variable can be used in the postprocessor -->
  </PointParameter>
</PointParameters>
```

If one or a set of **FilesID** attribute are defined, then the **PrintPosition** set to **Before** or **After** is restricted to the specified files. To specify more than one file ID, use “|” as a separator (for example: **FilesID="TheToolpathFile|TheExtraFile"** covers two files with the respective IDs: “TheToolpathFile” and “TheExtraFile”).

The **Name** attribute cannot contain space or invalid character (\*, /, \...)

## 2.7 Format of the variables

The format of the system and user defined variables can be defined in the **<Formats>** area. Settings like **DecimalPlaces**, **TrailingZeros**, **WordLength**, **ScaleFactor** are available:

```
<Formats>
  <Format Variable="X" DecimalPlaces="3" TrailingZeros="True" WordLength="0" ScaleFactor="1" />
  <Format Variable="Y" DecimalPlaces="3" TrailingZeros="True" WordLength="0" ScaleFactor="1" />
  <Format Variable="Z" DecimalPlaces="3" TrailingZeros="True" WordLength="0" ScaleFactor="1" />
  ...
</Formats>
```

## 2.8 File splitting

If a **Split Points** value is set with a value higher than "0", the postprocessor splits the robot program files according to the number of points specified.

```
<Split Points="40000" RetractDistance="100" RetractFeedrate="500" PlungeFeedrate="500" ProgramIndexSeparator="_"
```

The **RetractFeedrate** and **PlungeFeedrate** are used to define the retract and plunge feed rate of the moves, if a **RetractDistance** is set. The **RetractFeedrate** and **PlungeFeedrate** units are *mm/min* and are converted to the defined robot feed rate.

**Note:** The split option is not compatible with ‘CNC’ output type.

## 2.9 Feedrates

The **Feedrate** parameters convert the default *mm/min* speed units from PowerMill, into the robot specific feedrates.

```
<!-- Unit= "m/s" or "mm/min" or "mm/s" -->
<Feedrate Unit="m/s" DecimalPlaces="4" UseAvailableFeedrates="False" AvailableFeedrates="100|500|1000|5000"/>
```

If the robot has only a given list of feedrates available, the **AvailableFeedrates** list can then be used. PRIPost then picks the “less or equal” feedrates in the available list.

## 2.10 Counters

Counters can be used by the postprocessor if required. They must be defined in the `<Counters>` section first. There is no limits in the number of counter than can be created and used. The `Index` must be unique to each counter.

To reset the counter for each toolpath, the visibility must be set as "Local" (default is "Global"). To reset the counter for each file, the visibility must be set as "File". This is generally used when toolpath programs are split.

```
<Counters>
  <Counter Index="0" StartValue="0" Increment="1" Visibility="Global" />
  <Counter Index="1" StartValue="0" Increment="1" Visibility="Local" />
  <Counter Index="2" StartValue="0" Increment="1" Visibility="File" />
</Counters>
```

The defined counters can then be used in the postprocessor as below:

<code>!Counter[0].Value!</code>	=> This prints the value of counter index 0
<code>!Counter[1].Value!</code>	=> This prints the value of counter index 1
<code>!Counter[0].Count!</code>	=> This increments the counter 0 and print its new value
<code>!Counter[0].FinalValue!</code>	=> This prints the last counter value before closing the file
<code>!Counter[0].FinalValueIndex0!</code>	=> This prints the last counter value before closing the file (but with one increment less)

## 2.11 Arc moves output

To allow arcs to be written in the robot program files, the arc output option must be enabled with: `<Arcs Enabled="True"...`

Some robots needs an "ArcMoveStart" call between two consecutive arcs. In such case, the attribute `AddArcMoveStartBetweenTwoConsecutivesArcs` needs to be set to "True".

Some others need a "LinearMove" after the arc: `AddLinearMoveAfterArc="True"` can then be used.

The description of the arc move can be defined in "Toolpath" and "Transitions" files types, in the same way as the Linear or Joint move, using the sections below:

```
<ArcMoveStart Enabled="True">
  <![CDATA[]]>
</ArcMoveStart>

<ArcMoveMid Enabled="True">
  <![CDATA[]]>
</ArcMoveMid>

<ArcMoveEnd Enabled="True">
  <![CDATA[]]>
</ArcMoveEnd>
```

If Arcs are not enabled (`<Arcs Enabled="False"...`), the postprocessor uses linear points to polygonise the arcs.

See the available Arc variables in the variable list...

## 2.12 Tool definition

If the `ToolDefinition` section is enabled (`Enabled="True"`) then the tool definition of each tool used in the PowerMill Robot NC Program can be automatically calculated and added to the output file.

```
<ToolDefinition Enabled="True">  
<![CDATA[TOOL(!Tool.Number!) = X!Tool.X!, Y!Tool.Y!, Z!Tool.Z!, R1!Tool.R1!, R2!Tool.R2!, R3!Tool.R3!]]>  
</ToolDefinition>
```

**Warning:** For the tool definitions to be calculated and added to the robot program, a Spindle Calibration must be performed and tools must be added with their actual length and correct number to the PowerMill Robot tool length database... or select "PowerMill tool length".. or the complete tool definitions must added to the PowerMill Robot tool database.

The tool definitions are printed in the output file where the variable `!Include_ToolDefinitions!` is set.

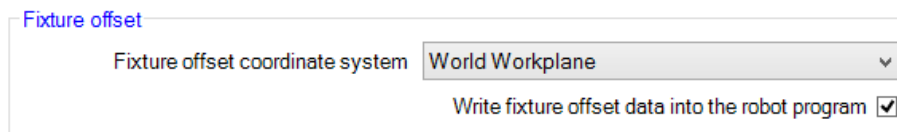
**Note:** Not all robot controllers support this feature. Please verify before enabling this functionality.

## 2.13 Fixture offset definition (part origin)

If the `FixtureOffsetDefinition` section is enabled (`Enabled="True"`) then the fixture offset definition selected in the PowerMill Robot NC Program can be automatically calculated and added to the output file.

```
<FixtureOffsetDefinition Enabled="True">  
<![CDATA[USER(!FixtureOffset.Number!) = X!FixtureOffset.X!, Y!FixtureOffset.Y!, Z!FixtureOffset.Z!,  
R1!FixtureOffset.R1!, R2!FixtureOffset.R2!,R3 !FixtureOffset.R3!]]>  
</ FixtureOffsetDefinition>
```

**Warning:** For the fixture offset definitions to be calculated and added to the robot program, the reference workplane (fixture offset coordinate system) must be selected in the robot configuration form:



Fixture offset

Fixture offset coordinate system World Workplane

Write fixture offset data into the robot program

The fixture offset definitions are printed in the output file where the variable `!Include_FixtureOffsetDefinitions!` is set.

**Note:** Not all robot controllers supports this feature. Please verify before enabling this functionality.

## 2.14 User Defined Action

The user can add between toolpath, in the PowerMill Robot NC Program, some user defined actions, when enabled: `UserDefinedAction Enabled="True"`.

If Enabled, the prefix and suffix are automatically added to each line of the user defined action set, to match with the robot output file format required.

```
<UserDefinedAction Enabled="True">
  <Note>
    <![CDATA[This is used when the user defined his own actions.
    Prefix and Suffix are added on each line if enabled.]]>
  </Note>

  <Prefix Enabled="True">
    <![CDATA[ !LineNumber!: ]]>
  </Prefix>

  <Suffix Enabled="True">
    <![CDATA[ ;]]>
  </Suffix>
</UserDefinedAction>
```

## 2.15 Home position

The home position can be called from the main program and generated in a sub program (using a point defined using joint angles). It could also simply be called in the main programs (in such case, the sub program containing the home position needs to be pre-defined in the robot controller but cannot be defined or changed by PowerMill Robot).

The home position, being defined as joint angles, is set in the `AxisMove` block and, if necessary, also in `AxisDefinition1`, `AxisDefinition2` and `AxisDefinition3` blocks.

See also "*Home*" file type (`<File Type="Home"`) section for more information.

## 2.16 Output files

Each entity starting with "`<File...`" creates files in the output directory.

The number of files that can be created is not limited so if the robot requests a special file then the user is free to add a new "`<File...`" section in the postprocessor.

**Note:** Each file must have a unique ID (`ID="<UniqueID">`). This ID can be text, numbers or a mix of text and numbers, can be defined by the user but MUST BE UNIQUE.

There are different kind of files that can be created:

### 2.16.1 "Main" file type (`Type="Main"`)

This file is the main program file. This file usually contains the tool and part origin definition and all the initializations. It also contains the calls of the sub program (the toolpaths), the tool changes, the spindle speed changes and the user defined actions...

As for the other file types, these files can be created in a special folder name if `CreateInSubFolder` is set to `True` and if a folder name is defined (`Folder=""`). This last field accepts variables.

The file `Extension` can be defined, as well as a file name. If the `FileName` is empty, the toolpath name is used.

User defined actions added to the NC program are added to the main file if the flag is enabled (`IncludeUserDefinedAction="True"`).

```
<File Type="Main" ID="<UniqueID>" FileName="!NCProgram.Name!" Extension=".txt" Folder=""  
CreateInSubFolder="False" IncludeUserDefinedAction="True">
```

The events available for this file type are:

ProgramStart, SpindleSpeedChange, ToolChangeFirst, ToolChange, NCEntityCall, SubProgramCallSingle, SubProgramCallFirst, SubProgramCall, SubProgramCallLast and ProgramEnd.

### 2.16.2 "Toolpath Main" file type (<File Type="ToolpathMain">)

This file is the toolpath program file (not the actual program file generated). It is generated once for each toolpath. This file usually contains the call of the sub programs for every single toolpath. As toolpaths can be split, this file can be used to call the sub programs generated for a single toolpath.

```
<File Type="ToolpathMain" ID="<UniqueID>" FileName="start" Extension=".txt" Folder="" CreateInSubFolder="False">
```

The sections available for this file type are the same than for the "Main" file type above.

Note: This file type is used by only a few robot languages.

### 2.16.3 "Transition" file type (<File Type="Transition">)

This file format is similar to the "**Toolpath**" file type.

If, in the main <Files> section, the attribute **ConciderTransitionAsToolpath** is set to **False**, then the postprocessor engine uses the Transition file type to write the transitions added to the NC program.

```
<Files ConciderTransitionAsToolpath="False">
```

This feature can be used if the transitions output file format needs another formatting than the toolpath files.

### 2.16.4 "Home" file type (<File Type="Home">)

This file format is similar to the "**Toolpath**" file type.

If, in the <Home> section, the attributes **Enabled** and **GenerateProgram** are set to **True**, then the postprocessor engine uses the home file type to write the home position and call it from the main program. If **GenerateProgram** is set to **False** the Home program is not generated so it must be defined in the controller first.

```
<Home Enabled="True" GenerateProgram="True" DefaultName="" Feedrate="5000"/>
```

The home feed rate (in mm/min) can also be set here. This is converted to the robot default speed unit during post processing.

The home move is described as joint move using joint angles in the **AxisMove** block (optionally **AxisDefinition1** can be used if the point description using joint position is different from a point description using Cartesian coordinates).

## 2.16.5 "Toolpath" file type (<File Type="Toolpath">

This file is the sub program file. This file usually contains the point coordinates and definition. This file is split to respect the `Split Points="40000"` value defined in the postprocessor settings.

**Notice:** The split option is not compatible with 'CNC' output type.

It also contains the retraction and plunge move in case this feature is enabled

(`RetractDistance="100"`).

```
<File Type="Toolpath" ID="<UniqueID>" PrintInMain="True" FileName="" Extension=".txt" Folder="" CreateInSubFolder="False">
```

The events available for this file type are:

ProgramStart, ProgramStartSplit, ToolpathStart, LinearMove, JointMove, AxisMove, PointDefinition1Start, PointDefinition1, AxisDefinition1, PointDefinition1End, PointDefinition2Start, PointDefinition2, AxisDefinition2, PointDefinition2End, PointDefinition3Start, PointDefinition3, AxisDefinition3, PointDefinition3End, ToolpathPointFirst, PlungeMoveStart, PlungeMoveEnd, LeadInMoveStart, LinearMoveStart, LinearMoveEnd, LeadOutMoveEnd, RapidMoveStart, RapidFeedMoveStart, ToolpathPointLast, SplitRetractionMoveRetractStart, SplitRetractionMoveRetractEnd, SplitRetractionMovePlungeStart, SplitRetractionMovePlungeEnd, FeedrateSet, ConfigurationSet, ToolpathEnd, ProgramEndSplit and ProgramEnd.

Drilling Cycle events are: CycleLinkMoveStart, CyclePlungeMoveStart, CycleRapidMoveStart, CycleOutputMoveStart, CycleLinkMoveEnd, CyclePlungeMoveEnd, CycleRapidMoveEnd and CycleOutputMoveEnd.

If the 'CNC' output type is set, a few more events are available, like:

SpindleSpeedChange, ToolChangeFirst, ToolChange, FixtureOffsetChangeFirst and FixtureOffsetChange.

If enabled, the point definitions (1 ,2, 3 and their "Start" and "End") are printed where the variables below are used:

```
!Include_PointDefinition1Start!, !Include_PointDefinition1!, !Include_PointDefinition1End!
!Include_PointDefinition2Start!, !Include_PointDefinition2!, !Include_PointDefinition2End!
!Include_PointDefinition3Start!, !Include_PointDefinition3!, !Include_PointDefinition3End!
```

The `LinearMove` output syntax can be different according to the type of move of the PowerMill point: the `MoveType` attribute needs to be added to a new linear move block.

```
<!--This will be written for MoveType 4, 0 and 5-->
<LinearMove MoveType="4|0|5">
  <![CDATA[Cutting (!X!,!Y!,!Z!,!R1!,!R2!,!R3!)]]>
</LinearMove>

<!--This will be written for all of the other MoveType-->
<LinearMove>
  <![CDATA[Linear (!X!,!Y!,!Z!,!R1!,!R2!,!R3!)]]>
</LinearMove>
```

Type of move	Num	Type of move	Num
Feed Move	0	Cycle Rapid Move	8
Rapid Move	1	Cycle Output Move	9
Rapid Feed Move	2	Toolpath Join Move	10
Plunge Move	3	Multi-axis Toolpath Join	11
Lead In Move	4	Additive Move	12
Lead Out Move	5	Probing Measure Move	13
Cycle Link Move	6	Probing Safe Move	14
Cycle Plunge Move	7	Probing Rapid Move	15

### 3. Multi-Axis Coordinate transform

The *Multi-Axis Coordinate transform* feature is designed to compensate for "non-linked" external axes workplane, i.e. when the position of the robot TCP (tool tip coordinates) needs to be recalculated according to the external axis position.

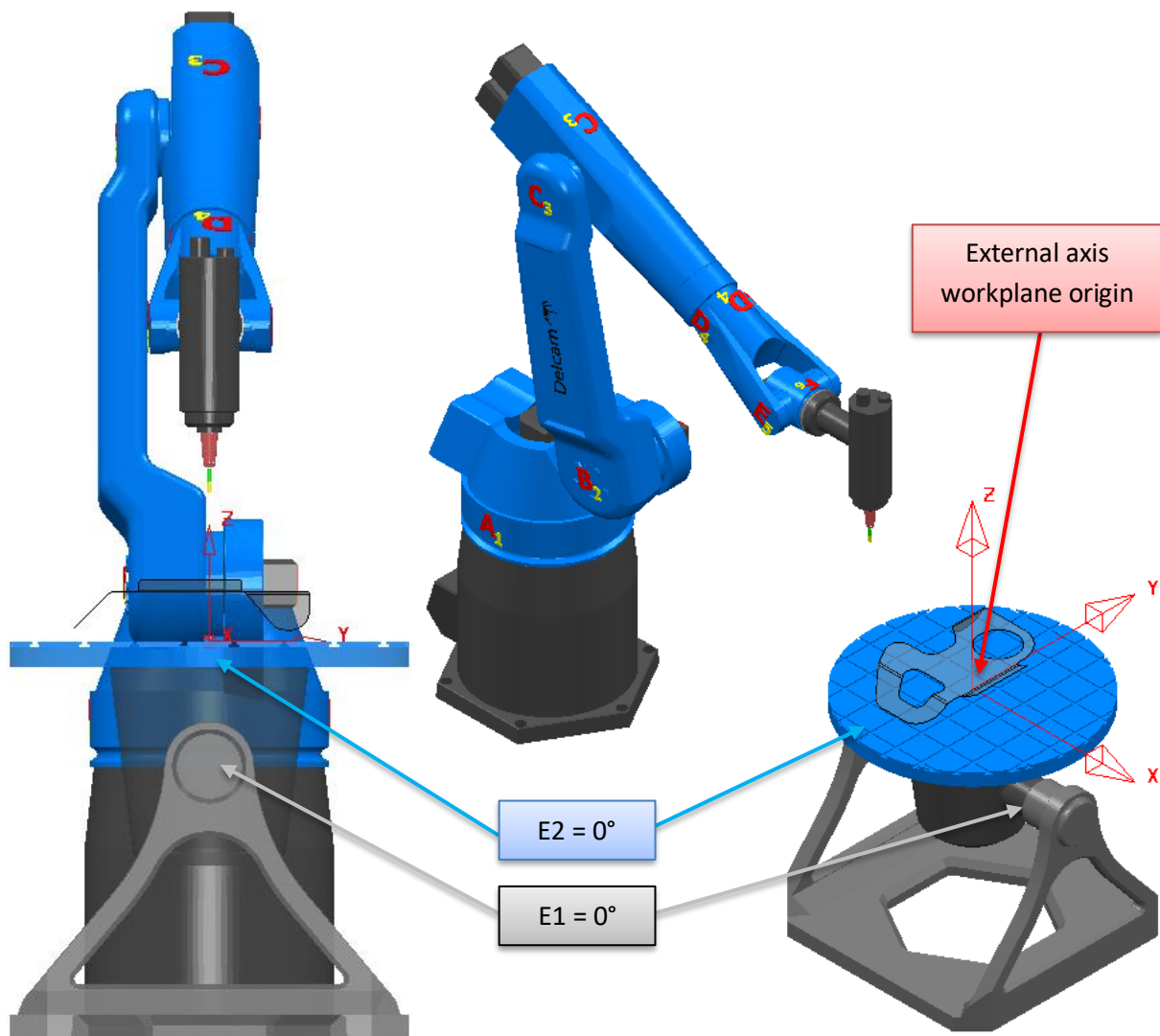
This is generally used when rotary table workplane do not follow the table rotation, but this functionality can also be used with linear tracks.

PowerMill Robot default output is set for robot using "linked" table workplane.

#### 3.1 How to determine if the external axis workplane is "linked"?

The images below describe the position of the external positioner, at its "zero" position.

Note: The table workplane (in red) is set at its initial position, at the center of the rotary table. Its direction is similar to the robot world workplane direction.

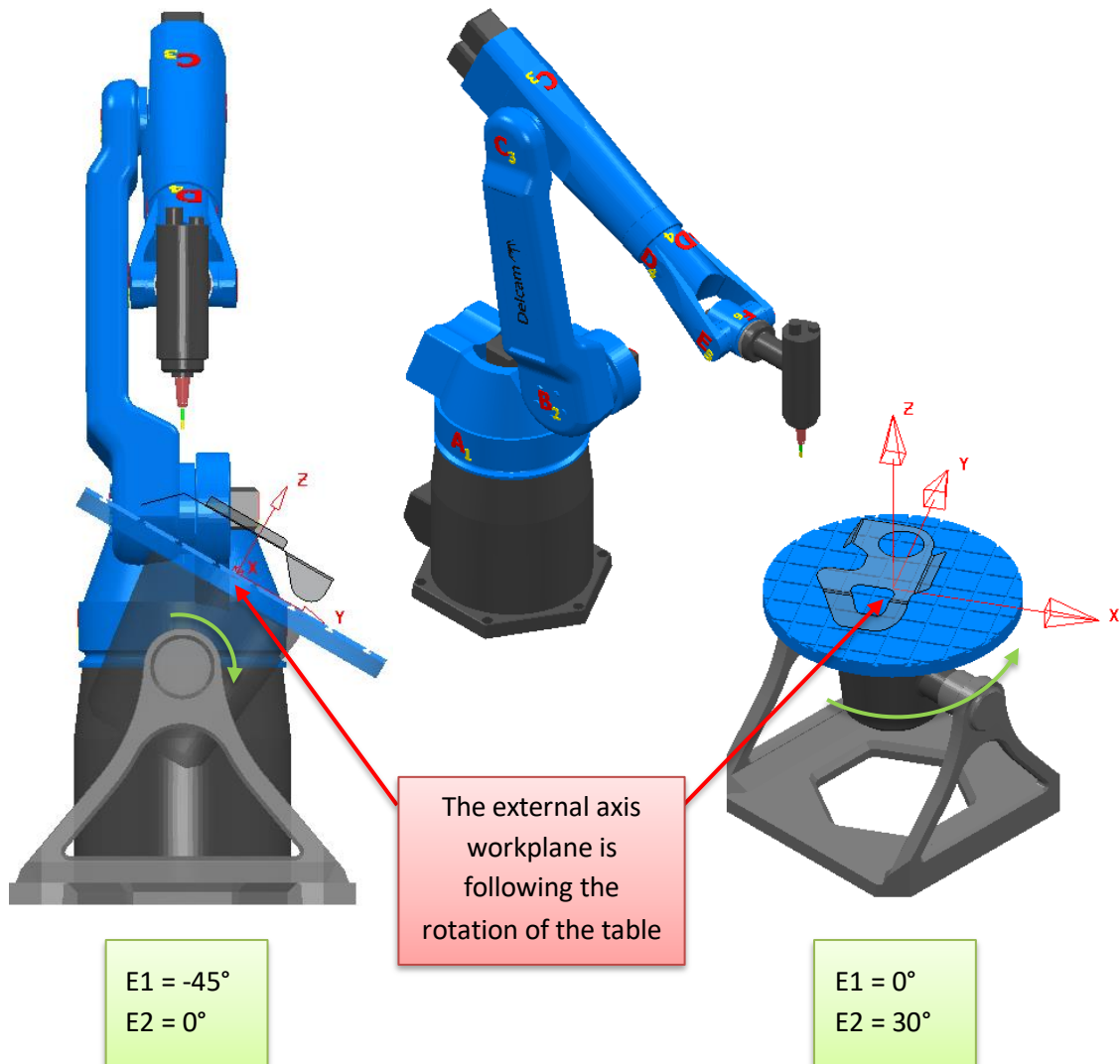


### 3.1.1 The external axis workplane is linked to the table

The images below describe the situation where the table workplane is linked to the table.

In such case the *Multi-Axis Coordinate transform* feature doesn't need to be enabled as this is what PowerMill Robot outputs by default: `<CoordinateTransform Enabled="False">`... The robot controller is doing the calculation.

Note: The table workplane (in red) is following the table during the rotations.

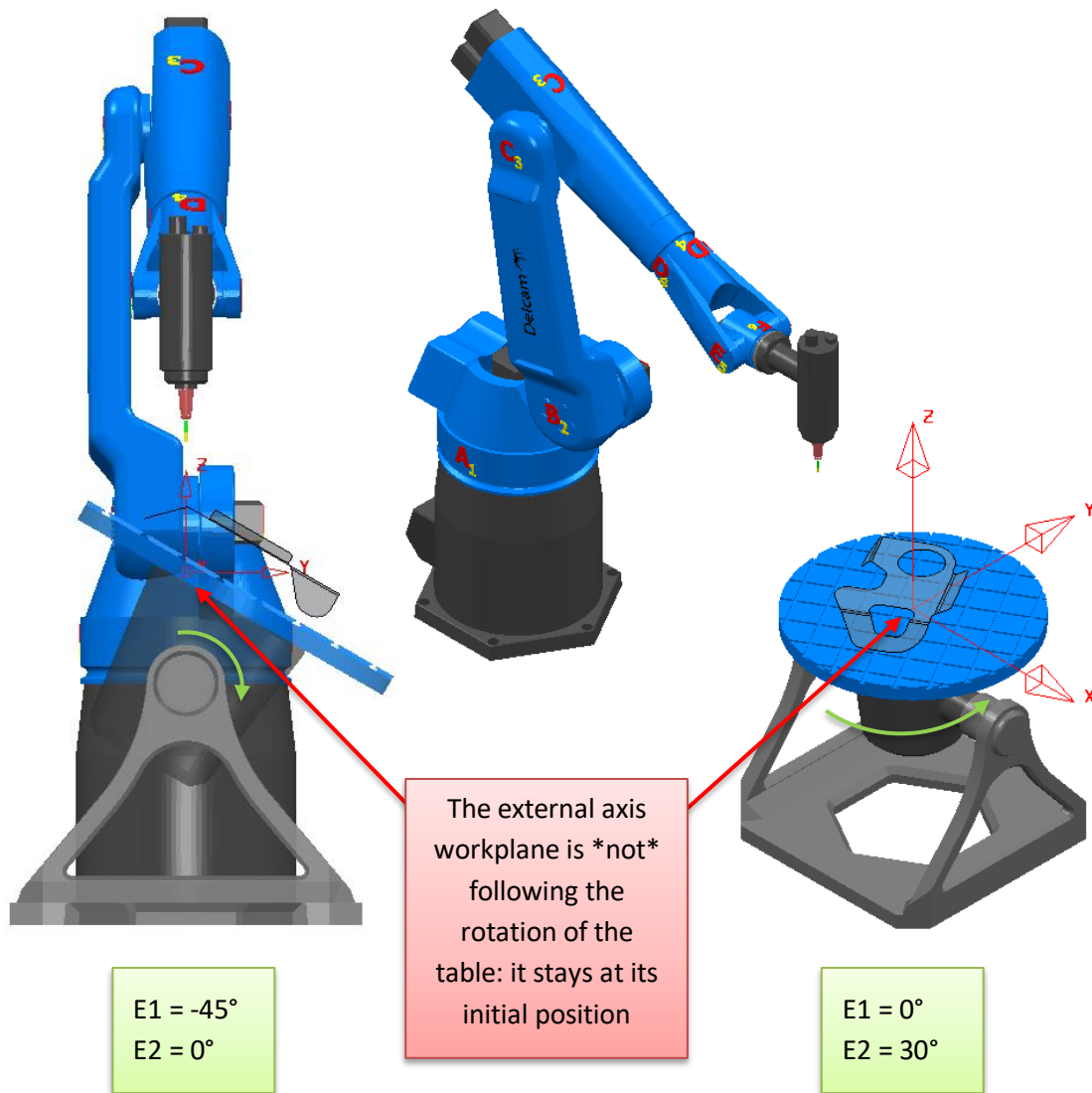


### 3.1.2 The external axis workplane is \*NOT\* linked to the table

The images below describe the situation where the table workplane is is not linked to the table.

In such case the *Multi-Axis Coordinate transform* feature needs to be enabled to recalculate the robot TCP (tool tip position and orientation): `<CoordinateTransform Enabled="True">...`  
The robot controller is not doing the calculation. The calculation is done by the postprocessor.

Note: The table workplane (in red) is not following the table during the rotations.



## 3.2 Example (the multi-axis coordinate transform feature is enabled)

```
<!-- Coordinate transform / Axis synchronisation simulation for robot without synchronised external axes -->
<!-- Enabled = "False" / "True" => Is the global compensation enabled -->
<!-- TransformOrder = "Auto" = "E12|E11|E10|E9|E8|E7|E6|E5|E4|E3|E2|E1 " / or any other order => Define the order of the
transformations to apply for the compensation -->
<CoordinateTransform Enabled="True" TransformOrder="E12|E11|E10|E9|E8|E7|E6|E5|E4|E3|E2|E1">
  <!-- Enabled = "False" / "True" => Is the compensation enabled on this axis -->
  <!-- Direction = "Positive" / "Negative" => Direction of the compensation -->
  <!-- Coord = "Auto" / "Manual" => Pivot point calculated from MTD file or defined by the user -->
  <!-- X, Y, Z (double) => Coordinate of the pivot point (when defined by user) -->
  <!-- I, J, K (vector) => Direction of the pivot point (when defined by user) -->
  <E1 Enabled="True" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E2 Enabled="True" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E3 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E4 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E5 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E6 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E7 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E8 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E9 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E10 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E11 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
  <E12 Enabled="False" Direction="Positive" Coord="Auto" X="0" Y="0" Z="0" I="1" J="0" K="0" />
</CoordinateTransform>
```

The example above shows that:

- The *Multi-Axis Coordinate transform* feature is turned ON

→ `<CoordinateTransform Enabled="True"`

- The order of transformations is set to its default value

→ `<CoordinateTransform TransformOrder="E12|E11|E10|E9|E8|E7|E6|E5|E4|E3|E2|E1"`

This describes the order of transformation to apply to the TCP point. Usually the order is from the further to the nearest axis, from the robot world workplane, in the kinematic chain.

- External axes 1 and 2 are used for the TCP compensation calculation

→ `<E1 Enabled="True"...`

→ `<E2 Enabled="True"...`

- The compensation direction is similar to the one defined in the \*.mtd file (same axis direction)

→ `<E1 Direction="Positive"`

→ `<E2 Direction="Positive"`

- The pivot points of the external axes used are the one defined in the \*.mtd file

→ `<E1 Coord="Auto"`

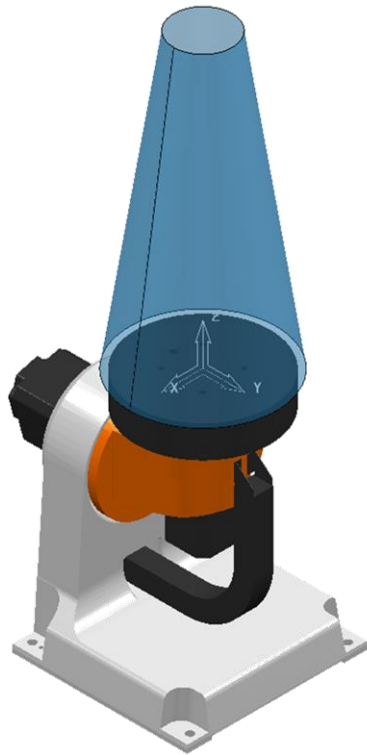
→ `<E2 Coord="Auto"`

When `Coord="Auto"`, the postprocessor is using the values defined in the \*.mtd file. The `X="0" Y="0" Z="0" I="1" J="0" K="0"`, defined in the postprocessor, are ignored.

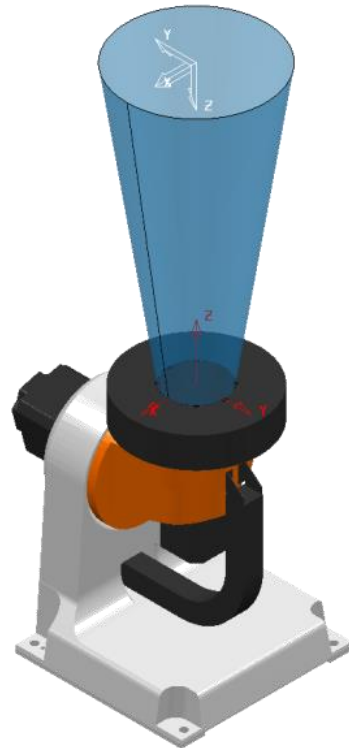
When `Coord="Manual"`, the postprocessor is using the values defined by the user in the postprocessor. The `X="0" Y="0" Z="0" I="1" J="0" K="0"` coordinates in the postprocessor are used, the \*.mtd file values are ignored.

These values represents the position and direction of the external axis pivot point, from the robot world workplane.

**WARNING:** When using this compensation with a rotary table, the *table\_attach\_point* **must** be set at the center of the rotary table and used as **output** workplane. The part must be moved to the correct position in the robot cell, without changing robot "Origin" (The "part positioner" cannot be used).



Supported



Not Supported

## 4. Function evaluator

Functions can be evaluated within the postprocessor. The supported languages are “PowerMill Macro” and “VB.NET”.

**Note:** To evaluate PowerMill Macro language functions, the post processing needs to be done from PowerMill Robot. PowerMill Macro code evaluator cannot be used if PRIPost is used by a third party application as it needs PowerMill to evaluate the function.

VB.NET function can be used with or without PowerMill. This can therefore be used if PRIPost is driven by a third party application.

To evaluate VB.NET function, the code needs to be compiled at run time, each time it is called. This compilation takes time and this can slow down the post processing. If possible, avoid using VB.NET in the `<LinearMove>` block for instance, as this is called for each point. In such case it's more efficient to use PowerMill Macro language as it is evaluated by PowerMill without the need of being compiled. PowerMill Macro code is faster to evaluate than VB.NET code.

**Example:** Function used in the `<ToolChange>` block to write the relevant text, based on the current tool number:

Function using PowerMill Macro language	Text printed in the program
<pre> &lt;ToolChange Enabled="True" Language="Macro"&gt;   &lt;![CDATA[ PRINT "UNLOAD_TOOL(!PreviousTool.Number!)" PRINT "LOAD_TOOL(!Tool.Number!)" SWITCH '!Tool.Number!' { CASE '1'   PRINT "COOLANT_FLOOD()"   BREAK CASE '2'   PRINT "COOLANT_MIST()"   BREAK DEFAULT   PRINT "COOLANT_STANDARD()"   BREAK } ]]&gt; &lt;/ToolChange&gt; </pre>	<pre> UNLOAD_TOOL(1) LOAD_TOOL(2) COOLANT_MIST() </pre>

Function using VB.NET language	Text printed in the program
<pre> &lt;ToolChange Enabled="True" Language="VB.NET"&gt;   &lt;![CDATA[ Dim L1 As String = "UNLOAD_TOOL(!PreviousTool.Number!)" Dim L2 As String = "LOAD_TOOL(!Tool.Number!)" Dim L3 As String = String.Empty Select Case "!Tool.Number!" Case "1"   L3 = "COOLANT_FLOOD()" Case "2"   L3 = "COOLANT_MIST()" Case Else   L3 = "COOLANT_STANDARD()" End Select Return (L1 &amp; Environment.NewLine &amp; L2 &amp; Environment.NewLine &amp; L3) ]]&gt; &lt;/ToolChange&gt; </pre>	<pre> UNLOAD_TOOL(1) LOAD_TOOL(2) COOLANT_MIST() </pre>

**Note:** Multiple “PRINT” commands can be used with PowerMill Macro code. However, only one “Return” value is permitted when using VB.NET.

## 5. Variables

Below is a list of variables that can be used in the postprocessor:

### Robot variables:

<b>!Robot.NumberOfAxes!</b>	Number of axes of the robot
<b>!Robot.NumberOfExternalAxes!</b>	Number of external axes
<b>!Robot.TotalNumberOfAxes!</b>	Number of axes in the robot cell (robot + external)
<b>!E1.Unit!</b>	Unit of the external axis 1 ("mm" for linear, "deg" for rotary)
<b>!E2.Unit!</b>	Unit of the external axis 2 ("mm" for linear, "deg" for rotary)
<b>!E3.Unit!</b>	Unit of the external axis 3 ("mm" for linear, "deg" for rotary)
<b>!E4.Unit!</b>	Unit of the external axis 4 ("mm" for linear, "deg" for rotary)
<b>!E5.Unit!</b>	Unit of the external axis 5 ("mm" for linear, "deg" for rotary)
<b>!E6.Unit!</b>	Unit of the external axis 6 ("mm" for linear, "deg" for rotary)
<b>!E7.Unit!</b>	Unit of the external axis 7 ("mm" for linear, "deg" for rotary)
<b>!E8.Unit!</b>	Unit of the external axis 8 ("mm" for linear, "deg" for rotary)
<b>!E9.Unit!</b>	Unit of the external axis 9 ("mm" for linear, "deg" for rotary)
<b>!E10.Unit!</b>	Unit of the external axis 10 ("mm" for linear, "deg" for rotary)
<b>!E11.Unit!</b>	Unit of the external axis 11 ("mm" for linear, "deg" for rotary)
<b>!E12.Unit!</b>	Unit of the external axis 12 ("mm" for linear, "deg" for rotary)
<b>!Home.A1!</b>	The Home value of the robot axis 1 (set in cell configuration)
<b>!Home.A2!</b>	The Home value of the robot axis 2 (set in cell configuration)
<b>!Home.A3!</b>	The Home value of the robot axis 3 (set in cell configuration)
<b>!Home.A4!</b>	The Home value of the robot axis 4 (set in cell configuration)
<b>!Home.A5!</b>	The Home value of the robot axis 5 (set in cell configuration)
<b>!Home.A6!</b>	The Home value of the robot axis 6 (set in cell configuration)
<b>!Home.E1!</b>	The Home value of the external axis 1 (set in cell configuration)
<b>!Home.E2!</b>	The Home value of the external axis 2 (set in cell configuration)
<b>!Home.E3!</b>	The Home value of the external axis 3 (set in cell configuration)
<b>!Home.E4!</b>	The Home value of the external axis 4 (set in cell configuration)
<b>!Home.E5!</b>	The Home value of the external axis 5 (set in cell configuration)
<b>!Home.E6!</b>	The Home value of the external axis 6 (set in cell configuration)
<b>!Home.E7!</b>	The Home value of the external axis 7 (set in cell configuration)
<b>!Home.E8!</b>	The Home value of the external axis 8 (set in cell configuration)
<b>!Home.E9!</b>	The Home value of the external axis 9 (set in cell configuration)
<b>!Home.E10!</b>	The Home value of the external axis 10 (set in cell configuration)
<b>!Home.E11!</b>	The Home value of the external axis 11 (set in cell configuration)
<b>!Home.E12!</b>	The Home value of the external axis 12 (set in cell configuration)

Tool definition variables:

<b>!Tool.Number!</b>	Tool number
<b>!Tool.Index!</b>	Tool index (internal 'counter' starting from index 0)
<b>!Tool.Name!</b>	Tool name (as set in PowerMill Robot Tool Database)
<b>!Tool.Name.NoSpace!</b>	Tool name without space
<b>!Tool.X!</b>	Tool workplane origin X
<b>!Tool.Y!</b>	Tool workplane origin Y
<b>!Tool.Z!</b>	Tool workplane origin Z
<b>!Tool.R1!</b>	Tool workplane rotation 1 (using robot convention)
<b>!Tool.R2!</b>	Tool workplane rotation 2 (using robot convention)
<b>!Tool.R3!</b>	Tool workplane rotation 3 (using robot convention)
<b>!Tool.Q1!</b>	Tool workplane quaternion 1 (if using Quaternion convention)
<b>!Tool.Q2!</b>	Tool workplane quaternion 2 (if using Quaternion convention)
<b>!Tool.Q3!</b>	Tool workplane quaternion 3 (if using Quaternion convention)
<b>!Tool.Q4!</b>	Tool workplane quaternion 4 (if using Quaternion convention)
<b>!Tool.OX.X!</b>	Tool workplane OX axis point coordinate X
<b>!Tool.OX.Y!</b>	Tool workplane OX axis point coordinate Y
<b>!Tool.OX.Z!</b>	Tool workplane OX axis point coordinate Z
<b>!Tool.OY.X!</b>	Tool workplane OY axis point coordinate X
<b>!Tool.OY.Y!</b>	Tool workplane OY axis point coordinate Y
<b>!Tool.OY.Z!</b>	Tool workplane OY axis point coordinate Z
<b>!Tool.Diameter!</b>	Diameter of the tool (from PowerMill Robot tool database) <i>if option is active</i>
<b>!Tool.Radius!</b>	Radius of the tool (Diameter / 2) <i>if option is active</i>
<b>!Tool.Length!</b>	Length of the tool (from PowerMill Robot tool database) <i>if option is active</i>
<b>!PreviousTool.Number!</b>	Previous tool Number
<b>!PreviousTool.Name!</b>	Previous tool Name (as set in PowerMill Robot Tool Database)
<b>!Tool.NumberOfTools!</b>	Number of tools used ( <i>accessible from toolpath or main program</i> )

Fixture offset definition variables:

!FixtureOffset.Number!	Fixture offset number
!FixtureOffset.Index!	Fixture offset index (internal 'counter' starting from index 0)
!FixtureOffset.Name!	Fixture offset name (as set in PowerMill Robot Tool Database)
!FixtureOffset.Name.NoSpace!	Fixture offset name without spaces
!FixtureOffset.X!	Fixture offset origin X
!FixtureOffset.Y!	Fixture offset origin Y
!FixtureOffset.Z!	Fixture offset origin Z
!FixtureOffset.R1!	Fixture offset workplane rotation 1 (using robot convention)
!FixtureOffset.R2!	Fixture offset workplane rotation 2 (using robot convention)
!FixtureOffset.R3!	Fixture offset workplane rotation 3 (using robot convention)
!FixtureOffset.Q1!	Fixture offset workplane quaternion 1 (if using Quaternion convention)
!FixtureOffset.Q2!	Fixture offset workplane quaternion 2 (if using Quaternion convention)
!FixtureOffset.Q3!	Fixture offset workplane quaternion 3 (if using Quaternion convention)
!FixtureOffset.Q4!	Fixture offset workplane quaternion 4 (if using Quaternion convention)
!FixtureOffset.OX.X!	Fixture offset workplane OX axis point coordinate X
!FixtureOffset.OX.Y!	Fixture offset workplane OX axis point coordinate Y
!FixtureOffset.OX.Z!	Fixture offset workplane OX axis point coordinate Z
!FixtureOffset.OY.X!	Fixture offset workplane OY axis point coordinate X
!FixtureOffset.OY.Y!	Fixture offset workplane OY axis point coordinate Y
!FixtureOffset.OY.Z!	Fixture offset workplane OY axis point coordinate Z
!FixtureOffset.NumberOffFixtureOffsets!	Number of fixture offsets used (accessible from toolpath or main program)

Counter variables ("n" represents the counter index):

!Counter[n].Value!	Value of the counter
!Counter[n].Count!	Increment the counter and returns the new value
!Counter[n].FinalValue!	Value of the counter before closing the file
!Counter[n].FinalValueIndex0!	<Value of the counter> - 1 <increment> (before closing the file)

Variables available for the "Main" program type only

!Programs.Count!	Number of programs created (printed in the main)
!Programs[n].FileName!	Program file name (index n)
!Programs[n].FileNameWithExtension!	Program file name with extension (index n)
!Programs[n].Extension!	Program file extension (index n)
!Programs.NoDuplicates.Count!	Number of programs created <u>without duplicates</u>
!Programs.NoDuplicates[n].FileName!	Program file name (index n) <u>without duplicates</u>
!Programs.NoDuplicates[n].FileNameWithExtension!	Program file name with extension (index n) <u>without duplicates</u>
!Programs.NoDuplicates[n].Extension!	Program file extension (index n) <u>without duplicates</u>

NCProgram/Toolpath variables:

<b>!MainProgram.Name!</b>	NC program name
<b>!NCProgram.Name!</b>	NC program name
<b>!Toolpath.Name!</b>	Toolpath name
<b>!Program.Name!</b>	Program name
<b>!Program.FileNameWithExtension!</b>	Program file name with extension
<b>!Program.FileName!</b>	Program file name without extension
<b>!Program.Extension!</b>	Program file extension
<b>!Feedrate.Cut!</b>	Cutting feedrate (not valid for multiple feedrate toolpath)
<b>!Feedrate.Plunge!</b>	Plunge feedrate
<b>!Feedrate.LeadIn!</b>	Lead In feedrate
<b>!Feedrate.LeadOut!</b>	Lead Out feedrate
<b>!Feedrate.RampLeadIn!</b>	Ramp Lead In feedrate
<b>!Feedrate.Rapid!</b>	Rapid feedrate
<b>!NCProgram.TimeD!</b>	NC program time in days
<b>!NCProgram.TimeH!</b>	NC program time in hours
<b>!NCProgram.TimeM!</b>	NC program time in minutes
<b>!NCProgram.Time!</b>	NC program time in seconds
<b>!NCProgram.TimeHMS!</b>	NC program time in “Hours:Minutes:Seconds” format
<b>!Toolpath.TimeD!</b>	Toolpath time in days
<b>!Toolpath.TimeH!</b>	Toolpath time in hours
<b>!Toolpath.TimeM!</b>	Toolpath time in minutes
<b>!Toolpath.Time!</b>	Toolpath time in seconds
<b>!Toolpath.TimeHMS!</b>	Toolpath time in “Hours:Minutes:Seconds” format
<b>!PreviousToolpath.Name!</b>	Previous toolpath name without extension
<b>!PreviousProgram.NameWithExtension!</b>	Previous program file name with extension
<b>!PreviousProgram.Name!</b>	Previous program file name without extension
<b>!Date!</b>	Date of the day (format set automatically according to robot used)
<b>!Time!</b>	Time (format set automatically according to robot used)
<b>!Date.Year!</b>	Year (4 digits)
<b>!Date.Month!</b>	Month (2 digits)
<b>!Date.Day!</b>	Day (2 digits)
<b>!Date.Hour!</b>	Hour (2 digits)
<b>!Date.Minute!</b>	Minutes (2 digits)
<b>!Date.Second!</b>	Seconds (2 digits)
<b>!LineNumber!</b>	Line number starting at index 1 (incremental counter +1)
<b>!LineNumberIndex0!</b>	Line number starting at index 0 (incremental counter +1)
<b>!Program.TotalPointNumber!</b>	Total number of toolpath points starting at index 1 (accessible from toolpath or main program)
<b>!Program.TotalPointNumberIndex0!</b>	Total number of toolpath points starting at index 0 (accessible from toolpath or main program)
<b>!NCProgram.TotalPointNumber!</b>	Total number of points written in the NCProgram starting at index 1 (accessible from main program only)
<b>!NCProgram.TotalPointNumberIndex0!</b>	Total number of points written in the NCProgram starting at index 0 (accessible from main program only)
<b>!NCProgram.Object.Type!</b>	Return the type of entity processed (“Transition”, “Toolpath”,...)

*Points definition variables (current point processed):*

<b>!X!</b>	X coordinate of the point
<b>!Y!</b>	Y coordinate of the point
<b>!Z!</b>	Z coordinate of the point
<b>!R1!</b>	First rotation (using robot convention)
<b>!R2!</b>	Second rotation (using robot convention)
<b>!R3!</b>	Third rotation (using robot convention)
<b>!Q1!</b>	First Quaternion (if robot convention uses quaternion)
<b>!Q2!</b>	Second Quaternion (if robot convention uses quaternion)
<b>!Q3!</b>	Third Quaternion (if robot convention uses quaternion)
<b>!Q4!</b>	Fourth Quaternion (if robot convention uses quaternion)
<b>!I!</b>	Tool axis vector I
<b>!J!</b>	Tool axis vector J
<b>!K!</b>	Tool axis vector K
<b>!U!</b>	Tool orientation vector U
<b>!V!</b>	Tool orientation vector V
<b>!W!</b>	Tool orientation vector W
<b>!A1!</b>	Robot axis 1 (*.mtd axis A)
<b>!A2!</b>	Robot axis 2 (*.mtd axis B)
<b>!A3!</b>	Robot axis 3 (*.mtd axis C)
<b>!A4!</b>	Robot axis 4 (*.mtd axis D)
<b>!A5!</b>	Robot axis 5 (*.mtd axis E)
<b>!A6!</b>	Robot axis 6 (*.mtd axis F)
<b>!E1!</b>	Robot external axis 1 (This number can differ from actual robot axis number)
<b>!E2!</b>	Robot external axis 2 (This number can differ from actual robot axis number)
<b>!E3!</b>	Robot external axis 3 (This number can differ from actual robot axis number)
<b>!E4!</b>	Robot external axis 4 (This number can differ from actual robot axis number)
<b>!E5!</b>	Robot external axis 5 (This number can differ from actual robot axis number)
<b>!E6!</b>	Robot external axis 6 (This number can differ from actual robot axis number)
<b>!E7!</b>	Robot external axis 7 (This number can differ from actual robot axis number)
<b>!E8!</b>	Robot external axis 8 (This number can differ from actual robot axis number)
<b>!E9!</b>	Robot external axis 9 (This number can differ from actual robot axis number)
<b>!E10!</b>	Robot external axis 10 (This number can differ from actual robot axis number)
<b>!E11!</b>	Robot external axis 11 (This number can differ from actual robot axis number)
<b>!E12!</b>	Robot external axis 12 (This number can differ from actual robot axis number)
<b>!Spindle.Speed!</b>	Current toolpath Spindle Speed (RPM)
<b>!Feedrate!</b>	Current toolpath feedrate (using robot feedrate unit)
<b>!Feedrate.FromSimulationPoint!</b>	Point Feedrate as initially defined into the simulation file (using robot feedrate unit)
<b>!MoveType!</b>	Move type of the point
<b>!CompType!</b>	Component type of the point
<b>!ContactNormalVector.I!</b>	Point contact normal vector I
<b>!ContactNormalVector.J!</b>	Point contact normal vector J
<b>!ContactNormalVector.K!</b>	Point contact normal vector K
<b>!CollisionState!</b>	Collision state of the point

Point number variables (current point processed):

<code>!Program.PointNumber!</code>	Point number starting at index 1
<code>!Program.PointNumber()!</code>	Point number starting at index 1 and formatted according to the split value (0000n)
<code>!Program.PointNumberIndex0!</code>	Point number starting at index 0
<code>!Program.PointNumberIndex0()!</code>	Point number starting at index 0 and formatted according to the split value (0000n)
<code>!Program.PreviousPointNumber!</code>	Point number starting at index 1 (minus 1)
<code>!Program.PreviousPointNumber()!</code>	Point number starting at index 1 and formatted according to the split value (0000n) (minus 1)
<code>!Program.PreviousPointNumberIndex0!</code>	Point number starting at index 0 (minus 1)
<code>!Program.PreviousPointNumberIndex0()!</code>	Point number starting at index 0 and formatted according to the split value (0000n) (minus 1)

Arc definition variables:

This table shows the list of variables available for the “Arc.Start” point. This list is also valid for the arc mid and arc end points, by replacing “Arc.Start” by “Arc.Mid” or “Arc.Start” by “Arc.End”, in the table below.

These variables are valid in any of the arc move definition blocks: ArcMoveStart, ArcMoveMid or ArcMoveEnd

The robot configurations variable are also accessible for the arc points. For example,

!Arc.Start.Configuration.Setting1! returns the configuration setting 1 of the “Arc.Start” point.

!Arc.Start.X!	X coordinate of the arc start point
!Arc.Start.Y!	Y coordinate of the arc start point
!Arc.Start.Z!	Z coordinate of the arc start point
!Arc.Start.R1!	First rotation (using robot convention)
!Arc.Start.R2!	Second rotation (using robot convention)
!Arc.Start.R3!	Third rotation (using robot convention)
!Arc.Start.Q1!	First Quaternion (if robot convention uses quaternion)
!Arc.Start.Q2!	Second Quaternion (if robot convention uses quaternion)
!Arc.Start.Q3!	Third Quaternion (if robot convention uses quaternion)
!Arc.Start.Q4!	Fourth Quaternion (if robot convention uses quaternion)
!Arc.Start.I!	Tool axis vector I
!Arc.Start.J!	Tool axis vector J
!Arc.Start.K!	Tool axis vector K
!Arc.Start.U!	Tool orientation vector U
!Arc.Start.V!	Tool orientation vector V
!Arc.Start.W!	Tool orientation vector W
!Arc.Start.A1!	Robot axis 1 (*.mtd axis A)
!Arc.Start.A2!	Robot axis 2 (*.mtd axis B)
!Arc.Start.A3!	Robot axis 3 (*.mtd axis C)
!Arc.Start.A4!	Robot axis 4 (*.mtd axis D)
!Arc.Start.A5!	Robot axis 5 (*.mtd axis E)
!Arc.Start.A6!	Robot axis 6 (*.mtd axis F)
!Arc.Start.E1!	Robot external axis 1 (This number can differ from actual robot axis number)
!Arc.Start.E2!	Robot external axis 2 (This number can differ from actual robot axis number)
!Arc.Start.E3!	Robot external axis 3 (This number can differ from actual robot axis number)
!Arc.Start.E4!	Robot external axis 4 (This number can differ from actual robot axis number)
!Arc.Start.E5!	Robot external axis 5 (This number can differ from actual robot axis number)
!Arc.Start.E6!	Robot external axis 6 (This number can differ from actual robot axis number)
!Arc.Start.E7!	Robot external axis 7 (This number can differ from actual robot axis number)
!Arc.Start.E8!	Robot external axis 8 (This number can differ from actual robot axis number)
!Arc.Start.E9!	Robot external axis 9 (This number can differ from actual robot axis number)
!Arc.Start.E10!	Robot external axis 10 (This number can differ from actual robot axis number)
!Arc.Start.E11!	Robot external axis 11 (This number can differ from actual robot axis number)
!Arc.Start.E12!	Robot external axis 12 (This number can differ from actual robot axis number)
!Arc.Start.Feedrate.FromSimulationPoint!	Point Feedrate as initially defined into the simulation file (using robot feedrate unit)
!Arc.Start.Program.PointNumber!	Point number starting at index 1
!Arc.Start.Program.PointNumber(!)	Point number starting at index 1 and formatted according to the split value (0000n)
!Arc.Start.Program.PointNumberIndex0!	Point number starting at index 0
!Arc.Start.Program.PointNumberIndex0(!)	Point number starting at index 0 and formatted according to the split value (0000n)

These variables are valid in the “ArcMoveEnd” definition blocks only:

<code>!Arc.Center.X!</code>	X coordinate of the arc center point
<code>!Arc.Center.Y!</code>	Y coordinate of the arc center point
<code>!Arc.Center.Z!</code>	Z coordinate of the arc center point
<code>!Arc.Center.X.Relative!</code>	X coordinate of the arc center point relative to the start
<code>!Arc.Center.Y.Relative!</code>	Y coordinate of the arc center point relative to the start
<code>!Arc.Center.Z.Relative!</code>	Z coordinate of the arc center point relative to the start
<code>!Arc.Radius!</code>	Arc radius
<code>!Arc.Plane!</code>	Arc plane (“XY”, “YZ” or “ZX”)
<code>!Arc.Direction!</code>	Arc direction (“CW” or “CCW”)

Robot configuration variables:

<p><b>!Configuration.Setting1!</b></p>	<p>ABB: CF1            FANUC: Flip/No Flip            KUKA: Status (as integer value)   Option="AA" for "Absolute Accuracy" robots            MOTOMAN: Type 1            STAUBLI: Elbow            HYUNDAI: Configuration in Hexadecimal (&amp;H0200)            KAWASAKI: Elbow            COMAU: Altitude and Turn            NACHI: Configuration (the 4 bits)            EPSON: Hand            MITSUBUSHI: FL1 (integer)</p>
<p><b>!Configuration.Setting2!</b></p>	<p>ABB: CF4            FANUC: Up/Down            KUKA: Status (as binary value)   Option="AA" for "Absolute Accuracy" robots            MOTOMAN: Type 2            STAUBLI: Shoulder            HYUNDAI: Configuration in Binary (00 0010 0000 0000)            KAWASAKI: Shoulder            COMAU: Altitude            EPSON: Elbow            MITSUBUSHI: FL1 (Hexadecimal)</p>
<p><b>!Configuration.Setting3!</b></p>	<p>ABB: CF6            FANUC: Toward/Backward            KUKA: Turn (as integer value)            MOTOMAN: Type 3            STAUBLI: Wrist            KAWASAKI: Wrist            COMAU: Turn            EPSON: Wrist            MITSUBUSHI: FL2 (integer)</p>
<p><b>!Configuration.Setting4!</b></p>	<p>ABB: CFX            FANUC: Conf J1            KUKA: Turn (as binary value)            MOTOMAN: Type 4            EPSON: J4Flag            MITSUBUSHI: FL2 (Hexadecimal)</p>
<p><b>!Configuration.Setting5!</b></p>	<p>FANUC: Conf J4            MOTOMAN: Type 5            EPSON: J6Flag</p>
<p><b>!Configuration.Setting6!</b></p>	<p>FANUC: Conf J6            MOTOMAN: Type 6</p>

"Include" variables:

<code>!Include_ToolDefinitions!</code>	Include the tool definitions (if enabled)
<code>!Include_FixtureOffsetDefinitions!</code>	Include the fixture offset definitions (if enabled)
<code>!Include_PointDefinition1Start!</code>	Include the point definitions 1 start (if enabled)
<code>!Include_PointDefinition1!</code>	Include the point & axis definitions 1 (if enabled)
<code>!Include_PointDefinition1End!</code>	Include the point definitions 1 end (if enabled)
<code>!Include_PointDefinition2Start!</code>	Include the point definitions 2 start (if enabled)
<code>!Include_PointDefinition2!</code>	Include the point & axis definitions 2 (if enabled)
<code>!Include_PointDefinition2End!</code>	Include the point definitions 2 end (if enabled)
<code>!Include_PointDefinition3Start!</code>	Include the point definitions 3 start (if enabled)
<code>!Include_PointDefinition3!</code>	Include the point & axis definitions 3 (if enabled)
<code>!Include_PointDefinition3End!</code>	Include the point definitions 3 end (if enabled)

PowerMill variables (valid when using PRIPost with PowerMill Robot only):

<code>!PowerMill.Version!</code>	PowerMill version number
<code>!PowerMill.Codebase!</code>	PowerMill codebase number
<code>!PowerMill.ProjectName!</code>	PowerMill project name
<code>!PowerMill.ProjectPath!</code>	PowerMill project path

PowerMill Robot variables (valid when using PRIPost with PowerMill Robot only):

<code>!Cell.Configuration.Name!</code>	Robot cell configuration name
<code>!Cell.Configuration.Name.NoSpace!</code>	Robot cell configuration name without space
<code>!Cell.Origin.Name!</code>	Robot cell origin name
<code>!Cell.Origin.Name.NoSpace!</code>	Robot cell origin name without space

Postprocessor variables:

<code>!CoordinateTransform.Enabled!</code>	State of the Coordinate Transform feature (returns 0 or 1)
<code>!Split.RetractDistance!</code>	Retract distance used when splitting programs

Other variables:

<code>!Space!</code>	Inserts a space character " "
<code>!Enter!</code>	Insert to new line
<code>!XML.CDATAStart!</code>	Replaces "<![CDATA[" (Compatibility issue with XML format)
<code>!XML.CDATAEnd!</code>	Replaces "]]>" (Compatibility issue with XML format)

# Autodesk Legal Notice

© 2020 Autodesk, Inc. All Rights Reserved. Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License that can be viewed online at <http://creativecommons.org/licenses/by-nc-sa/3.0/>. This license content, applicable as of 16 December 2014 to this software product, is reproduced here for offline users:

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

## License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

## 1. Definitions

a. "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(g) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

e. "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

f. "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

g. "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

h. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

i. "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

j. "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

**2. Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

**3. License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
- b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
- c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
- d. to Distribute and Publicly Perform Adaptations.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights described in Section 4(e).

**4. Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(d), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(d), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under: (i) the terms of this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-NonCommercial-ShareAlike 3.0 US) ("Applicable License"). You must include a copy of, or the URI, for Applicable License with every copy of each Adaptation You Distribute or Publicly Perform. You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License. You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

d. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and, (iv) consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(d) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,

iii. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c).

f. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

## **5. Representations, Warranties and Disclaimer**

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING AND TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THIS EXCLUSION MAY NOT APPLY TO YOU.

**6. Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## **7. Termination**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

## **8. Miscellaneous**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

### **Creative Commons Notice**

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at <http://creativecommons.org/>.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

### **Creative Commons FAQ**

Autodesk's Creative Commons FAQ can be viewed online at <https://knowledge.autodesk.com/customer-service/share-the-knowledge>, and is reproduced here for offline users.

Creative Commons is a simple, open licensing model which allows individuals to freely modify, remix, and share digital content created for learning and support.

Borrow from the Autodesk Learning, Support and Video libraries to build a new learning experience for anyone with any particular need or interest. It's out there. You can use it. It's yours.

In collaboration with Creative Commons, Autodesk invites you to share your knowledge with the rest of the world, inspiring others to learn, achieve goals, and ignite creativity.

### **What is Creative Commons?**

Creative Commons (CC) is a nonprofit organization that offers a simple licensing model that frees digital content to enable anyone to modify, remix, and share creative works.

### **How do I know if Autodesk learning content and Autodesk University content is available under Creative Commons?**

All Autodesk learning content and Autodesk University content released under Creative Commons is explicitly marked with a Creative Commons icon specifying what you can and cannot do. Always follow the terms of the stated license.

### **What Autodesk learning content is currently available under Creative Commons?**

Over time, Autodesk will release more and more learning content under the Creative Commons licenses.

Currently available learning content:

- Autodesk online help-Online help for many Autodesk products, including its embedded media such as images and help movies.
- Autodesk Learning Videos-A range of video-based learning content, including the video tutorials on the Autodesk YouTube™ Learning Channels and their associated iTunes® podcasts.
- Autodesk downloadable materials-Downloadable 3D assets, digital footage, and other files you can use to follow along on your own time.

## Is Autodesk learning and support content copyrighted?

Yes. Creative Commons licensing does not replace copyright. Copyright remains with Autodesk or its suppliers, as applicable. But it makes the terms of use much more flexible.

## What do the Autodesk Creative Commons licenses allow?

Autodesk makes some of its learning and support content available under two distinct Creative Commons licenses. The learning content is clearly marked with the applicable Creative Commons license. You must comply with the following conditions:

- **Attribution-NonCommercial-ShareAlike (CC BY-NC-SA)** This license lets you copy, distribute, display, remix, tweak, and build upon our work noncommercially, as long as you credit Autodesk and license your new creations under the identical terms. Terms of this license can be viewed online at <https://creativecommons.org/licenses/by-nc-sa/3.0/us/>
- **Attribution-NonCommercial-No Derivative Works (CC BY-NC-ND)** This license lets you copy, distribute, and display only verbatim copies of our work as long as you credit us, but you cannot alter the learning content in any way or use it commercially. Terms of this license can be viewed online at [https://creativecommons.org/licenses/by-nc-nd/3.0/us/deed.en\\_US](https://creativecommons.org/licenses/by-nc-nd/3.0/us/deed.en_US)
- **Special permissions on content marked as No Derivative Works** For video-based learning content marked as No Derivative Works (ND), Autodesk grants you special permission to make modifications but only for the purpose of translating the video content into another language.

These conditions can be modified only by explicit permission of Autodesk, Inc. Send requests for modifications outside of these license terms to [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

## Can I get special permission to do something different with the learning content?

Unless otherwise stated, our Creative Commons conditions can be modified only by explicit permission of Autodesk, Inc. If you have any questions or requests for modifications outside of these license terms, email us at [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

## How do I attribute Autodesk learning content?

You must explicitly credit Autodesk, Inc., as the original source of the materials. This is a standard requirement of the Attribution (BY) term in all Creative Commons licenses. In some cases, such as for the Autodesk video learning content, we specify exactly how we would like to be attributed.

This is usually described on the video's end-plate. For the most part providing the title of the work, the URL where the work is hosted, and a credit to Autodesk, Inc., is quite acceptable. Also, remember to keep intact any copyright notice associated with the work. This may sound like a lot of information, but there is flexibility in the way you present it.

Here are some examples:

"This document contains content adapted from the Autodesk® Maya® Help, available under a Creative Commons Attribution-NonCommercial-Share Alike license. Copyright © Autodesk, Inc."

"This is a Finnish translation of a video created by the Autodesk Maya Learning Channel @ [www.youtube.com/mayahowtos](http://www.youtube.com/mayahowtos). Copyright © Autodesk, Inc."

"Special thanks to the Autodesk® 3ds Max® Learning Channel @ [www.youtube.com/3dsmaxhowtos](http://www.youtube.com/3dsmaxhowtos). Copyright © Autodesk, Inc."

### **Do I follow YouTube's standard license or Autodesk's Creative Commons license?**

The videos of the Autodesk Learning Channels on YouTube are uploaded under YouTube's standard license policy. Nonetheless, these videos are released by Autodesk as Creative Commons Attribution-NonCommercial-No Derivative Works (CC BY-NC-ND) and are marked as such.

You are free to use our video learning content according to the Creative Commons license under which they are released.

### **Where can I easily download Autodesk learning videos?**

Most of the Autodesk Learning Channels have an associated iTunes podcast from where you can download the same videos and watch them offline. When translating Autodesk learning videos, we recommend downloading the videos from the iTunes podcasts.

### **Can I translate Autodesk learning videos?**

Yes. Even though our learning videos are licensed as No Derivative Works (ND), we grant everyone permission to translate the audio and subtitles into other languages. In fact, if you want to recapture the video tutorial as-is but show the user interface in another language, you are free to do so. Be sure to give proper attribution as indicated on the video's Creative Commons end-plate. This special permission only applies to translation projects. Requests for modifications outside of these license terms can be directed to [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

**How do I let others know that I have translated Autodesk learning content into another language?**

Autodesk is happy to see its learning content translated into as many different languages as possible. If you translate our videos or any of our learning content into other languages, let us know. We can help promote your contributions to our growing multilingual community. In fact, we encourage you to find creative ways to share our learning content with your friends, family, students, colleagues, and communities around the world. Contact us at [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

**I have translated Autodesk learning videos into other languages. Can I upload them to my own YouTube channel?**

Yes, please do and let us know where to find them so that we can help promote your contributions to our growing multilingual Autodesk community. Contact us at [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

**Can I repost or republish Autodesk learning content on my site or blog?**

Yes, you can make Autodesk learning material available on your site or blog as long as you follow the terms of the Creative Commons license under which the learning content is released. If you are simply referencing the learning content as-is, then we recommend that you link to it or embed it from where it is hosted by Autodesk. That way the content will always be fresh. If you have translated or remixed our learning content, then by all means you can host it yourself. Let us know about it, and we can help promote your contributions to our global learning community. Contact us at [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

**Can I show Autodesk learning content during my conference?**

Yes, as long as it's within the scope of a noncommercial event, and as long as you comply with the terms of the Creative Commons license outlined above. In particular, the videos must be shown unedited with the exception of modifications for the purpose of translation. If you wish to use Autodesk learning content in a commercial context, contact us with a request for permission at [creativecommons@autodesk.com](mailto:creativecommons@autodesk.com).

**Can I use Autodesk learning content in my classroom?**

Yes, as long as you comply with the terms of the Creative Commons license under which the learning material is released. Many teachers use Autodesk learning content to stimulate discussions with students or to complement course materials, and we encourage you to do so as well.

### **Can I re-edit and remix Autodesk video learning content?**

No, but for one exception. Our Creative Commons BY-NC-ND license clearly states that "derivative works" of any kind (edits, cuts, remixes, mashups, and so on) are not allowed without explicit permission from Autodesk. This is essential for preserving the integrity of our instructors' ideas. However, we do give you permission to modify our videos for the purpose of translating them into other languages.

### **Can I re-edit and remix Autodesk downloadable 3D assets and footage?**

Yes. The Autodesk Learning Channels on YouTube provide downloadable 3D assets, footage, and other files for you to follow along with the video tutorials on your own time. This downloadable material is made available under a Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license. You can download these materials and experiment with them, but your remixes must give us credit as the original source of the content and be shared under the identical license terms.

### **Can I use content from Autodesk online help to create new materials for a specific audience?**

Yes, if you want to help a specific audience learn how to optimize the use of their Autodesk software, there is no need to start from scratch. You can use, remix, or enrich the relevant help content and include it in your book, instructions, examples, or workflows you create, then Share-Alike with the community. Always be sure to comply with the terms of the Creative Commons license under which the learning content is released.

### **What are the best practices for marking content with Creative Commons Licenses?**

When reusing a CC-licensed work (by sharing the original or a derivative based on the original), it is important to keep intact any copyright notice associated with the work, including the Creative Commons license being used. Make sure you abide by the license conditions provided by the licensor, in this case Autodesk, Inc.

## Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, ADSK, Alias, ArtCAM, ATC, AutoCAD LT, AutoCAD, Autodesk, the Autodesk logo, Autodesk 123D, Autodesk Alias, Autodesk Forge, Autodesk Fusion, Autodesk Inventor, AutoSnap, BIM 360, Buzzsaw, CADmep, CAMduct, Civil 3D, Configurator 360, Dancing Baby (image), DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWGX, DXF, Eagle, ESTmep, FBX, FeatureCAM, Flame, FormIt 360, Fusion 360, The Future of Making Things, Glue, Green Building Studio, InfraWorks, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor HSM, Inventor LT, Make Anything, Maya, Maya LT, Moldflow, MotionBuilder, Mudbox, Navisworks, Netfabb, Opticore, PartMaker, Pier 9, PowerInspect, PowerMill, PowerShape, Publisher 360, RasterDWG, RealDWG, ReCap, ReCap 360, Remake, Revit LT, Revit, Scaleform, Shotgun, Showcase, Showcase 360, SketchBook, Softimage, Tinkercad, TrustedDWG, VRED.

All other brand names, product names or trademarks belong to their respective holders.

## Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.



*Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. Please see the Autodesk Creative Commons FAQ for more information.*