
Mech-Mind Software Release Notes

Mech-Mind

Dec 23, 2022

CONTENTS

1	Before You Upgrade	2
2	Mech-Vision 1.6 Release Notes	8
3	Mech-Viz 1.6 Release Notes	20
4	Mech-Center 1.6 Release Notes	26

This section introduces the new features, improvements, and resolved issues in Mech-Vision 1.6, Mech-Viz 1.6, and Mech-Center 1.6.

Attention: Before upgrading Mech-Vision, Mech-Viz and Mech-Center, please read this section carefully.

For the software upgrade notes, refer to *Before You Upgrade* .

For the new features, improvements, and resolved issues introduced in Mech-Vision 1.6, refer to *Mech-Vision 1.6 Release Notes* .

For the new features, improvements, and resolved issues introduced in Mech-Viz 1.6, refer to *Mech-Viz 1.6 Release Notes* .

For the new features, improvements, and resolved issues introduced in Mech-Center 1.6, refer to *Mech-Center 1.6 Release Notes* .

BEFORE YOU UPGRADE

1.1 1.6.2 Upgrade Notes

This section introduces the upgrade notes for Mech-Vision 1.6.2.

1.1.1 Mech-Vision Upgrade Notes

Notes for “Predict Pick Points” Steps

Mech-Vision 1.6.2 supports two “Predict Pick Points” Steps, which are “Predict Pick Points (Single Object Type)” and “Predict Pick Points (Any Objects)”. You do not need to install the Mech-Mind Software Environment to run these two Steps. However, you will need to use a deep learning server with a port number of 60000 or above.

If you are using Mech-Vision 1.5.3 or earlier versions, some deep learning Steps require that the Mech-Mind Software Environment or a special Mech-Vision version should be installed, and a deep learning server with a port number below 60000 should be used. Please refer to Compatibilities of Deep Learning Steps for detailed information.

In order to avoid environment conflict, it is prohibited to start a deep learning server with a port number below 60000 and another one with a port number of 60000 or above at the same time. Therefore, please pay attention to the following compatibility issues when you use these two Steps:

- When you create a new project and add the “Predict Pick Points” Step in the project, do not add other deep learning Steps that require the Mech-Mind Software Environment to be installed.
- If you have already opened a project containing the new “Predict Pick Points” Step in Mech-Vision, opening another project containing deep learning Steps that require the Mech-Mind Software Environment may cause environment conflict, and the projects cannot be run. To resolve the environment conflict, you can replace the deep learning Steps of earlier versions with the “Deep Learning Model Package Inference (Mech-DLK 2.2.0+)” or “Deep Learning Model Package CPU Inference” Steps. After replacing the Steps, you should train and import the deep learning model again.

1.2 1.6.1 Upgrade Notes

This section introduces the upgrade notes for Mech-Vision 1.6.1, Mech-Viz 1.6.1, and Mech-Center 1.6.1.

1.2.1 Mech-Vision Upgrade Notes

Notes for “Pick Anything” Projects

Currently, only Mech-Vision 1.6.1 special version supports the “Pick Anything” typical application projects. If you need to use “Pick Anything” projects, please contact Mech-Mind pre-sales engineers or sales to obtain the installation package of the special version.

When a “Pick Anything” project is run on the new version for the first time, the model will be converted. The higher the hardware configuration, the longer time the model conversion takes.

The new version is not compatible with “Pick Anything” projects created on an earlier version. If you want to use old “Pick Anything” projects on the new version, you need to delete and re-add the “Estimate Pick Points” Step.

V4 Cameras Should Be Used with Mech-Vision 1.6.0 or Later

Starting from 1.6.0, Mech-Vision is compatible with V4 cameras. Therefore, V4 cameras should be used together with Mech-Vision 1.6.0 or later.

Usage Notes for LSR V4/DEEP V4 Cameras

After upgrading Mech-Vision to 1.6.1, pay attention to the following notes when you are using LSR V4/DEEP V4 cameras:

- When opening old projects that use LSR V4/DEEP V4 cameras in Mech-Vision 1.6.1 for the first time, you need to reconnect the camera for Step “Capture Images from Camera” so as to make the “Camera Model” and “2D Image Type” parameters available.
- If the cameras used in existing projects have been upgraded to LSR V4/DEEP V4 cameras and these projects have used deep learning functions, you need to set the “2D Image Type” parameter to “Internal IR Image” for Step “Capture Images from Camera.” If you want to use “External Color Image” for image capturing instead, you need to retrain the models.
- If LSR V4/DEEP V4 cameras are used in new projects, especially if DEEP V4 cameras are used in depalletizing projects, you need to set the “2D Image Type” parameter to “External Color Image.” You can choose the 2D image type according to your project requirements in other situations.

Usage Notes for Step “3D Fine Matching”

- Mech-Vision 1.6.1 improved the standard deviation algorithm for Step “3D Fine Matching”, which makes the matching result filtering stricter. If you find that the detection rate decreases, you can manually set the “Standard Deviation” parameter to a lower value (0.003 m is recommended).
- It is not recommended to enable the **Speed Up on Large Object Quantities** option when only single object is in the scene. When this option is enabled, recognizing the single object will take longer.

Notes Related to Deep Learning

Before Mech-Vision 1.6.0, inference can be carried out only when a proper deep learning environment is installed. While Mech-Vision 1.6.0 and later have built-in inference models, inference can be carried out without installation of any deep learning environment.

On the industrial PCs or PCs without the deep learning environment, Mech-Vision 1.6.1 supports only the following deep learning Steps:

- Deep Learning Model Package Inference (DLK 2.2.0+): supports the models trained by Mech-DLK 2.2.0 or later
- Deep Learning Model Package CPU Inference: supports the models trained by Mech-DLK 2.2.1 or later

If an earlier project has already used deep learning Steps (other than above mentioned ones), you need to pay attention to the following notes to ensure that the earlier project can run normally after Mech-Vision is upgraded to 1.6.1:

- If the earlier project uses Mech-Vision 1.4.x or earlier, you **MUST** upgrade the deep learning environment to 2.0.0 or 2.1.0 (recommended) after Mech-Vision is upgraded to 1.6.1.
- If the earlier project uses Mech-Vision 1.5.x, keep deep learning environment at 2.0.0 or 2.1.0 after Mech-Vision is upgraded to 1.6.1.

Attention: Mech-Vision 1.6.0 and later no longer support the “Defect Detection” Step. This Step is available only in Mech-Vision 1.4.x with the deep learning environment 1.4.0 installed. If an earlier project still needs to use this Step, do not upgrade Mech-Vision.

Extra Python Library Should Be Installed under the “python” Directory

Mech-Vision 1.6.0 has integrated the Python 3.6.8 environment. If an extra Python library is required by the project, you must install it under the “python” directory of the Mech-Vision software.

1.2.2 Mech-Viz Upgrade Notes

Mech-Viz 1.6.1 has streamlined the built-in robot library. If a project uses the robot model that is not included in the built-in robot library, please download the robot model package from [Robot Model Package](#) and import it into the software, and then reload the project.

In Mech-Viz 1.6.1, the move and palletizing Tasks with the index will not automatically have the index reset when the index value exceeds the limit. Please check the project and connect "reset_task" in the proper part of the project to ensure the project can run as expected.

Mech-Viz 1.6.1 supports cylinder workobjects. If the vision result of the Mech-Vision project includes the object dimensions, you need to add the "object_type" label to the "Procedure Out" Step to notify Mech-Viz of the object type, either a cuboid, cylinder_vertical, or cylinder_horizontal.

In Mech-Viz 1.6.1, point cloud collision detection is not supported for picked cylinder workobjects. In addition, generation of object models based on received vision information and removing point cloud of picked object in the "visual_move" Task are not supported for cylinder workobjects to pick.

1.2.3 Mech-Center Upgrade Notes

Full-Control Program Reloading Required for ABB/FANUC/Kawasaki/KUKA

Mech-Center 1.6.1 has upgraded the full-control function. Therefore, after upgrading Mech-Center to 1.6.1, you need to reload the new full-control program for ABB/FANUC/Kawasaki/KUKA robots.

Attention: Before reloading the full-control program, please back up the current program files according to the instructions for each robot in full_control_program.

Usage Notes for New TM Full-Control Program

Mech-Center 1.6.1 provides a new full-control program for TM robots. The new full-control program supports controller 1.84 and higher.

If your project needs to use the new full-control program, upgrade your robot system to 1.84 or higher and reload the full-control program. If the robot in your project cannot be upgraded to the required version, please do not upgrade Mech-Center to 1.6.1.

Other Notes

Mech-Center 1.6.1 must be used with Mech-Viz 1.6.0 or later. A version mismatch may result in issues such as failing to obtain robot parameters.

If the standard interface command "Get DO List" has been used, you need to change the receiver of the "set_do_list" Task to "StandardInterface" in the Mech-Viz project.

Mech-Center 1.6.0 has built-in Python 3.6.5 environment. If extra Python libraries are required by an Adapter project, they must be installed under the "python" directory of Mech-Center.

Attention: After Mech-Center is upgraded to 1.6.1, Python libraries that have been installed on Mech-Vision 1.6.0 should be installed again.

1.3 1.6.0 Upgrade Notes

This section introduces the upgrade notes for Mech-Vision 1.6.0, Mech-Viz 1.6.0, and Mech-Center 1.6.0.

1.3.1 Mech-Vision Upgrade Notes

Before Mech-Vision 1.6.0, inference can be carried out only when a proper deep learning environment is installed. While Mech-Vision 1.6.0 has built-in inference models, inference can be carried out without installation of any deep learning environment.

On the industrial PCs or PCs without the deep learning environment, Mech-Vision 1.6.0 supports only the following deep learning Steps:

- Deep Learning Model Package Inference (DLK 2.2.0+): supports the models trained by Mech-DLK 2.2.0 or later
- Instance Segmentation (CPU): supports the models trained by Mech-DLK 2.2.0 or later

If an earlier project has already used deep learning Steps (other than above mentioned ones), you need to pay attention to the following notes to ensure that the earlier project can run normally after Mech-Vision is upgraded to 1.6.0:

- If the earlier project uses Mech-Vision 1.4.x or earlier, you MUST upgrade the deep learning environment to 2.0.0 or 2.1.0 (recommended) after Mech-Vision is upgraded to 1.6.0.
- If the earlier project uses Mech-Vision 1.5.x, keep deep learning environment at 2.0.0 or 2.1.0 after Mech-Vision is upgraded to 1.6.0.

Attention: Mech-Vision 1.6.0 no longer supports the “Defect Detection” Step. This Step is available only in Mech-Vision 1.4.x with the deep learning environment 1.4.0 installed. If an earlier project still needs to use this Step, do not upgrade Mech-Vision.

In addition, Mech-Vision 1.6.0 has integrated the Python 3.6.8 environment to support the “Calc Results by Python” Step. If the Python script called by this Step has used an extra Python library, you must install it under the “python” directory of the Mech-Vision software.

1.3.2 Mech-Viz Upgrade Notes

Mech-Viz 1.6.0 has streamlined the built-in robot library. If a project uses the robot model that is not included in the built-in robot library, please download the robot model package from [Robot Model Package](#) and import it into the software, and then reload the project.

Mech-Viz 1.6.0 does not support collision detection involving OBJ models composed of non-convex polyhedra. Collision detection of related models in earlier-version projects may fail after upgrade.

In Mech-Viz 1.6.0, the move and palletizing Tasks with the index will not automatically have the index reset when the index value exceeds the limit. Please check the project and connect "reset_task" in the proper part of the project to ensure the project can run as expected.

Mech-Viz 1.6.0 supports cylinder workobjects. If the vision result of the Mech-Vision project includes the object dimensions, you need to add the "object_type" label to the "Procedure Out" Step to notify Mech-Viz of the object type, either a cuboid, cylinder_vertical, or cylinder_horizontal.

In Mech-Viz 1.6.0, point cloud collision detection is not supported for picked cylinder workobjects. In addition, generation of object models based on received vision information and removing point cloud of picked object in the "visual_move" Task are not supported for cylinder workobjects to pick.

1.3.3 Mech-Center Upgrade Notes

Mech-Center 1.6.0 must be used with Mech-Viz 1.6.0 or later. A version mismatch may result in issues such as failing to obtain robot parameters.

If the standard interface command "Get DO List" has been used, you need to change the receiver of the "set_do_list" Task to "StandardInterface" in the Mech-Viz project.

Mech-Center 1.6.0 has built-in Python 3.6.5 environment. If extra Python libraries are required by an Adapter project, they must be installed under the "python" directory of Mech-Center.

In addition, Mech-Center 1.6.0 has changed the usage of some standard interface commands. For details, refer to [Mech-Center 1.6.0 Release Notes](#).

MECH-VISION 1.6 RELEASE NOTES

This chapter introduces the new features, improvements, and resolved issues of Mech-Vision 1.6.

2.1 Mech-Vision 1.6.2 Release Note

2.1.1 New Feature

New Step - Predict Pick Points (Single Object Type)

A new Predict Pick Points (Single Object Type) Step was added in Mech-Vision 1.6.2, which is used to detect pickable objects in 2D images and depth maps and then output corresponding pick points. This Step is usually used for picking randomly stacked objects of the same type.

You do not need to install the Mech-Mind Software Environment to run this Step. You should use a port number of 60000 or above and import a deep learning model. Different types of objects require different models. Please contact Mech-Mind Technical Support for the model.

Please refer to *Notes for "Predict Pick Points" Steps* for detailed upgrade notes.

2.1.2 Improvements

"Pick Anything" Typical Application Projects Improved

The Pick Anything Typical Application projects include Pick Anything and Pick Anything (Without Bin). In Mech-Vision 1.6.2, you can use the improved "Pick Anything" typical application projects and do not need to download a special version.

Updated Step Name

Before Mech-Vision 1.6.2	In Mech-Vision 1.6.2
Grasp Pose Estimation	Predict Pick Points (Any Objects)

2.1.3 Resolved Issue**Issue with “Capture Images from Camera” Step**

- Resolved the issue that the setting of play back mode could not take effect for the Virtual Mode in the Capture Images from Camera Step.

2.2 Mech-Vision 1.6.1 Release Notes**2.2.1 New Features****Added Save All Option to File Menu**

A **Save All** option was added to the **File** menu in the menu bar of Mech-Vision 1.6.1. You can use it to save all the opened projects in one click.

New Step - Convert Data Type

The newly added `convert_data_type` Step can be used to convert one data type to another.

The currently supported data types include BoolList, DoubleList, String, StringList, Variant, VariantList, etc.

New Step - Convert Lengths Pixel-Wise to Physical

The newly added `convert_lengths_pixel_wise_to_physical` Step can be used to calculate the actual length of a specified line segment in 2D images. This Step is applicable to scenarios where 2D images are used for measuring dimensions of objects with relatively planar surfaces.

New Step - Detect and Measure Oblong Hole

The newly added `detect_and_measure_oblong_hole` Step can be used to detect the pixel-wise positions and sizes of oblong holes in images to facilitate subsequent calculation of physical dimensions in measurement scenarios.

New Step - Deep Learning Model Package CPU Inference

The newly added `deep_learning_model_package_cpu_inference` Step can be used for classification, instance segmentation, and object detection.

- You can edit the ROI directly in this Step without depending on `scale_image_in_2d_roi` and `recover_scaled_images_in_2d_roi` Steps.
- This Step is only compatible with DLKPACKC models exported by Mech-DLK 2.2.1 or higher versions.
- It is recommended to use this Step for model inference when the requirement on inference speed is not high, and it is also recommended to deploy CPU models on computers with 12th Gen Intel Core i5 processors or above.

Added Features in Deep Learning Model Package Inference (DLK 2.2.0+)

- Added the **ROI Settings** parameter for editing the ROI.
- Added the **Font Settings** parameter for customizing font size in the visualized output.
- When this Step is used with instance segmentation models, results below the confidence threshold are also displayed in the visualized output. The results above the threshold will be displayed in green and the results below the threshold will be displayed in red.
- Added support for non-Latin characters in the model package path.

Added Rectify to Depth Map Parameter in Capture Images from Camera

The **Camera Model** and **Rectify to Depth Map** parameters were added in the `capture_images_from_camera` Step to make the pixels of the color image and depth map have one-to-one correspondences when cameras of DEEP V4 series and LSR V4 series are used.

2.2.2 Improvements

Example Projects Improved

Added a **Brake Discs** machine tending project in example projects.

Typical Applications - Pick Anything Projects Improved

The **Piece Picking** project is renamed as **Pick Anything** project in typical application projects.

The **Pick Anything** typical application projects are divided into **Pick Anything** projects and **Pick Anything (Without Bin)** projects. You do not need to deploy a deep learning model file any more. The project can be run after you calibrate the camera and configure camera parameters.

In Mech-Vision 1.6.1, the `grasp_pose_estimation` Step used in the project was updated as well.

Attention:

- You have to use a special version of Mech-Vision 1.6.1 to open the Pick Anything typical application project, and the Piece Picking typical projects of older versions cannot be executed in this special version software. Please contact Mech-Mind Technical Support to obtain the installation package of the special version of Mech-Vision 1.6.1.
- Please use a server port number of 5000 or below while running the Pick Anything typical application project.
- The new Pick Anything typical application projects do not support preload models.

Typical Applications - Large Non-Planar Workpieces Projects Improved

Deployment guidance for large non-planar workpieces machine tending projects was added. The deployment guidance includes camera configuration, recognition, and deployment, which can help users to construct the project more conveniently.

Matching Model and Pick Point Editor Improved

The improvements of the `matching_model_and_pick_point_editor` are as follows:

- Added **Show normals** option to show normals of the point cloud.
- Added an eye icon in the upper right corner of the Model files area for displaying/hiding all point cloud models and pick points.
- Added a **Pose manipulator settings** button in the lower left corner of the interface for configuring the display settings of the pose manipulator.

3D Fine Matching Step Improved

The improvements of the `3d_fine_matching` Step are as follows:

- Added the **Speed Up on Large Object Quantities** option in **Correspondence Settings**. It is recommended to enable this option when there are a large number of objects in the scene.
- Added the **Pose Filtering Settings** parameter for filtering overlapping objects.
- Deleted the **Minimum Standard Deviation** parameter.

Template Matching Step Improved

The improvements of the `template_matching` Step are as follows:

- Added the **ROI** parameter.
- Added the **Tile Gray Scale Upper Threshold** parameter.

If you cannot find the above parameters, please right-click on the blank area in the **Step Parameters** panel and select **Show all parameters** in the context menu.

Blob Analysis Step Improved

The algorithm processing speed of this Step was improved.

Added Convert Types in the Convert Color Space Step

The three newly added convert types in the `convert_color_space` Step are as follows:

- RGB to HSI
- RGB to HSV
- RGB to YUV

Measurement Mode Improved

The improvements of the measurement mode are as follows:

- The pixel coordinates of the cursor's position is now displayed in the sketchpad.
- Added *Show coordinates* option in the sketchpad settings for displaying/hiding the coordinates in the sketchpad.

Output Results from Check Pose Repeatability by Statistics Improved

- When the input type is **PoseListInput**, the default unit of the output statistics is mm, and the numbers will be rounded up to 3 decimal places to meet the requirement for high-precision repetitive positional statistics.
- When the input type is **PoseListInput**, the value of the **Acceptable Position Coordinate Deviation** parameter is allowed to be less than 1 mm to meet the requirement for high-precision repetitive positional statistics.
- The upper limit of **Threshold for Acceptable Position Coordinate Deviations** was raised to 1000mm, and the upper limit of **Threshold for Acceptable Euler Angle Deviations** was raised to 360°.

Safety Alert Improved

When virtual data is used in a Mech-Vision project, warning alerts will pop up in Mech-Center and Mech-Viz. You will need to select a proper option in the pop-up windows to ensure production safety.

Support DEEP V4 and LSR V4 series in Depalletizing Typical Application Projects

In Mech-Vision 1.6.1, the DEEP V4 and LSR V4 series of Mech-Eye Industrial 3D Camera are supported in depalletizing typical application projects.

Step-related Features Improved

- Added icons for more Steps.
- Added parameter tooltips for more Steps.

Pop-Up Windows Improved

When there are multiple missing Steps in the project, all of these Steps will be listed in one pop-up window instead of multiple separate pop-up windows.

2.2.3 Resolved Issues

3D Fine Matching Step

- Resolved the issue that the data of color point cloud could not be loaded properly.
- Resolved the issue that small objects might be incorrectly matched.

Camera Calibration

- Resolved the issue that the Save button was unavailable for calibration on 4-axis and 5-axis robots.
- Resolved the issue that the calculation result could still be incorrect when a wrong camera had been disconnected and a right one was reconnected properly in ETE calibration mode.
- Resolved the issue that 2D camera could not be calibrated properly.

2.3 Mech-Vision 1.6.0 Release Notes

2.3.1 New Features

Data Storage Restructured

The `data_storage` feature is restructured for better performance in troubleshooting and regression testing on-site. The detailed changes are as follows:

- **Data Storage** features no longer require the Procedure **Save Images and Step Parameters** to be used.
- Abnormal data are now stored in the new **error_data** folder in the **data** folder.
- Data from multiple cameras can be saved altogether.

- The naming conventions of the file and folder names in the **data** folder are changed.

Added Example Projects

Example Projects of typical applications in various industries are introduced in Mech-Vision V1.6.0 to provide a general picture of what Mech-Vision can do and how Steps work. These projects are highly user-friendly for beginners as they can be run with one click.

New 2D Matching Steps

The following 2D Matching Steps are added:

- make_template
- template_matching
- record_criterion_pose_and_calc_transformation
- transform_image

These Steps can be used to obtain the positions of target objects in an image and align the target objects/ROIs in different images to the same position/orientation.

New Step - Deep Learning Model Package Inference

Mech-Vision V1.6.0 provides built-in deep learning inference models. With the `deep_learning_model_package_inference` Step, deep learning inference can be enabled without installing Mech-Mind Software Environment.

New Step - 3D Coarse Matching V2

A newer version of the 3D coarse matching Step, `3d_coarse_matching_v2`, is added. This Step is used to coarsely match the point cloud model with the original point cloud and output the coarsely calculated candidate poses of the target objects in the scene.

New Step - Calc Results by Python

Mech-Vision V1.6.0 now has the Python 3.6.8 environment integrated, which allows you to run a Python script directly in Mech-Vision with the **Calc Results by Python** Step.

Note: If extra Python libraries are required when you use this Step, these libraries must be installed to the *python* folder under the installation directory of Mech-Vision.

Tip: A Python library can be installed as follows:

1. Open the **Command Prompt** or **PowerShell** program.
2. Switch to the "python" directory of Mech-Vision, such as `C:\Mech-Mind\Mech-Vision-1.6.x\python`.

3. Execute the following command: `./python -m pip install library_name`.
-

Debug Output Improved

The function of the **Debug Output** panel is improved to better facilitate you in building and analyzing your Mech-Vision project.

Improvements include:

- When **Debug Output** mode is disabled, output results (shown by running a Step with **Visualize Output** or by clicking on the connection between two Steps) are also displayed in the **Debug Output** panel.
- Display output in a separate window: click on the icon in the upper right of the **Debug Output** panel to open a separate maximized window.

When **Debug Output** mode is enabled, the separate window also allows for the following usage:

- With a separate window opened, the result of the same Step is updated every time you run the Step/project and displayed in the same window.
- Viewpoint adjustment of a 3D result in a separate window is memorized, and when you run the Step/project again, the result of the same Step is displayed with the same viewpoint, so that you can compare results from multiple executions easily.
- You can open the results from multiple Steps in separate windows simultaneously for easy comparison.

Matching Model and Pick Point Editor Improved

- Add point cloud from camera:
 - Adaptation for applications with target objects that are heavy and/or not easily movable: now you can directly generate a point cloud of the entire scene and then edit the point cloud, separate depth map acquisition of the background and target object not mandatory.
 - When acquiring depth maps of the background and target object separately, background depth map is acquired first to better fit actual usage.
- New features for point cloud editing:
 - **Remove selected points** button added
 - Invert selection: select the points you want to keep first, and then invert the selection to quickly delete the unwanted points.
 - Generate edge point cloud: you can now generate a point cloud model of the object edges in the editor.

2.3.2 Improvements

User Interface Improved

The UI of Mech-Vision V1.6.0 is improved in the following aspects for better user experience:

- Layout and color scheme redesigned
- **Default Layout** option added to the **View** menu: automatically adjust the interface layout with a single click
- Appearance and user interaction of the **Step Parameters** tab improved
- **Save**, **Undo**, and **Redo** buttons added to the project toolbar at the top of the graphical programming workspace
- Appearance of the search box in **Step Library** improved
- Interface of **Camera Viewer** improved
- Layout and text in **Camera Calibration** improved

Edit Procedure Parameter Feature Improved

In Mech-Vision V1.6.0, the **Edit Procedure Parameter** feature is improved in the following aspects:

- Added Parameter Info Customization feature that enables users to customize the display name, value tooltip, and key tooltip of Procedures
- Added custom_mapped_parameter for customizing the mapping relationship in a Procedure with JavaScript code
- Added custom_recipe_parameter for setting the selected parameters as a recipe parameter of a Procedure

Model Selection Parameter Added to the 3D Coarse Matching and 3D Fine Matching Steps

In Mech-Vision V1.6.0, the 3d_coarse_matching and 3d_fine_matching Steps now have a **Model Settings** parameter category for easy model selection from the model gallery.

The Check Pose Repeatability by Statistics Step Improved

The **Check Pose Repeatability by Statistics** Step is improved in the following aspects for better usability:

- Name updated: from **Poses Repeatability Statistics** to **Check Pose Repeatability by Statistics**
- **Threshold for Acceptable Position Coordinate Deviations** and **Threshold for Acceptable Euler Angle Deviations** parameters added for filtering out errors
- Output orientation data now in Euler angles
- Number of decimal places of the output data customizable
- Layout of output XLSX table improved

- Inputting multiple pose data simultaneously and saving results of different poses to different table sheets now supported
- **Restart** option now automatically unchecked after the Step is run once with the option checked
- Step Quick Info and Parameter tooltips updated

Compatibility for V4 and UHP Cameras

Mech-Vision V1.6.0 now supports image capturing and hand-eye calibration with V4 and UHP cameras.

Performance of the Instance Segmentation Step Improved

In Mech-Vision V1.6.0, instance segmentation conducted with the new `deep_learning_model_package_inference` Step has better performance compared to the old **Instance Segmentation** Step.

Updated Step Names

The names of the following Steps are updated for easier understanding of the functions. When you open a project in Mech-Vision V1.6.0, the name of the Steps in your project are automatically updated.

Before V1.6.0	V1.6.0
Map to Multi Pick Points	Map to Multiple Pick Points
Calc Included Angle Between Specified Axis of Poses	Calc Included Angles between Specified Axes of Poses
Validate Box Masks	Validate Box Object Masks
Segment Depth Image	Segment Depth Map
Get Highest Areas in Depth Image	Get Highest Layer Regions in Depth Map
Pose Adjustment Collection	Adjust Poses
Rotate Poses to Goal Direction	Rotate Poses' Axes to Specified Directions
Classification by Point Clouds' Sizes	Classify Point Clouds by Dimensions
Sort and Output Index List	Sort List and Output Index List
Grasp Pose Estimation	Estimate Pick Points
Set Pose Quaternion	Set Pose Quaternions
Inverse Poses	Invert Poses
Validate Poses by Included Angle to Reference Direction	Validate Poses by Included Angles to Reference Direction
Binarize Image	Image Thresholding
CloudXYZ To CloudNormal	From CloudXYZ to CloudNormal
Cloud Num Limit	Trim Point Cloud List
From Poses to Euler Angles	Convert Quaternions to Euler Angles in Poses
Merge Cloud Vector	Merge Point Cloud Lists
Find 2D Contour at Specified Hierarchical Level	Find 2D Contour at Specified Inner-Outer Level
Calc Length Along Axis	Calc Point Cloud Spans along Axes
Poses Repeatability Statistics	Check Pose Repeatability by Statistics
Calc Minimum Bounding Rectangles of Masks	Calc Minimum Circumscribed Rectangles of Masks
Calc Distance to Reference Pose	Calc Distances between Poses
Smooth Trajectory	Smooth Path
Make Poses Point to Reference Place	Point Poses to Reference Positions

continue

Table 1 – continued from previous page

From Disparity Image to Depth Image	Convert Disparity Image to Depth Map
Compose Pose From Quaternion and Translation	Compose Poses from Quaternions and Translation Vectors
Merge LineSegment Vector	Merge LineSegment Lists
Merge Depth Images	Merge Depth Maps
Calc Angle Between Vector3D	Calc Angles between Vector3Ds
Mask Cluster	Mask Clustering
Cloud Smooth And Normal Estimation	Smooth Point Cloud and Estimate Normals
Compose Vector3D From Numbers	Compose Vector3Ds from Numbers
Compare Two Depth Image	Compare Two Depth Maps
Generate Point Cloud of Ring	Generate Ring Point Cloud
Adjust Poses to Obtain Accurate Trajectory	Adjust Targets to Get Correct Path
Load Poses in Trajectory and Apply Affine Transform	Load Targets in Path and Apply Affine Transform
Calc Poses from Heat Map of Graspability	Calc Poses from Heat Map of Pickability
Reverse	Reverse List
Cloud Filter By Model And Pose	Filter Point Cloud by Model and Poses
Merge Point Clouds with Similar Height	Merge Point Clouds with Similar Heights
Calc Mask's Span on Given Line	Calc Mask Spans on Given Lines
Save Local Areas Around Poses as 3D ROI	Save Regions around Poses as 3D ROIs
Adjust Trajectory Circular Motion	Change Circular Motion Direction of Path
Detect Bin (Inscribed Rect Sides)	Detect Bin (Max Inscribed Rect)
Detect Bin (Largest Inscribed Rect)	Detect Bin (Max Inscribed Rect) V2
Filter Poses Outside Bin	Remove Poses outside Bin
Pixel-wise Graspability Evaluation	Pixel-Wise Pickability Evaluation
Mask Filter	Filter Masks
2D Poses To 3D Poses Base Orthographic Projection	Convert Poses 2D to 3D According to Orthographic Projection
Get Valid Ring Clouds	Filter Ring Point Cloud List
Project A Point onto A Plane	Project Points onto Plane
Compose Quaternion From Two Axis (Right-Hand)	Compose Quaternions from Two Axes (Right Hand Rule)
Calc Diagonal Length	Calc Diagonal Lengths
Generate Rect Traj	Generate Rect Path
Generate Traj By Contour	Generate Path from Contour
Is Z Value of Input Greater than Threshold	Compare Z Values of Poses with Threshold
Rectify Ring Pose	Rectify Ring Object Poses
Calc Center Point of Non-zero Areas	Calc Center Points of Non-Zero Regions
Calc Rect 2D Pose	Calc 2D Poses of Rectangles
Calc Pixel Size at Specified Height	Calc Pixel Sizes at Specified Heights
Coherent Line Drawing	Draw Coherent Lines
Image Transform	Adjust Image
Determine Pixel Size	Determine Pixel Sizes
Extract Empty Areas in Depth Image within 3D ROI	Extract Empty Regions in Depth Map within 3D ROI
Rotate Images By Provided Poses	Rotate Images by Specified Poses
Perspective Transform	Perspective Transformation
Clouds In 3d Box	Extract Point Cloud in 3D Box
Move Cloud Along Set Dir	Move Point Cloud along Specified Direction
Cloud Distortion Correction	Correct Point Cloud Distortion
Detect Obscured Objects	Detect Occluded Objects
Transform Plane Cloud To Align Direction	Align Plane Point Clouds
Cloud Scale	Scale Point Cloud

continue

Table 1 – continued from previous page

Calc Point Cloud Curvature	Calc Point Cloud Curvatures
Calc Edge Points Normal	Calc Edge Point Cloud Normals
Pose Transformed by Quaternion in Object Coordinate	Rotate Poses by Quaternion Vectors in Object Frames
Calc Length of Vector3D	Calc Lengths of Vector3Ds
Pose Transformed by Pose2 in Object Coordinate	Transform Poses by Matrix in Object Frames
Calc Cross Product of Vector3D	Calc Vector3D Cross Products
Calc Normalized Vector3D	Calc Normalized Vector3Ds
Compose Quaternion From Axis and Angle	Compose Quaternions from Axes and Angles
Rotate Poses to Directions With Symmetry Constraint	Rotate Poses' Axes to Specified Directions under Symmetry
Easy Point to Reference Place	Easy Point Poses to Reference Position
Easy Coordinate Transform	Easy Frame Transformation
Inverse Quaternions	Get Inverses of Quaternions
Calc Dot Product of Vector3D	Calc Vector3D Dot Products
Calc Distance from 3D Points to Plane	Calc Distances from 3D Points to Plane
Measure Circle	Measure Circles
Calc Distance from 3D Points to Intersection of Two Planes	Calc Distances from 3D Points to Intersection of Two Planes
Replace Element In Vector	Replace Elements in List
Generate Test Cloud	Generate Test Point Cloud
Save Trajectory Points	Save Path Targets
Evaluate Variation of Depth Image	Evaluate Depth Map Fluctuation
Calculate Calib-board Pose	Calc Calibration Board Poses
Insert End Points And Send Motion Params	Insert End Target and Send Motion Params
Load 2d Trajectory	Load 2D Path
Generate Spiral Traj	Generate Spiral Path
Generate Trajectory Given Depth Image	Generate Path Given Depth Map
Generate Zigzag Traj	Generate Zigzag Path
Auto Trajectory	Extract 2D Path
Merge Label List	Merge Label Lists
Adjust Poses by Obstacles	Adjust Targets by Obstacles
Trajectory Points Matching	Path Target Matching
Adjust Poses by Obstacles V2	Adjust Targets by Obstacles V2
Validate 2D Poses Within Mask	Validate 2D Poses by Mask
Calc Mask Distance	Calc Mask Distances
Varying Normal Area	Extract Regions of Large Normal Deviations
Map Mask Non-zero Area	Extract Image Regions by Mask
Detect Graspable Rectangles	Detect Pickable Rectangles
Depth Cluster along Scan Lines	Depth Clustering along Scan Lines
Divide Cloud into Smaller Parts Evenly	Divide Point Cloud into Smaller Parts Evenly
Deep Learning Inference	Deep Learning Inference (DLK 2.1.0/2.0.0)

MECH-VIZ 1.6 RELEASE NOTES

This chapter introduces the new features, improvements, and resolved issues of Mech-Viz 1.6.

3.1 Mech-Viz 1.6.2 Release Notes

3.1.1 Improvements

Model Editor Supporting End Effector Models in STP and STEP Formats

In Mech-Viz 1.6.2, Model Editor starts to support end effector models in STP and STEP formats. You can directly import an end effector model in STP or STEP format without converting formats by using SolidWorks or other software.

3.1.2 Resolved Issue

- Resolved the issue that the software crashed occasionally.

3.2 Mech-Viz 1.6.1 Release Notes

3.2.1 Improvements

Support OBJ Models from Sources Besides the Model Editor

In Mech-Viz 1.6.0, all OBJ models must be generated from the built-in Model Editor, and it is not user-friendly when OBJ models from other sources are used. In Mech-Viz 1.6.1, there are no restrictions on the source of the OBJ models. When you add a collision model of the end effector, Mech-Viz can convert the OBJ model to one that is entirely composed of convex polyhedra automatically.

Hint:

- OBJ models already entirely composed of convex polyhedra are loaded directly without conversion.

- As for OBJ models not entirely composed of convex polyhedra, their shapes may change after the one-click conversion. In this case, it is recommended to use Blender or Model Editor to edit these OBJ models before loading to ensure that they can meet the requirement.
 - As for models in projects of previous versions, a message asking if you would like to perform one-click conversion will pop up when you open the project in Mech-Viz 1.6.1. If you confirm to convert the model, please check if the shape of the converted model is changed or not.
 - The one-click conversion function currently does not support display models of end effectors, display models of scene objects, and collision models of scene objects.
-

Safety Alert of Virtual Camera

When Mech-Viz is used to guide a real robot and a “virtual camera” is used in Mech-Vision at the same time, a warning alert will pop up in order to avoid production accidents.

3.2.2 Resolved Issues

Mech-Viz 1.6.1 has fixed the following issues:

- The ground style would be reset to Chessboard after the project was re-opened.
- Certain robots could not be displayed in the Robot Library.
- The original collision model remained displayed after switching the collision model of the end effector.
- The collision model might be displayed in the wrong way when a collision was detected after switching the end effector.
- Two collision models were highlighted at the same time when a collision was detected.
- The ASCII STL model could not be loaded as an end effector model.
- The DAE model could not be loaded as an end effector display model.
- The TCP of the default display robot was not reset to the original position after the project was closed.
- The end effector models was not cleared after the project was closed.
- The ASCII STL scene models could not be converted to binary and saved.
- The collided picked workpiece could not be recorded in the plan history.
- Collision could not be displayed during the planning process.
- The software might crash when a deleted Task was searched.
- The “dynamic_move” Task could not adjust the pose properly according to the result from the “visual_move” Task.
- There might be missed detection when the **Pcl Collision Check Mode** in the “relative_move” Task was set to **Auto**.
- An error might occur when the DI port number was large in the “check_di” Task.

- The classification would fail and the project would stop when the “classify” Task preceded the “visual_move” Task.
- The service could not be found when “custom_pallet_pattern” or “predefined_pallet_pattern” Task was used as the visual service.
- An error might occur in collision detection after the TCP direction had been modified in vacuum gripper configurator of the “visual_move” Task.
- The vacuum gripper offset calculation might be incorrect when **Picking Setting** was set to **Pick Single Carton** in the “visual_move” Task.
- The scene object controlled by the “update_scene_object” Task might not be selectable.
- The workpiece model might not be displayed when the **Property Value** was selected as the **Update Info Source** and the **Size Needs Updating** checkbox was not selected in the “update_picked_obj” Task.
- The size of the workpiece might be incorrect when the **Property Value** was selected as the **Update Info Source** and the **Size Needs Updating** checkbox was not selected in the “update_picked_obj” Task.
- The collision model of the workpiece might be incorrectly displayed after the “update_picked_obj” Task was executed.
- The vacuum gripper offset information printed in the log was incorrect.

3.3 Mech-Viz V1.6.0 Release Notes

3.3.1 New Features

Newly Added Model Editor

A built-in model_editor is added in Mech-Viz V1.6.0 to facilitate users to simplify end effector models and scene models and obtain collision models in OBJ format.

Newly Added Cylindrical Scene Models and Workobjects

Besides the existing box and cuboid models, cylinder models are introduced in Mech-Viz V1.6.0 to make it easier to build the scene. Please refer to :ref: create_load_model for detailed information.

Cylindrical workobject models can only be obtained via the vision results sent by Mech-Vision or generated by the **update_picked_obj** Task. In addition, in Mech-Viz V1.6.0, only the collision between cylindrical workobjects and scene objects can be detected; detection of the collision between cylindrical workobjects and point cloud is currently not supported.

Updates on Tasks

- General updates

Checkboxes are introduced for enabling/disabling parameters, replacing the previous True/False drop-down options.

A new **Task ID** parameter is added to all Tasks. The Tasks will be numbered automatically with unique Task IDs. Please note that Task IDs cannot be modified.

- Updates on move-related Tasks

A new **Send Target** parameter is added to all move-related Tasks. This parameter is enabled by default, suggesting that the targets will be sent to the receiving end (e.g., a robot). You can disable this parameter if you only need the Task to plan a path but not send targets.

A new **Intelligent Obstacle Avoidance** feature is added to the move-type Task. This feature provides multiple candidate targets that are close to the target of the Task, which increases the feasible paths planned by Mech-Viz and the success rate of planning.

- Updates on palletizing-related Tasks

A new **multi_pick_palletizing** Task is added. This task can compute a path for multiple-box palletizing according to the positions of the boxes to be picked and the target pallet pattern to increase the efficiency of palletizing.

A new **Send Placing Pose Only** parameter is added, which is disabled by default. Enabling this parameter will send the final placing pose to the receiving end (e.g., a robot).

A new **Match Symmetry** parameter is added in the **visual_pallet_pattern** Task. This parameter is used to solve the problem that the previous pallet pattern cannot match with the newly specified one since the orientations of the previously palletized cartons do not comply with the orientations of cartons in the newly specified pallet pattern.

- Renamed Tasks

Before V1.6.0	V1.6.0
branch_by_service_message	branch_by_msg
branch_by_tag	branch_by_guidepost
set_tag	set_guidepost

Other Updates

Added **Default Layout** in *Menu Bar* → *View* for resetting the layout of Mech-Viz to default.

Added the option **Trajectory Dragger Size** in *Menu Bar* → *Display* → *Display Settings* for adjusting the display size of the trajectory dragger.

3.3.2 Improvements

Robot Model Package

In previous versions of Mech-Viz, all of the supported robot model packages are integrated into the software, which occupies a large portion of the disk space of the computer.

From Mech-Viz V1.6.0, only the model packages of the widely used robots (about 100 types in total) are integrated in the software.

If you need to use robot models that are not included in the robot library of Mech-Viz, please download it from [Robot Model Package List](#) and then import it into Mech-Viz.

Decouple DI from Blocks in Vacuum Gripper Configurator

In Mech-Viz V1.6.0, DI is not dependent on block configuration of the vacuum gripper anymore. Only the DI covered by the carton will be included in the carton dropping detection.

Support TCP Rotation Adjustment in Vacuum Gripper Configurator and Array Gripper Configurator

You can now configure the TCP rotation in Vacuum Gripper Configurator and Array Gripper Configurator in Mech-Viz V1.6.0.

Support for Displaying and Adjusting the End Effector Sequence Number

Sequence number of end effectors are added in the **End Effector Configuration** panel and displayed in the drop-down list of **Set End Effector** in the **tcp** Task. You can drag the end effector in the **End Effector Configuration** panel to adjust its sequence number.

Add Euler Angles Option in Edit Pose Window

Poses can now be edited in Euler angles in the **Edit pose** window. You can now copy and paste the TCP sent by Mech-Vision or Mech-Center directly into the **Edit pose** window.

Singularity Detection Mode

A new singularity detection mode **Detect joint angle of the robot** is added. You can set the joint and lower/upper bound of the angle to filter the planning result that may lead to singularities.

Add Fixed Edge Corner Offset Parameter in Depallet Vacuum Gripper Mode of the Task vision_move

In the vision_move Task, in *Operational mode* → *Tool Type* → *Depallet Vacuum Gripper*, a new parameter **Fixed Edge Corner Offset** is added for offsetting with the fixed edge corner of the vacuum gripper during depalletizing.

MECH-CENTER 1.6 RELEASE NOTES

This chapter introduces the new features, improvements, and resolved issues of Mech-Center 1.6.

4.1 Mech-Center 1.6.1 Release Notes

This section introduces the new features, improvements, and resolved issues of Mech-Center 1.6.1.

4.1.1 New Features

Support Modbus TCP Protocol

Mech-Center 1.6.1 can be used as a slave device software, providing the standard interface option MODBUS TCP SLAVE for data communication with the master device (PLC or robot controller).

You need to set the slave IP, port number, host address, and byte order in *Mech-Interface* → *Deployment Settings*. For specific samples, please see Modbus TCP: Siemens SIMATIC S7 PLC and Modbus TCP: Mitsubishi Q series PLC.

Support Standard Interface Integration with UR Robot

Mech-Center 1.6.1 has the newly added UR Cap plugin Mech-Mind 3D Vision Interface, which supports the seamless integration of UR robots with the Mech-Mind Vision System through the Standard Interface.

The plugin provides plug-and-play programming templates to reduce programming difficulty for users and instantly enable UR robots to perform vision-guided picking tasks.

The plugin supports the following UR robots:

- UR E series (Polyscope 5.3 and higher)
- UR CB series (Polyscope 3.9 and higher)

Support Standard Interface Integration with TM Robot

Mech-Center 1.6.1 has added a TM Standard Interface program. The integration of the robot with the Mech-Mind Vision System can be implemented instantly by burning the TM standard interface program to the TM robot.

Updated the TM Full-control Program

Mech-Center 1.6.1 has been re-adapted to the newer-version TM robot system. After upgrading the software, please re-load the TM full-control burn-in program.

Added Version Check for FANUC, ABB, KUKA, and Kawasaki Robot-side Full-control Burn-in Programs

To avoid errors in flange poses during calibration due to the out-of-date programs on the robot side, Mech-Center 1.6.1 added a version check function for the full-control programs of FANUC, ABB, KUKA, and Kawasaki.

Attention: Version check will be performed only when the full-control function is upgraded and the robot program version needs to be updated. Therefore, it is not necessary to re-load the robot program every time the software is upgraded. For temporary full-control, or if the full-control program has been modified on-site, the technical support team can help upgrade the robot program to be compatible with Mech-Center 1.6.1.

4.1.2 Improvements

Added a Pop-up Prompt When a Virtual Camera Was Detected

When a Mech-Vision project uses a virtual camera to guide a real robot, the software will pop up a window to remind the potential risks of the ongoing operation to avoid production accidents caused when the user forgets to switch the virtual camera to a real camera in Mech-Vision.

Added the Function of Sending Flange Pose for Command 201 of the Standard Interface

To avoid the position error caused by using joint angle calculation under EIH, Mech-Center 1.6.1 has the new function of sending the flange pose with command 201 of the Standard Interface, which helps more accurately set the robot poses for image capturing.

Optimized Error Log Messages

In Mech-Center 1.6.1, the log can print error codes for some error messages, and the log supports linking to instructions related to the error codes in the user manual. This improvement helps you intuitively understand the codes and error messages received by the robot, thus quickly locating and solving problems.

Optimized the Sync Button of the Mech-Vision Project List

In Mech-Center versions earlier than 1.6.1, when the Sync button was clicked, the entire project numbering would be reset to the order in which Mech-Vision auto-loaded the projects, after which the order you edited would be reset. In Mech-Center 1.6.1, the Sync button has been optimized to facilitate adding Mech-Vision projects.

Attention:

- After adding a new project in Mech-Vision, clicking the project list Sync button in Mech-Center will not affect the order of previous projects.
- When deleting or renaming a project in Mech-Vision, the project's number in Mech-Center will be reset. Before clicking the Sync button, please keep a record of the original order.

Interface Optimizations

If Mech-Vision and Mech-Viz are already running, clicking **Start** in the toolbar of Mech-Center will bring the Mech-Vision and Mech-Viz software windows to the front.

4.1.3 Resolved Issues

Failed to Start Automatically After the Computer Was Started

Mech-Center 1.6.0 might fail to start automatically after the computer was started. This issue has been fixed in Mech-Center 1.6.1.

Siemens Snap7 Interface Service Failed to Start

The issue that Siemens Snap7 interface service failed to start has been fixed in Mech-Center 1.6.1.

Slow Startup of the Mech-Vision and Mech-Viz Software

The issue of slow startup of the Mech-Vision and Mech-Viz software has been fixed in Mech-Center 1.6.1.

Issues in Kawasaki, ABB, YASKAWA Standard Interface Example Programs

In Mech-Center 1.6.1, the running errors of Kawasaki, ABB, YASKAWA Standard Interface example programs have been fixed.

4.2 Mech-Center 1.6.0 Release Notes

4.2.1 New Features

Logs Formatted as a Table

To effectively help locate issues, in Mech-Center V1.6.0, logs have been improved in the following respects:

- Added log source information in the log panel. Log sources currently include Mech-Vision, Mech-Viz, Mech-Center, Interface, and Adapter.
- Added the function of filtering logs by level. Log levels currently include “debug”, “info”, “warning”, and “error”.
- The Log Viewer supports viewing logs generated by Mech-Vision, Mech-Viz, and Mech-Center.

Changes in Standard Interface Commands

The Standard Interface of Mech-Center V1.6.0 must be used with Mech-Viz V1.6.0. In Mech-Viz V1.6.0, the Task ID parameter is added to Tasks. The Standard Interface also supports setting branches and indexes using Task names or Task IDs. The usages of the following commands have been modified in Mech-Center V1.6.0:

- When command 203 is used to set the branch exit in the Mech-Viz project, the branching Task is specified by Task ID instead of Task name.
- When using command 204 to set the move index, the move-type Task is specified by Task ID instead of Task name.
- When using command 206 to obtain the DO signal list of suction cup blocks, the parameter “Receiver” of Task “set_do_list” should be set to “StandardInterface”. Besides, this command puts no constraints on the names, Task names, and numbers of Tasks of Mech-Viz projects.
- When using command 601 to receive user-defined notifications, the “notify” Task’s name no longer needs to be manually set to “Standard Interface Notify”; instead, the “notify” Task’s parameter “Receiver” needs to be set to “StandardInterface”.

Added TCP Interface Commands

To work with Mech-Viz V1.6.0, the following TCP interface commands have been added to Mech-Center V1.6.0:

- Command 110: used to support sending user-defined port data when the parameter "Port Type" of Step "Procedure Out" is set to "Dynamic".
- Command 210: used to get the planning results calculated by Task "visual_move" (a new feature of Mech-Viz V1.6.0).
- Command 207: used to get Mech-Viz Task parameters from configuration files.
- Command 208: used to set Mech-Viz Task parameters to configuration files.

Multiple TCP Client Connections to Mech-Center

Mech-Center V1.6.0 supports multiple TCP clients to connect to Mech-Center at the same time, which is convenient for multiple robot stations to communicate with the vision system simultaneously.

Built-in Python Environment in the Software

To facilitate the setup and usage of the software, Python 3.6.5 environment is built into Mech-Center V1.6.0. The software package no longer contains the environment checking tool. The built-in Python environments of Mech-Center V1.6.0 and the computer's system are separated. If an Adapter project needs an extra Python library, please install it to the "python" directory of the file location of Mech-Center.

Tip: A Python library can be installed as follows:

1. Open the **Command Prompt** or **PowerShell** program.
 2. Switch to the "python" directory of the Mech-Center software, such as C:\Mech-Mind\Mech-Center-1.6.x\python.
 3. Execute the following command: `./python -m pip install library_name`.
-

Online Help Documentation

The online help documentation is available for Mech-Center V1.6.0. You can access the Mech-Center V1.6.0 online help documentation via Help → User Manual. In the online help documentation, instructions on full-control programming and interface programming will be updated timely and extensive instructions on using interface examples of PLC and robots are available.

4.2.2 Improvements

Added Robot and PLC Example programs

Mech-Center V1.6.0 has example programs for performing simple picking through the Standard Interface for five robot brands: ABB, FANUC, KUKA, KAWASAKI, and YASKAWA. In addition, Mech-Center V1.6.0 has function block samples of various interface commands for Siemens PLC S7, which makes it easy for you to quickly build PLC projects.

Added Fanuc CRX Robot Plug-in Installation Package

Mech-Center V1.6.0 has a teach pendant plug-in for Fanuc CRX robots, which modularizes the Standard Interface functions after the smart teach pendant plug-in is installed to achieve code-free robot programming.

Added Adaptations to Doosan and Dobot Robots

Mech-Center V1.6.0 supports full control for Doosan and Dobot robots.

Support for 4K Display

Mech-Center V1.6.0 supports 4K monitors.

4.2.3 Resolved Issues

Fixed Pose Transmission Errors

In earlier Mech-Center versions, if the Standard Interface was called by multiple Mech-Viz projects, there would be wrong numbers of poses transmitted. This issue has been fixed in Mech-Center V1.6.0.

Fixed Kawasaki RS Robot Project Execution Issue

When a KAWASAKI RS robot was under full control by Mech-Viz, if the Mech-Viz project involved linear motion, the project might not be executed properly. This issue has been fixed in Mech-Center V1.6.0.

Fixed YASKAWA Robot Calibration Program Issues

In Mech-Center V1.6.0, the issues in the calibration program of YASKAWA robots using the Standard Interface have been fixed.

Fixed Quaternion Calculation Errors

In Mech-Center V1.6.0, the issue of quaternion calculation errors has been fixed.