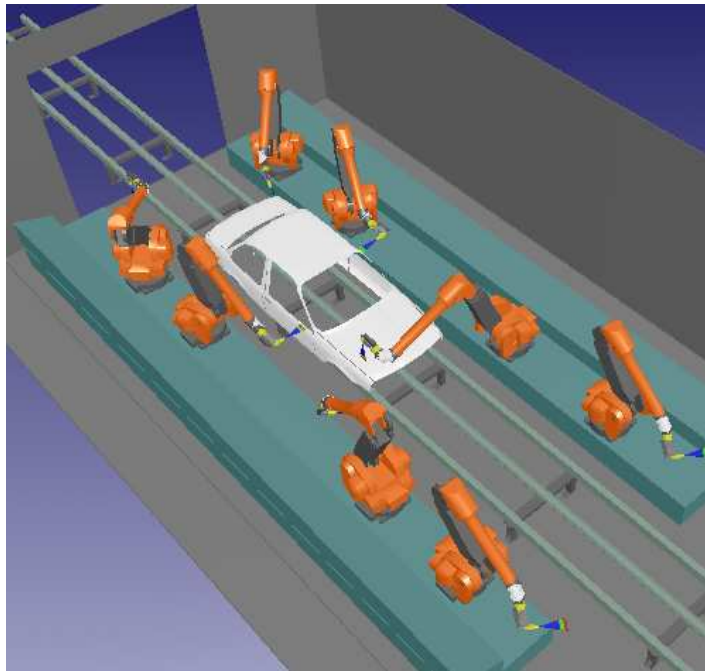


# Update

## EASY-ROB™ V4.606



March 2008

Version 1.0



# EASY-ROB™

## Table of contents

|  |    |
|--|----|
| Introduction .....                                   | 5  |
| The Project Manager.....                             | 7  |
| Project overview .....                               | 8  |
| Detailed information.....                            | 8  |
| Display screenshots and run videos.....              | 9  |
| Load a workcell into EASY-ROB™ .....                 | 9  |
| New functionality in the Visual File Interface ..... | 10 |
| Path history.....                                    | 10 |
| Preferred Paths .....                                | 11 |
| Pre-selection of the displayed file type.....        | 11 |
| Sort by date .....                                   | 12 |
| Screenshots as JPG and BMP.....                      | 12 |
| Sending data to the Project Manager.....             | 13 |
| Creating screenshots with Capture Image .....        | 14 |
| Improvement of the Merge Function .....              | 15 |
| Other Enhancements .....                             | 16 |
| Toolbar „Select Robot“ .....                         | 16 |
| Menu in Dialogs.....                                 | 16 |
| Extension in the Teach Window Dialogs .....          | 17 |
| ERC Searcher .....                                   | 18 |
| New example workcells.....                           | 19 |
| Extended tutorial for digital signals.....           | 19 |
| ERPL-Commands .....                                  | 20 |
| API-Functions.....                                   | 21 |
| Motion Planning.....                                 | 21 |
| Robot Attributes.....                                | 22 |
| Save .....   | 23 |
| Mouse Mode.....                                      | 24 |
| Mouse Picking .....                                  | 24 |
| AuxUpdate.....                                       | 26 |
| Contact.....   | 27 |
| Notes .....  | 29 |



# EASY-ROB™ V4.606

## Introduction

The current EASY-ROB™ Version 4.606 provides again new functions and improvements.

The new Project Manager offers different functionalities to administer EASY-ROB™ workcells and project related information. This helps the user to keep an overview about the projects.

The Visual File Interface has been extended. Beside a path history, preferred paths, a pre-selection of the file type and a sort-by-date-function, it's possible to send project related information straight to the Project Manager.

The EASY-ROB™ Capture Image provides now the function to save screenshots as well in the JPEG-format.

Beside the new example workcells to clarify the mode of operation of digital signals and how to program with digital signals, the tutorial including the TrainLib has been extended by a chapter called „Multi-Program“.

Information about the operation of the new functions you will find in the revised operation references.

We would like to thank our customers and users, who send suggestions and requirements for further development.

Thank you



Stefan Anton

EASY-ROB  
3D Robot Simulation Tool



## The Project Manager

The **Project Manager** – available with EASY-ROB™ V4.606 – is a new functionality supporting the user to keep the overview about the projects and to ensure a quick access to its workcells.

EASY-ROB™ workcells can be administered together with project related information in the Project Manager. By using different search filter the information and the workcells can be recalled in very short time.

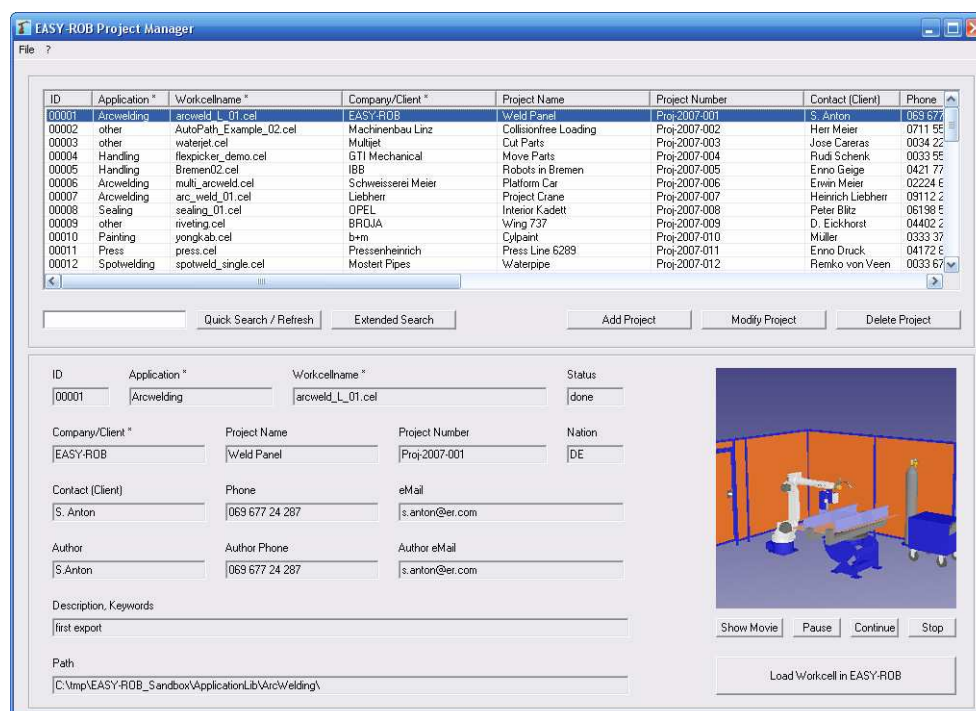
The project data can be inserted either through a mask into the Project Manager or through the **Visual File Interface**.

The Project Manager can play back video files of the simulation.

To start the Project Manager use the short-key „Ctrl+Shift+P“ or use the menu: File -> Load -> Start Project Manager. The data of the Project Manager will be stored in the file

„ ./ProjectManagerDb/er\_project\_manager.dat “ gespeichert.

Backup copies will be placed in the directory „ ./ProjectManagerDb/backup\_db/ “.



## Project overview

The project overview shows all stored projects in table form.

Useful functions beside add, modify and delete a project are the search functions with different filters to search for specific entries.

| ID    | Application * | Workcellname *          | Company/Client *   | Project Name          | Project Number | Contact (Client)  | Phone   |
|-------|---------------|-------------------------|--------------------|-----------------------|----------------|-------------------|---------|
| 00001 | Arcwelding    | arcweld_L_01.cel        | EASY-ROB           | Weld Panel            | Proj-2007-001  | S. Anton          | 069 677 |
| 00002 | other         | AutoPath_Example_02.cel | Machinenbau Linz   | Collisionfree Loading | Proj-2007-002  | Herr Meier        | 0711 55 |
| 00003 | other         | waterjet.cel            | Multijet           | Cut Parts             | Proj-2007-003  | Jose Careras      | 0034 22 |
| 00004 | Handling      | flexpicker_demo.cel     | GTI Mechanical     | Move Parts            | Proj-2007-004  | Rudi Schenk       | 0033 55 |
| 00005 | Handling      | Bremen02.cel            | IBB                | Robots in Bremen      | Proj-2007-005  | Enno Geige        | 0421 77 |
| 00006 | Arcwelding    | multi_arcweld.cel       | Schweisserei Meier | Platform Car          | Proj-2007-006  | Erwin Meier       | 02224 6 |
| 00007 | Arcwelding    | arc_weld_01.cel         | Liebherr           | Project Crane         | Proj-2007-007  | Heinrich Liebherr | 09112 2 |
| 00008 | Sealing       | sealing_01.cel          | OPEL               | Interior Kadett       | Proj-2007-008  | Peter Blitz       | 06198 5 |
| 00009 | other         | riveting.cel            | BRQJA              | Wing 737              | Proj-2007-009  | D. Eickhorst      | 04402 2 |
| 00010 | Painting      | yongkab.cel             | b+m                | Cylpaint              | Proj-2007-010  | Müller            | 0333 37 |
| 00011 | Press         | press.cel               | Pressenheinrich    | Press Line 6289       | Proj-2007-011  | Enno Druck        | 04172 6 |
| 00012 | Spotwelding   | spotweld_single.cel     | Mostert Pipes      | Waterpipe             | Proj-2007-012  | Remko von Veen    | 0033 67 |

Quick Search / Refresh   Extended Search   Add Project   Modify Project   Delete Project

## Detailed information

Detailed information about the project will be displayed in a separate area when the user selects a project from the list.

|   |                |                  |        |
|---|----------------|------------------|--------|
| ID  | Application *  | Workcellname *   | Status |
| 00001   | Arcwelding     | arcweld_L_01.cel | done   |
| Company/Client *                                  | Project Name   | Project Number   | Nation |
| EASY-ROB  | Weld Panel     | Proj-2007-001    | DE     |
| Contact (Client)                                  | Phone          | eMail            |        |
| S. Anton  | 069 677 24 287 | s.anton@er.com   |        |
| Author  | Author Phone   | Author eMail     |        |
| S. Anton  | 069 677 24 287 | s.anton@er.com   |        |
| Description, Keywords                             |                |                  |        |
| first export                                      |                |                  |        |
| Path  |                |                  |        |
| C:\programme\EASY-ROB\ApplicationLib\Arc\welding\ |                |                  |        |

the following information will be displayed:

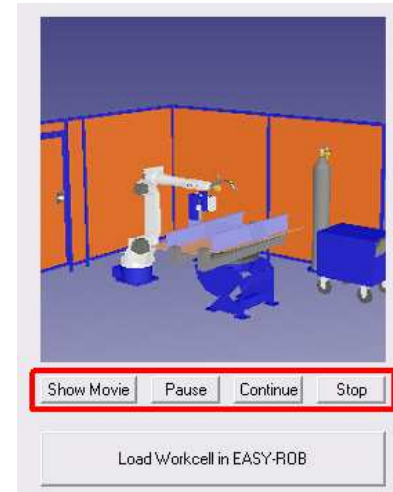
- Application
- Workcell name
- Status of the project
- Name of the Company/Client
- Name of the project
- Project number
- Nation
- Contact person with telephone and eMail
- Author of the cell with telephone and eMail
- a short comment
- absolute path of the workcell file



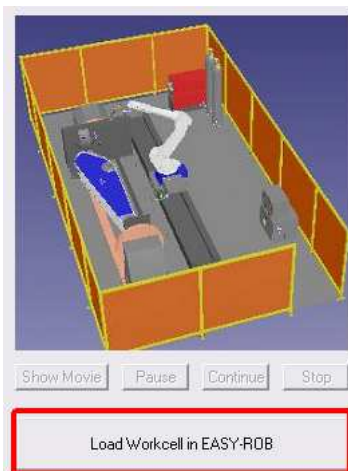
## Display screenshots and run videos

If the user created a screenshot for the workcell with the EASY-ROB™ Capture Image, this screenshot will be displayed automatically when selecting the project.

If the user created a video of the workcell, it is possible to run and watch the video in the Project Manager. The video must have exact the same filename like the workcell and need to be the format “AVI”. (Example: MyCell.cel, MyCell.jpg, MyCell.avi)



## Load a workcell into EASY-ROB™



To check a simulation again or to present the workcell, the user can load the cell from the Project Manager into EASY-ROB™ by using the „Load Workcell in EASY-ROB“-button.

## New functionality in the Visual File Interface

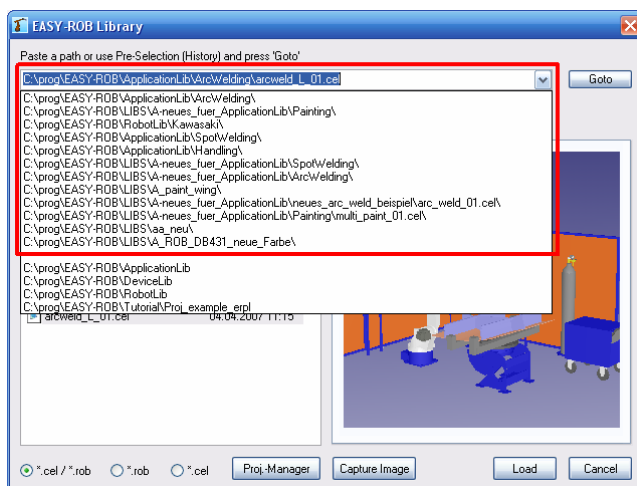
The Visual File Interface has been improved again by adding new functions.

### Overview of new functionalities:

- Path history
- Preferred paths
- Pre-selection of the displayed file type in the list
- Sort the files by date
- Screenshots as JPEG and BMP
- Send data of the workcell to the Project Manager
- new file format for Capture Image (JPEG)

### Path history

Every time when loading a workcell or a robot into EASY-ROB™, the system will store the origin path where the data coming from into the path history. The maximum number of listed paths is 12. Identical path names will be listed only one time.

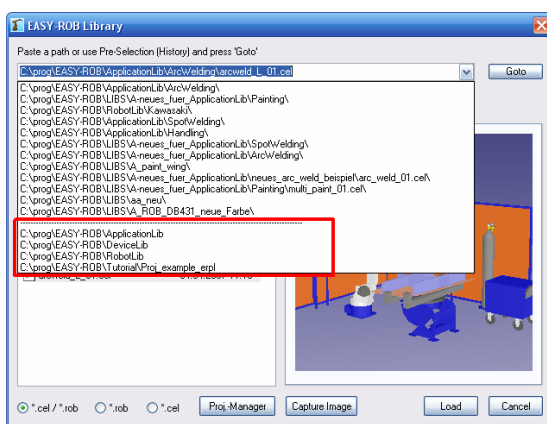


The idea of the path history is to speed up and to ease the access the last loaded cells or robots by using a Pull-down-Menu.

The file „er\_LoadFromLibPb.ini“, which contains the path history, will be administrated automatically. Do not edit is manually

## Preferred Paths

In addition to the automatically generated path history the new version provides in the same Pull-down-Menu „preferred paths“. The list of „preferred paths“ paths has to be created by the user and will be stored in the file „er\_LoadFromLibPb\_prefered.ini“. The basic file will be created automatically if it doesn't exist. The difference to the path history is, that the preferred paths won't change and that enables the user to have a quick access to the most important paths.



The list of preferred paths has to be created by the user and will be stored in the file „er\_LoadFromLibPb\_prefered.ini“ in the EASY-ROB™-target directory.

## Pre-selection of the displayed file type

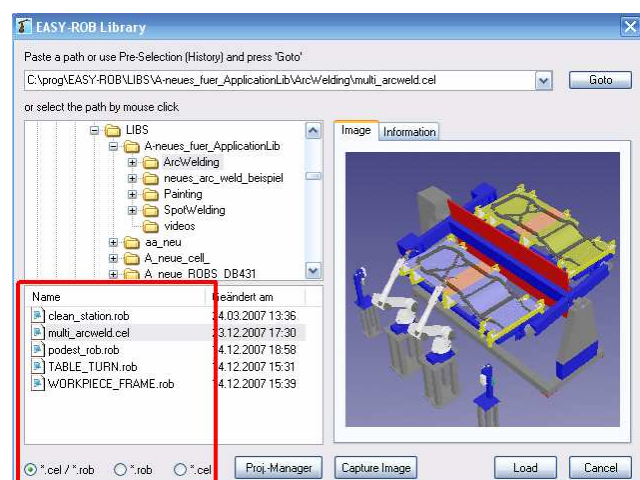
To keep track of the project when it's getting bigger with several workcells and robots, the displayed file type can be pre-selected.

The three options:

- Workcell and Robot (\*.cel, \*.rob)
- only Robot (\*.rob)
- only Workcell (\*.cel)

The default setting after the start EASY-ROB™ is „Workcells and Robots“.

While a session the last selected filter is the current one and will be used on the next call of the library.



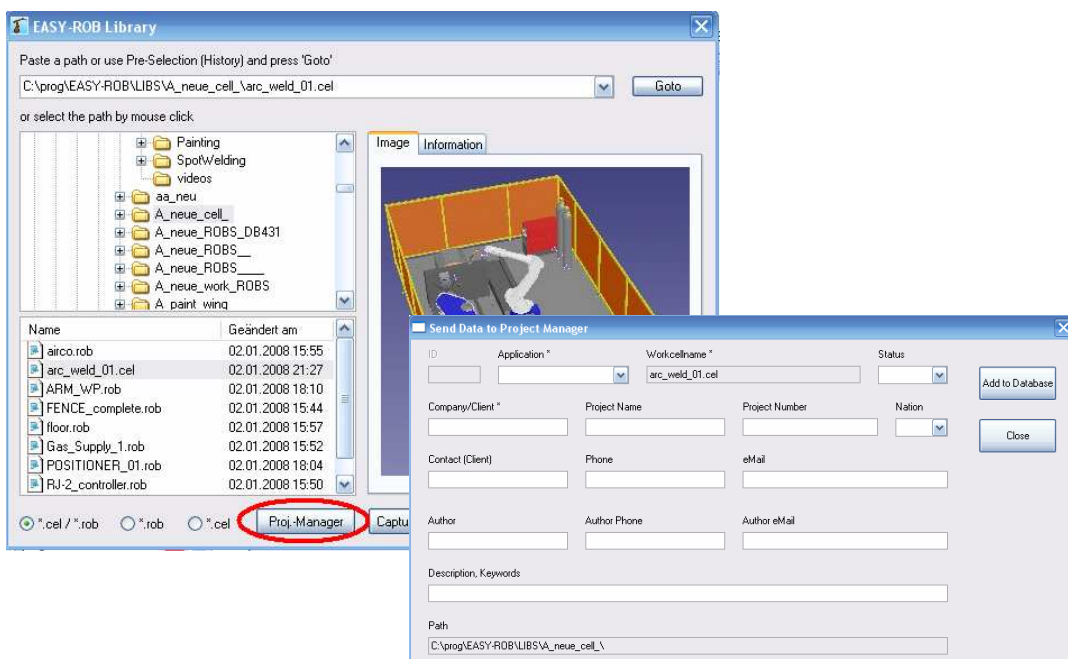


## Sending data to the Project Manager

Project data can be added through the Visual File Interface to the Project Manager.

To send project data to the Project Manager the user has to select first a workcell in the Visual File Interface and activate the function „Proj.-Manager“. A new mask will come up to take the project related data and save them in the Project Manager.

**Tip:** Select first the desired file (in the example „arc\_weld\_01.cel“) – only by doing this the function „Project Manager“ will be activated.

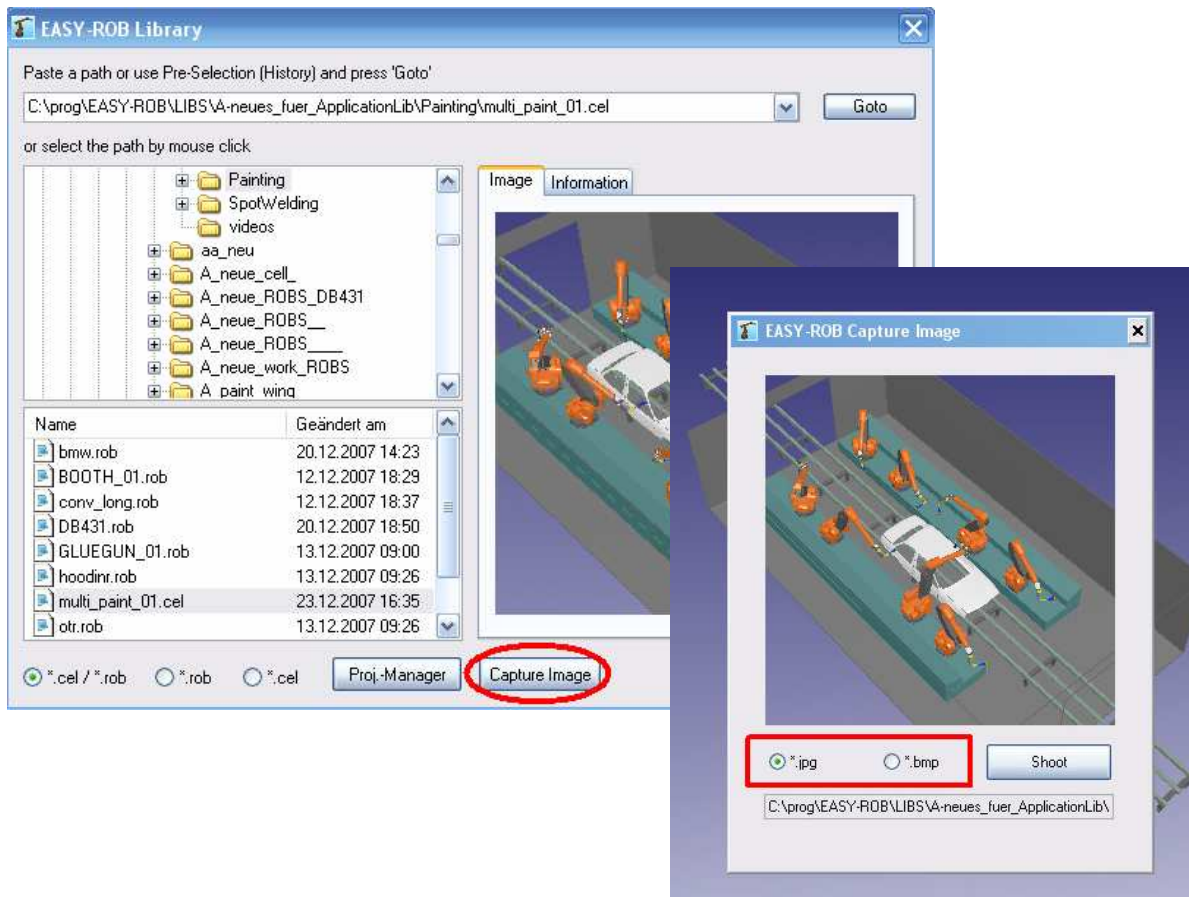


## Creating screenshots with Capture Image

In addition to the BMP-format the Capture Image can create the JPEG-format now.

After loading the cell or the robot, the „Capture Image“-function from the EASY-ROB™ library dialog has to be called. After positioning the window and the selection of the file type the screenshot can be taken by hitting the „Shoot-button“.

**Tip:** Select first the file you want to capture (in the example „*multi\_paint\_01.cel*“), before you activate „Capture Image“. By doing this the system will know the file name and can take it for the picture.



## Improvement of the Merge Function

The existing “Merge-Function” (since 4.6) has been improved to get a better performance while rendering huge CAD data.

When doing a “merge” the system will check how many polygons each object contains and how many objects from the same color existing. Depending on that information the system will determine whether the object is in good or bad condition.

All objects with the same color will be combined in one object. So the number of objects will decrease and the number of polygons per object will increase. That will result in a better performance which is as well an advantage while collisions check.

A problem may occur if a workpiece consist of a lot of objects with the same color and in total more than 2.000.000 polygons. In that case the “merge” will create a single object with all the polygons in long polygon chains. Those long chains can cause trouble while loading them into the graphic board.

To avoid problem in those cases the maximum number of polygons per object is limited to 100.000 now. That means for the example above that the system will create more than 20 objects with the same color.



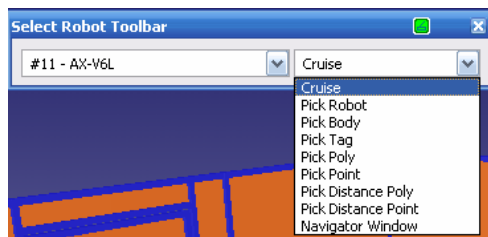
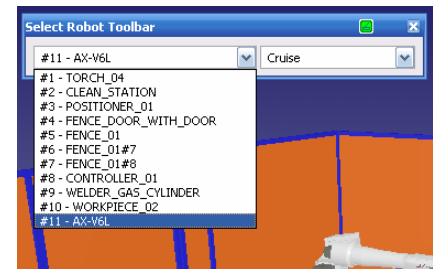
## Other Enhancements

### Toolbar „Select Robot“

The Version 4.606 provides a new Toolbar to speed up the daily work.

The „Select Robot“-Toolbar contains two Pull-down-menus.

The first Pull-down-menu provides a list of the loaded robots. The user can select one to manipulate it.

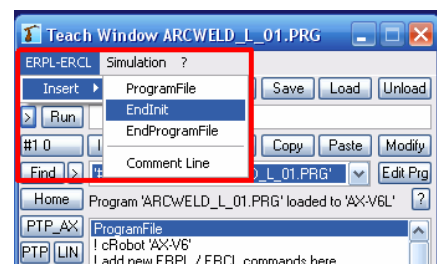
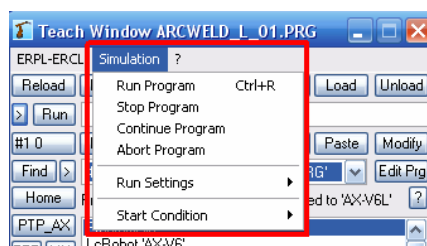


By the second Pull-down-menu the user can run an activity like „Cruise“ or „measuring a distance“.

The usage of the Pull-down-menus will reduce the number of mouse clicks.

## Menu in Dialogs

The new Menus in the dialogs containing dialog related functions to enable the user to start the action by few clicks.



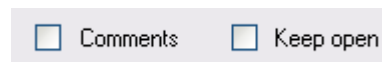


## Extension in the Teach Window Dialogs

The three Teach Window Dialogs „Motion Commands“, „Control Commands“ and „ERCL Commands“ have been reorganized and got two new check-boxes.

When adding a command, it is now up to the user whether the comments will be added or not. To add the comments the „Comments“-check box has to be activated.

If the user wants to add several commands from the same dialog, this dialog can be kept open by the check box „Keep open“.



## ERC Searcher

The example library contains a workcell and related program for each ERPL-/ERCL-command.

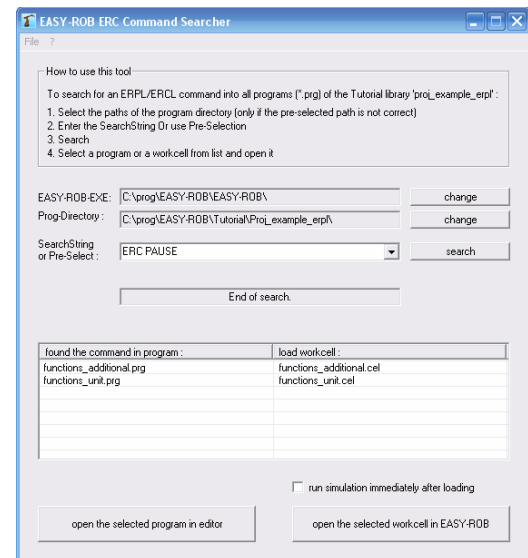
The ERC Command Searcher will support the user to search for a specific ERC command in the example library “Proj\_example\_erpl”, which is normally installed in the directory:

*“.\EASY-ROB\Tutorial\Proj\_example\_erpl”*

The library contains many short example programs with different ERC commands to give an impression how the commands are working.

The user can insert the command manual or can use the pre-selection to search for it

The user can open the program in an editor or load a workcell into EASY-ROB™ to see the correct usage of the command.



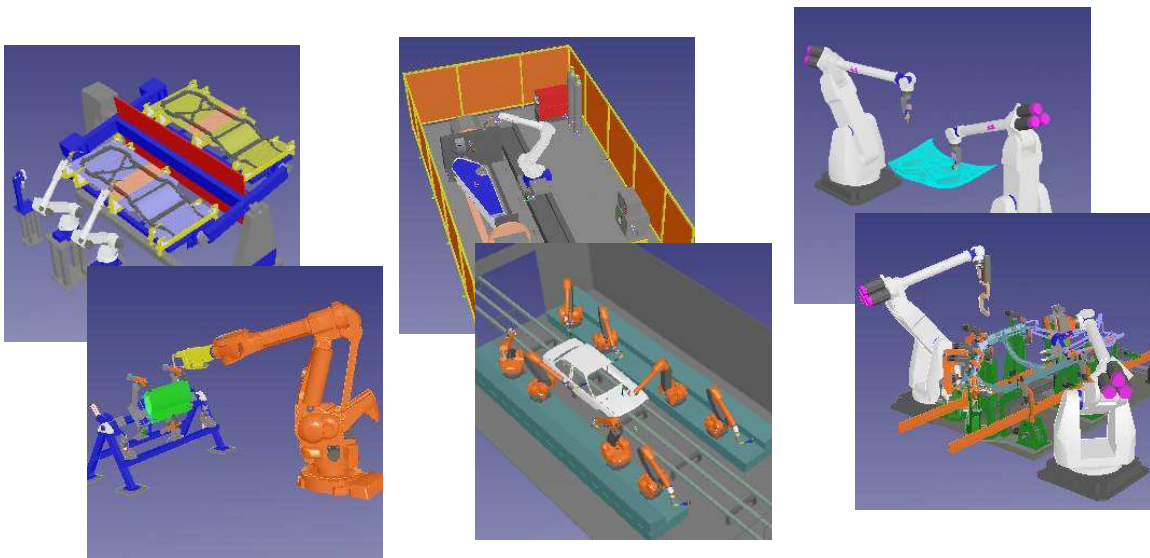
Read as well the chapter “ERC Command Searcher” in the Operation References.

## New example workcells

The functionality „**Multi-Program**“ – available since the major release V4.6 – and the possibility to use digital signals was the reason to create a few new example workcells for different applications.

With the new examples and the extended tutorial its easy for the user to understand the programming with digital signals.

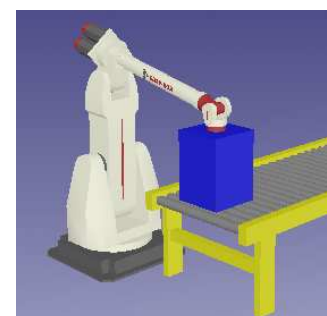
The new examples are placed in the ApplicationLib.



## Extended tutorial for digital signals

According to the new functions we extended the tutorial. It will guide the user step-by-step through the task to build up a simulation with Multi-Program and digital signals.

The TrainLib-Tutorial is located in the manual folder.



## ERPL-Commands

- SPEED\_PTP\_AX v1 .. vn [m/s, deg/s]
- ACCEL\_PTP\_AX a1 .. an [m/s², deg/s²]
- SPEED\_PTP\_OV [%]
- ACCEL\_PTP\_OV [%]
- SPEED\_CP\_OV [%]
- ACCEL\_CP\_OV [%]
- SPEED\_ORI\_CP\_OV [%]
- ACCEL\_ORI\_CP\_OV [%]

The speeds (SPEED\_PTP\_AX) and the acceleration (ACCEL\_PTP\_AX) for a PTP movement will be programmed for each axis. That means that the „old“ SPEED\_PTP- and ACCEL\_PTP-commands (all axis will be programmed with the speed/acceleration) are obsolete. The system will copy the values for every rotational axis to speed\_ptp\_ax, if the old commands will be used anyhow.

Example: The 3rd axis of the robot is translational, axis 1,2 and 4 rotational.

```
speed_ptp_ax 50 60 0.5 50 // deg deg m deg
speed_ptp 20
```

will result internal in:

```
speed_ptp_ax 20 20 0.5 20 // deg deg m deg
```

It will be the same for ACCEL-commands.

Leading (dominant) axis on PTP-movement:

When a robot moves from A to B, the dominant axis will move with the given speed\_ptp\_ax speed, if this speed can be reached. But that depends from the distance AB and the accelerations.

```
SPEED_PTP_OV 20
```

result in: 20% of the maximum axis speed for each axis

Example: The maximum axis speeds of the robot are: 200° 250° 1500mm 400°.

```
speed_ptp_ov 50 // %
```

will result internal in:

```
speed_ptp_ax 100 125 0.75 200 // deg deg m deg
```

The new OV commands enable the user to set the speeds and accelerations in an easy way.

## API-Functions

See header files „er\_dvlp.h“ und „er\_dvlp\_ext.h“

### Motion Planning

- **float \*inq\_ipo\_path\_vq\_axis (void)**  
Programmed joint speed  
Return: pointer to programmed joint speed
- **float \*inq\_ipo\_path\_aq\_axis (void)**  
Programmed joint acceleration  
Return: pointer to programmed joint acceleration
- **float \*inq\_ipo\_path\_aq\_decel\_axis (void)**  
Programmed joint deceleration  
Return: pointer to programmed joint deceleration
- **float \*inq\_ipo\_path\_vq\_ov\_set (float ov)**  
Calculates programmed joint speed vq\_axis as percentage override value of the maximum robot joint speed \*inq\_v\_max()  
In: ov Percentage override value [1 . . 200]  
Return: pointer to programmed joint speed
- **float \*inq\_ipo\_path\_aq\_ov\_set (float ov)**  
Calculates programmed joint acceleration aq\_axis as percentage override value of the maximum robot joint acceleration \*inq\_a\_max()  
In: ov Percentage override value [1 . . 200]  
Return: pointer to programmed joint acceleration
- **float \*inq\_ipo\_path\_vx\_ov\_set (float ov)**  
Calculates programmed cartesian speed vx as percentage override value of the maximum robot cartesian speed \*inq\_vx\_max()  
In: ov Percentage override value [1 . . 200]  
Return: pointer to programmed cartesian speed

## New API-Functions

- float \*inq\_ipo\_path\_ax\_ov\_set (float ov)**  
 Calculates programmed cartesian acceleration ax as percentage override value of the maximum robot cartesian acceleration \*inq\_ax\_max()  
 In:                                      ov                      Percentage override value [1 . . 200]  
 Return:    pointer to programmed cartesian acceleration
- float \*inq\_ipo\_path\_vx\_ori\_ov\_set (float ov)**  
 Calculates programmed cartesian orientation speed vx\_ori as percentage override value of the maximum robot cartesian orientation speed \*inq\_vx\_ori\_max()  
 In:                                      ov                      Percentage override value [1 . . 200]  
 Return:    pointer to programmed cartesian orientation speed
- float \*inq\_ipo\_path\_ax\_ori\_ov\_set (float ov)**  
 Calculates programmed cartesian orientation acceleration ax\_ori as percentage override value of the maximum robot cartesian orientation acceleration \*inq\_ax\_ori\_max()  
 In:                                      ov                      Percentage override value [1 . . 200]  
 Return:    pointer to programmed cartesian orientation acceleration

## Robot Attributes

- float \* inq\_vx\_max (void)**  
 Maximum cartesian TCP speed [m/s]  
 Return:    pointer to max. cartesian speed
- float \* inq\_ax\_max (void)**  
 Maximum cartesian TCP acceleration [m/s<sup>2</sup>]  
 Return:    pointer to max. cartesian acceleration
- float \* inq\_vx\_ori\_max (void)**  
 Maximum cartesian TCP orientation speed [rad/s]  
 Return:    pointer to max. cartesian orientation speed
- float \* inq\_ax\_ori\_max (void)**  
 Maximum cartesian TCP orientation acceleration [rad/s<sup>2</sup>]  
 Return:    pointer to max. cartesian orientation acceleration

## New API-Functions

### Save

- `int _info_line_msg_q (int moni, char *s, float *q, int no_auto_scale=0)`

Vector output into Message Window, for each joint of current robot, scaled by DEG or 1000

|         |               |   |
|---------|---------------|---|
| In:     | moni          | 0 - message in message window only, 1- save in moni_msg.txt |
|         | s             | comment   |
|         | q             | vector of robot joint values for example                    |
|         | no_auto_scale | 0 - scale to deg or mm if rotational or translational joint |
|         |               | 1 - no auto scale   |
| Return: |               | 0 - OK, 1 - Error   |

- `int er_vad_DataSaveCell_ext (char *new_cel_fln=NULL,int confirm_overwrite=1, int reset_pos=1,int save_pos=0)`

Save current loaded cell file as ...

|         |                   |   |
|---------|-------------------|---|
| In:     | new_cel_fln       | Cell file name                              |
|         | confirm_overwrite | Prompts user to confirm if file exist       |
|         | reset_pos         | 1 - Reset to start condition, 0 - no reset  |
|         | save_pos          | 1 - Save as start condition, 0 - don't save |
| Return: |                   | 0 - OK, 1 - Error                           |

- `int er_vad_DataSaveRobot_ext (char *new_rob_fln=NULL,int confirm_overwrite=1)`

Save current selected robot as ...

|         |                   |                                       |
|---------|-------------------|---------------------------------------|
| In:     | new_rob_fln       | Robot file name                       |
|         | confirm_overwrite | Prompts user to confirm if file exist |
| Return: |                   | 0 - OK, 1 - Error                     |

- `int er_vad_DataSaveTool_ext (char *new_tol_fln=NULL,int confirm_overwrite=1)`

Save current selected robot tool as ...

|         |                   |                                       |
|---------|-------------------|---------------------------------------|
| In:     | new_tol_fln       | Tool file name                        |
|         | confirm_overwrite | Prompts user to confirm if file exist |
| Return: |                   | 0 - OK, 1 - Error                     |





## New API-Functions

- `int er_vad_PickDistancePoint (int hold_first_point=0)`

Enables the user to measure the distance between two vertices.

In:      `hold_first_point`      1<sup>st</sup> ChkBox in NavigatorWindow

Return:      0-OK, 1-Error

Read the current MousePickung status.

New functions in addition to „`int er_vad_Get_cPickMode (void)`“:

- `int er_vad_Get_cPickMode_Chk_1 (void)`

Current check\_1 box mode in Navigator Window

Return:      check box mode

- `int er_vad_Get_cPickMode_Chk_2 (void)`

Current check\_2 box mode in Navigator Window

Return:      check box mode

```
const int PICK_RESULT_DEFAULT      = 0    // same as iPOINTS
const int PICK_RESULT_oPOINTS     = 1    // position w.r.t. obj
const int PICK_RESULT_oNORMALS    = 2    // normal w.r.t. obj
const int PICK_RESULT_iPOINTS     = 3    // position w.r.t. inertia system
const int PICK_RESULT_iNORMALS    = 4    // normal w.r.t. inertia system
const int PICK_RESULT_idXYZ       = 5    // distance w.r.t. inertia system
const int PICK_RESULT_DISTANCE    = 6    // lengths of idXYZ
```

- `float *er_vad_Get_cPickResults (int result_idx)`

XYZ position or vector component of picked item

In:      `result_idx`      [PICK\_RESULT\_DEFAULT ... PICK\_RESULT\_DISTANCE]

Return:      pointer to result vector, xyz component

## New API-Functions

### AuxUpdate

New AuxUpdate(int idx) indexes, if the user picks out an object (geometry, robot, tag, etc.).

```
const int AUX_UPDATE_IDX_SELECTION_NOHIT = 32;
const int AUX_UPDATE_IDX_SELECTION_HIT   = 33;

const int MP_HIT_INVALID      = 0;      // invalid item picked
const int MP_HIT_ROBOT       = 1;      // geometry from robot group picked
const int MP_HIT_TOOL        = 2;      // geometry from tool group picked
const int MP_HIT_BODY        = 3;      // geometry from body group picked
const int MP_HIT_TAG         = 4;      // tag picked
const int MP_HIT_POLY        = 5;      // center of polygon picked (normal
                                       // avialable)
const int MP_HIT_LINE        = 6;      // line picked
const int MP_HIT_POINT       = 7;      // point picked
const int MP_HIT_COORSYS     = 8;      // geomety coorsys picked
const int MP_HIT_CADPREVIEW  = 9;      // geomety in CAD Preview picked
```

- **int \*er\_vad\_Get\_cPickItem (void)**

Return picked item, case AUX\_UPDATE\_IDX\_SELECTION\_HIT in AuxUpdate(.

Return: Item [MP\_HIT\_INVALID . . . MP\_HIT\_CADPREVIEW]

### Example:

```
EXPORT_C int AuxUpdate(int idx)
// This function is called from EASY-ROB for some below cases
// It allows you to update dialog or start some special actions
{
    switch (idx) {
        case AUX_UPDATE_IDX_SELECTION_HIT: {
            int cPickItem = er_vad_Get_cPickedItem();
            _info_line_msg(0, "pick item is = %d", cPickItem);
            // do something
        } break;
        case AUX_UPDATE_IDX_SELECTION_NOHIT:
            _info_line_msg(0, "nothing hit, pick again", b);
            break;
        default:
            break;
    }
    return 0; // Success
}
```

## Contact

### EASY-ROB

3D Robot Simulation Tool

Stefan Anton

Hans - Thoma - Str. 26a  
60596 Frankfurt/Main, Germany

Tel. +49(0)69 677 24 287

Fax. +49(0)69 677 24 320

Email: [info@easy-rob.com](mailto:info@easy-rob.com), [sales@easy-rob.com](mailto:sales@easy-rob.com)

Web: <http://www.easy-rob.com>

### EASY-ROB Customer area

Online available: Program Updates and Robot libraries

Web: <http://www.easy-rob.com/en/special/customer-area>

Data:

User: customer

Password: \*\*\*\*\*



## Notes