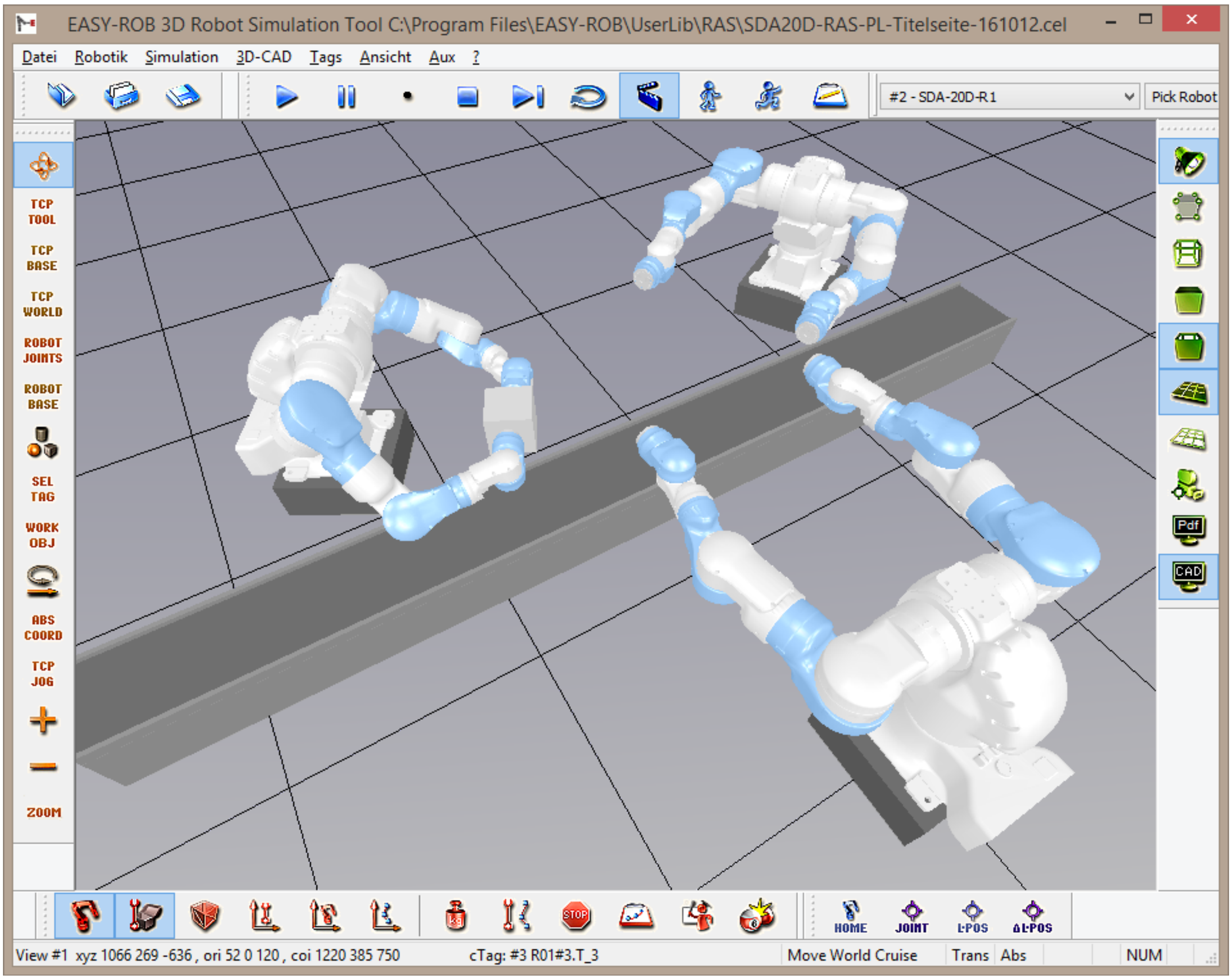
The New Version

EASY-ROB™ V7.0



EASY-ROB 3D Robot Simulation Tool C:\Program Files\EASY-ROB\UserLib\RAS\SDA20D-RAS-PL-Titelseite-161012.cel

View #1 xyz 1066 269 -636 , ori 52 0 120 , coi 1220 385 750 cTag: #3 R01#3.T_3 Move World Cruise Trans Abs NUM

October 2016

Version 1.0

EASY-ROB™

Table of content

EASY-ROB™ V7.0 Update

Dear EASY-ROB™ Community!

Finally we did it –
EASY-ROB ™ version 7.0 is introduced and the new version in the quick overview offers the following highlights:

- **Language switching**
  In addition to the English language as the default, the GUI can now be displayed in German and Chinese. By the „easy-rob-localizationx64.ini“ you can quickly and conveniently change the language.
- **Robot Assembly *.ras**
  Complex kinematic robotic models can now be created easily and conveniently as a Robot Assembly and can be used as many times as necessary in * .cel files. Thus among others also Dual-Arm robots like the Motoman/Yaskawa SDA20D incl. programs, paths and Tags can be used modularly.
- **Layout functions for the 3D-PDF Export**
  For the first time there is the possibility to use a personalized layout for the 3D-PDF. The layout of the 3D PDF Export can now be supplemented with logos and static information on the labeling. This allows you to create customized 3D PDF files of the simulation results of your clients for better identification etc.
- **TCP JOG coordinate system**
  In order to keep always the right direction in mind, a smart coordinate system can be seen at the tip of the robot during the Cartesian or axial jogging, which illustrates the flange coordinate system.
- **Dongle access is now thread-safe**
  The dongle access is synchronized by Mutex objects and semaphores, so that accesses from threads are made safely, which is only conditionally given by the API of the manufacturers.
- **Collision test is now thread-safe in the Robotics Simulation Kernel**
  The class ERK_CAPI_SIM_COLLISION delivers new methods, so that the collision test happens Thread-safe.
- **New robot models**
  The robot library incl. external axis is growing steadily thanks to your stimulus and models of manufacturers like Motoman/Yaskawa, Fanuc, KUKA, ABB, Stäubli, Comau and UniversalRobots have been added.
- **Multi-Robot**
  With increasing simulation complexity, a separation of the Multi-Kin and Multi-Program version was hardly possible. EASY-ROB therefore decided to combine both versions in the new Multi-Robot version as a powerful simulation tool for multi-robot applications.

We would like to inform you about the following:

Due to the low demand for x86 applications of EASY-ROB ™, the 32bit development have been stopped with version 7.0. EASY-ROB can now concentrate on the further development of the 64bit version with its entire development performance. If you still have an x86 application, please contact our sales department.

From now on, the new EASY-ROB ™ version 7.0 is available free of charge to all customers with a valid v7.0 license or a software maintenance contract.

For customers of older versions it is possible to purchase an update. Please contact our sales department at +49 6192 921 70 79 or sales@easy-rob.com and provide your hardware / dongle number and the version number of your license. To determine your hardware number (HwNr): Start EASY-ROB ™ and open the menu

- „? → License Info“.

If necessary, repeat this step for your additional licenses.

We would like to thank you by now for your suggestions and improvement proposals.

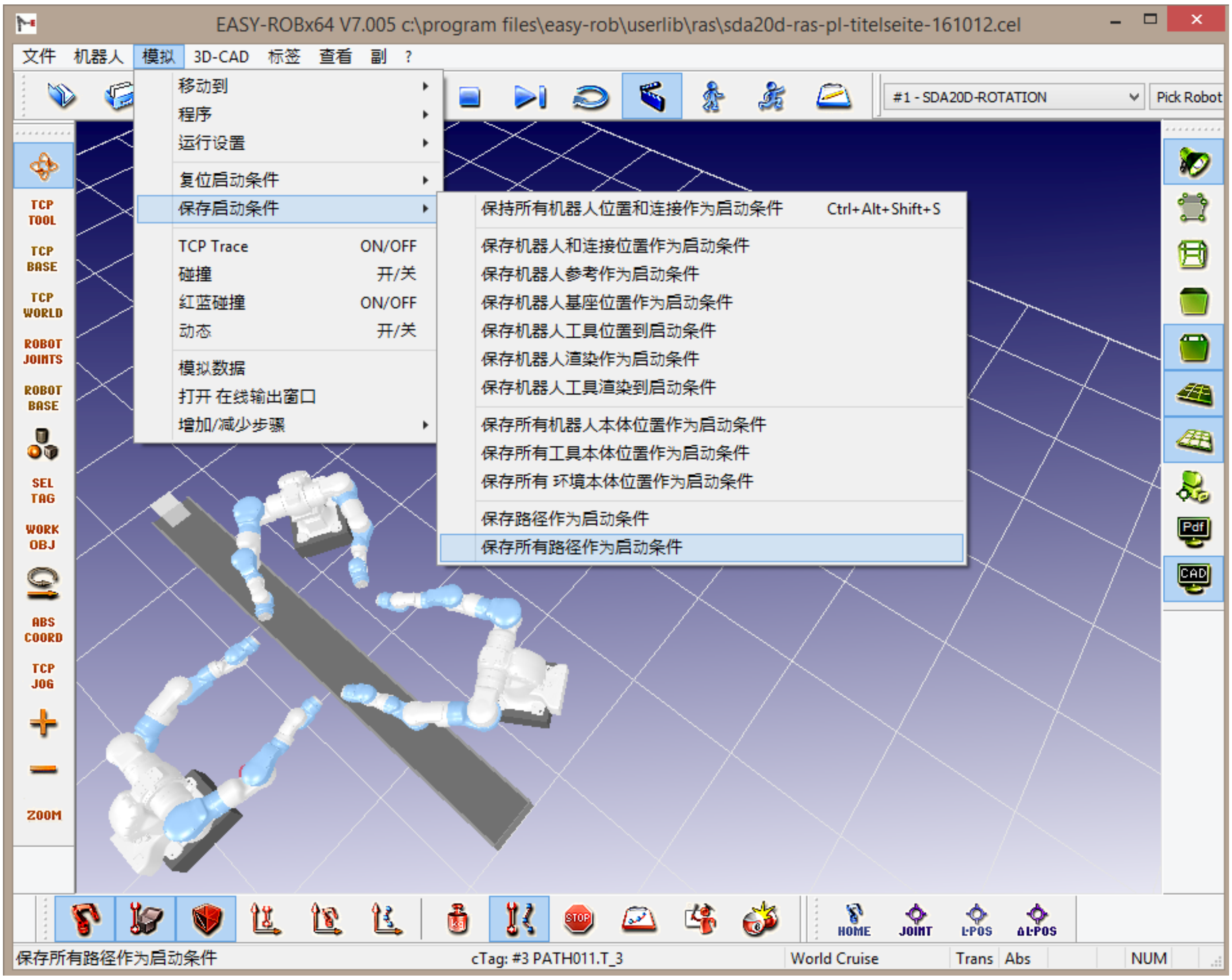
Thank you very much!

EASY-ROB
3D Robot Simulation Tool

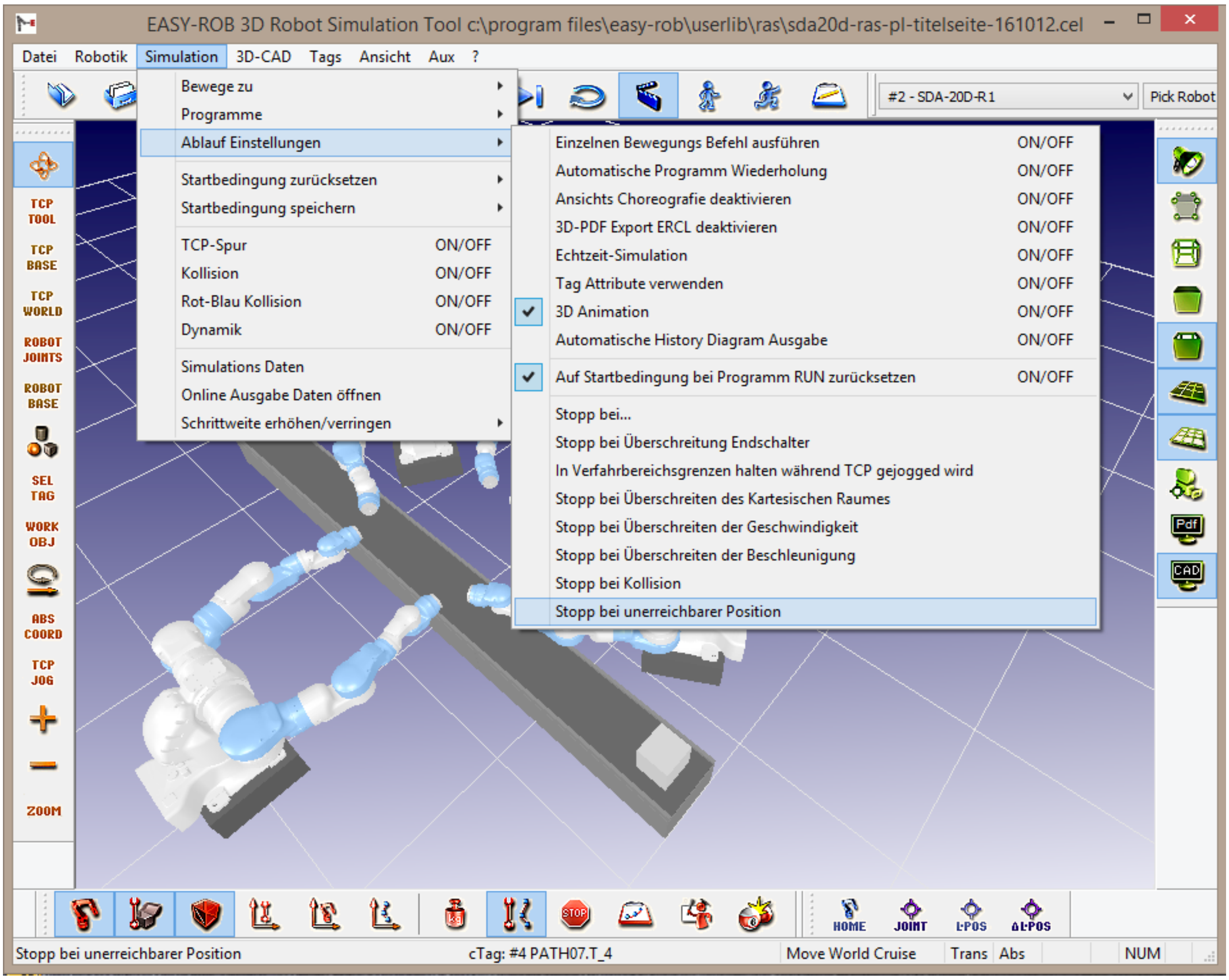
Language switching (ENGLISH / GERMAN / CHINESE)

For the first time in the history of EASY-ROB ™, you can optionally switch the graphical user interface (GUI) between the following languages:

- English
- German or
- Chinese.


EASY-ROBx64 V7.005 c:\program files\easy-rob\userlib\ras\sda20d-ras-pl-titelseite-161012.cel
文件 机器人 模拟 3D-CAD 标签 查看 副 ?
移动到
程序
运行设置
复位启动条件
保存启动条件
TCP Trace ON/OFF
碰撞 开/关
红蓝碰撞 ON/OFF
动态 开/关
模拟数据
打开 在线输出窗口
增加/减少步骤
保持所有机器人位置和连接作为启动条件 Ctrl+Alt+Shift+S
保存机器人和连接位置作为启动条件
保存机器人参考作为启动条件
保存机器人基座位置作为启动条件
保存机器人工具位置到启动条件
保存机器人渲染作为启动条件
保存机器人工具渲染到启动条件
保存所有机器人本体位置作为启动条件
保存所有工具本体位置作为启动条件
保存所有 环境本体位置作为启动条件
保存路径作为启动条件
保存所有路径作为启动条件
#1 - SDA20D-ROTATION
Pick Robot
TCP TOOL
TCP BASE
TCP WORLD
ROBOT JOINTS
ROBOT BASE
SEL TAG
WORK OBJ
ABS COORD
TCP JOG
+
−
ZOOM
Pdf
CAD
STOP
HOME
JOINT
L-POS
ΔL-POS
保存所有路径作为启动条件
cTag: #3 PATH011.T_3
World Cruise
Trans
Abs
NUM

GUI in Chinese


EASY-ROB 3D Robot Simulation Tool c:\program files\easy-rob\userlib\ras\sda20d-ras-pl-titelseite-161012.cel
Datei Robotik Simulation 3D-CAD Tags Ansicht Aux ?
Bewege zu
Programme
Ablauf Einstellungen
Startbedingung zurücksetzen
Startbedingung speichern
TCP-Spur ON/OFF
Kollision ON/OFF
Rot-Blau Kollision ON/OFF
Dynamik ON/OFF
Simulations Daten
Online Ausgabe Daten öffnen
Schrittweite erhöhen/verringen
Einzelnen Bewegungs Befehl ausführen ON/OFF
Automatische Programm Wiederholung ON/OFF
Ansichts Choreografie deaktivieren ON/OFF
3D-PDF Export ERCL deaktivieren ON/OFF
Echtzeit-Simulation ON/OFF
Tag Attribute verwenden ON/OFF
✓ 3D Animation ON/OFF
Automatische History Diagram Ausgabe ON/OFF
✓ Auf Startbedingung bei Programm RUN zurücksetzen ON/OFF
Stopp bei...
Stopp bei Überschreitung Endschalter
In Verfahrbereichsgrenzen halten während TCP gejogged wird
Stopp bei Überschreiten des Kartesischen Raumes
Stopp bei Überschreiten der Geschwindigkeit
Stopp bei Überschreiten der Beschleunigung
Stopp bei Kollision
Stopp bei unerreichbarer Position
#2 - SDA-20D-R1
Pick Robot
TCP TOOL
TCP BASE
TCP WORLD
ROBOT JOINTS
ROBOT BASE
SEL TAG
WORK OBJ
ABS COORD
TCP JOG
ZOOM
STOP
HOME
JOINT
L-POS
ΔL-POS
Pdf
CAD
Stopp bei unerreichbarer Position
cTag: #4 PATH07.T_4
Move World Cruise
Trans Abs
NUM

GUI in German

How can the language of the GUI be changed?

The localization takes place under the menu point
„File → Edit → EASY-ROB System Files → Localization file“
and is achieved by excluding the *undesired* languages with a "!" before the lines.

en-US	for English
de-DE	for German
zh-Hans	for Chinese (traditional)

On the following page the German language was activated in the localization file.

Restart of EASY-ROB™ is required for activation

To change the GUI language, the localization file must be saved after modification and EASY-ROB™ has to be restarted.

EASY-ROB ™ searches for the following localization DLLs in the installation directory of version 7.0:

- EasyRobwx64CHS.dll
- EasyRobwx64DEU.dll
- EasyRobwx64ENU.dll

If the localization DLLs do not exist, EASY-ROB ™ starts with English GUI, which is also the default setting.

Localization file "easy-rob-localizationx64.ini" at a glance

```
! EASY-ROB 3D Robot Simulation Tool
!
! EXE - Simulation Version x64
!
! Localization file
!
!      "easy-rob-localizationx64.ini"
!
! Make sure that the localization Dlls exist
!
! Allowed language keys are
!
! en-US        ==>    EasyRobwENUx64.dll (english)
! de-DE        ==>    EasyRobwDEUx64.dll (german)
! zh-Hans      ==>    EasyRobwCHSx64.dll (chinese)
!
! Note:'!' or ';' comment a line
!
!en-US
de-DE
!zh-Hans
!
```

The localization file easy-rob-localizationx64.ini is located in the installation directory of EASY-ROB ™.

On our own behalf

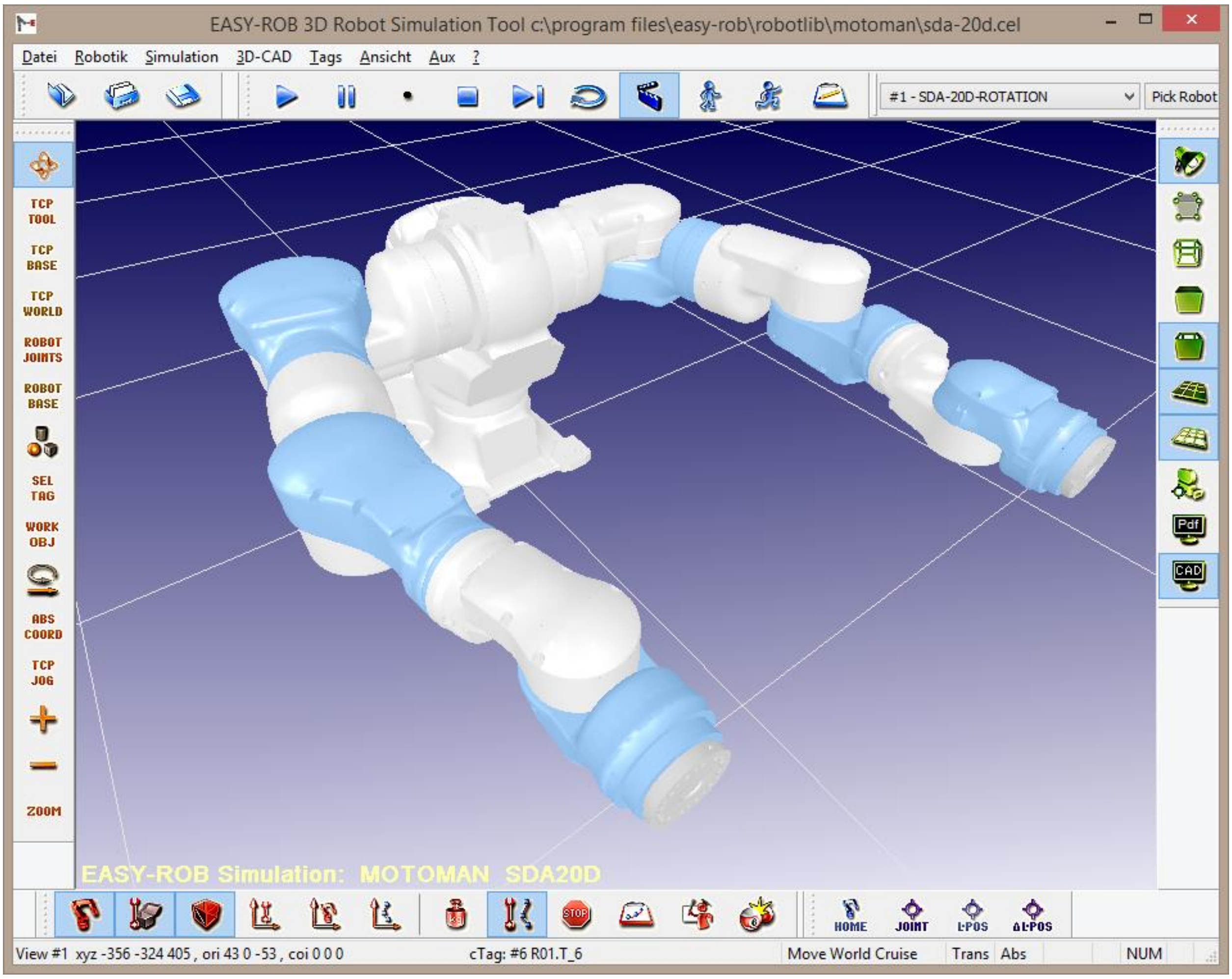
With approximately 2200 lines of translation per new GUI language, EASY-ROB is proud of the work being done and is looking forward to the growing EASY-ROB ™ community.

Robot Assembly

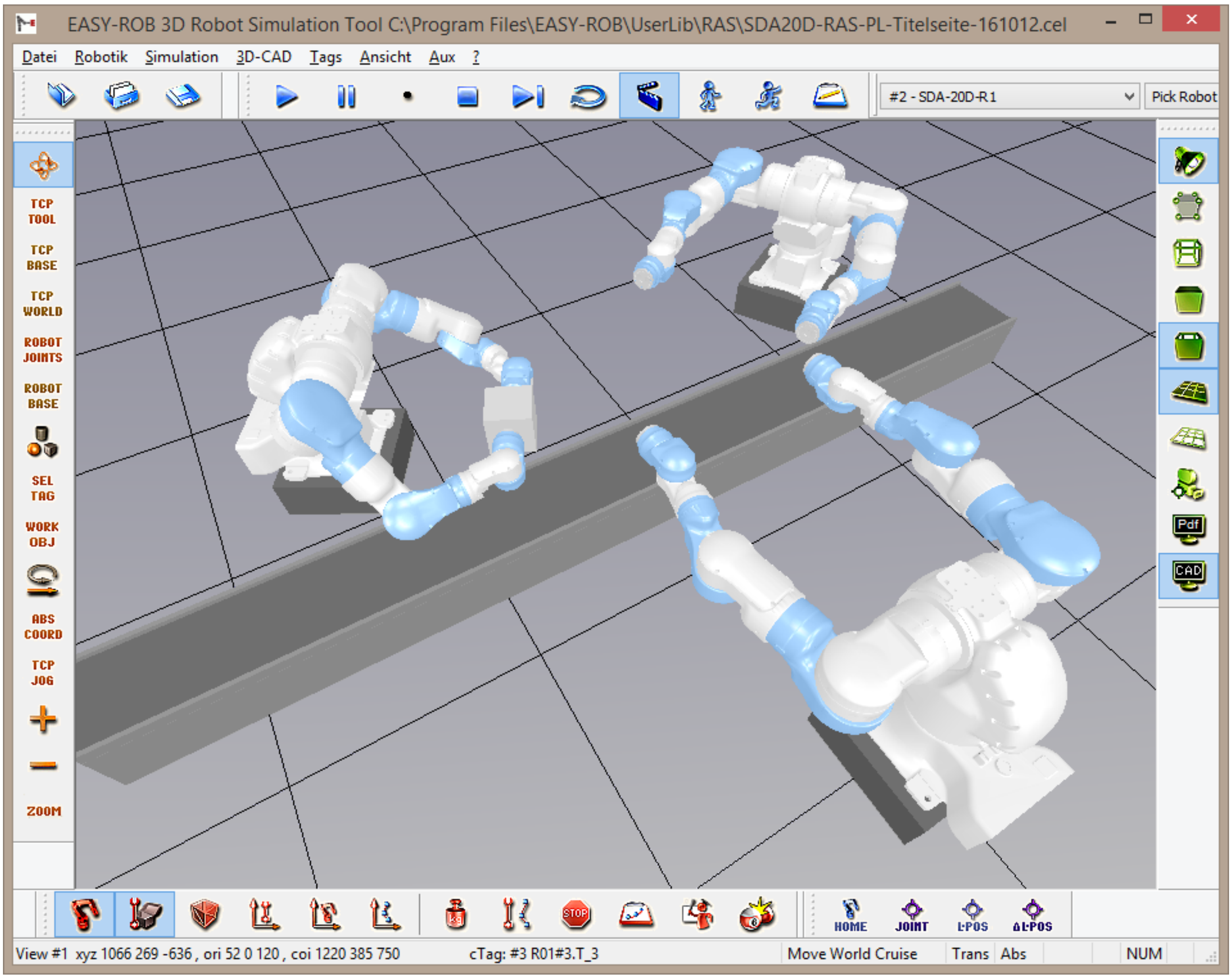
Complex kinematic models can now be easily and conveniently assembled as robot assemblies from individual robots, stored and reused in a work cell-independent manner. The designation assembly is based on assemblies from CAD. In these assemblies, models from individual robots / devices / devices can be assembled and constructed. Optionally, programs, paths and Tags can also be stored in the Robot Assembly.

Your EASY-ROB ™ V7.0 generates a new file extension for Robot Assemblies: *.ras; the Robot Assembly file.

Thus, e.g. the dual arm robot type "SDA20D" from Motoman/Yaskawa consisting of 3 robots with a total no. of 15 axes must not longer be saved as *.cel file, but can now be loaded and simulated as a Robot Assembly as often as desired in *.cel files, please see screenshot:



Motoman/Yaskawa SDA20D as a single Robot Assembly file "SDA20D-RAS-PL.ras"


EASY-ROB 3D Robot Simulation Tool C:\Program Files\EASY-ROB\UserLib\RAS\SDA20D-RAS-PL-Titelseite-161012.cel

View #1 xyz 1066 269 -636 , ori 52 0 120 , coi 1220 385 750 cTag: #3 R01#3.T_3 Move World Cruise Trans Abs NUM

Multiple Robot Assemblies type SDA20D in cooperation in one work cell
"SDA20D-RAS-PL-Titelseite-161012.cel"

Just store and load like robots

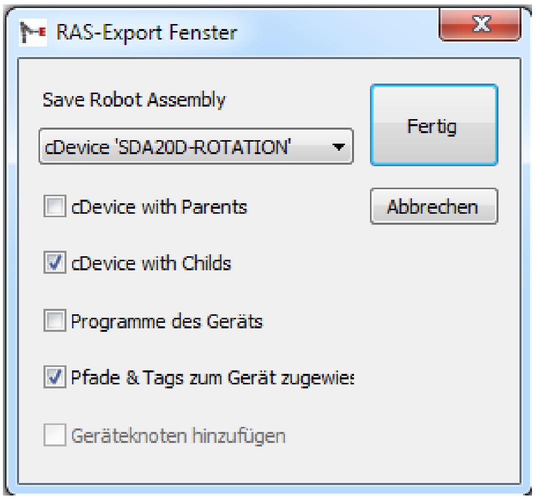
Robot assemblies are treated like robots and you can use not only attributes such as Home positions, but also created paths or tag points and even programs can be saved and loaded.

Under the menu

- "File → Save → Roboter Assembly file" your individual robot kinematics can be stored and

Loaded under

- " File → Load → Roboter Assembly file "



RAS-Export Fenster

Save Robot Assembly

cDevice 'SDA20D-ROTATION'

□ cDevice with Parents

☑ cDevice with Childs

□ Programme des Geräts

☑ Pfade & Tags zum Gerät zugewie:

□ Geräteknoten hinzufügen

Robot Assembly Save Dialog

When you save a Robot Assembly, you can optionally save all "Child Devices", the associated ERPL programs, and the assigned paths & Tags.

When reloading a Robot Assembly into one and the same workcell * .cel file, the path names are uniquely identified in the Tag Window by a "#" + index. Thus, on the one hand, a confusion of the Tag points is excluded and, on the other hand, it is possible that even a simulate without additional modification, even if the Robot Assembly has been shifted with respect to the world coordinate system, can be successfully executed.
In the ERPL programs of the various Robot Assemblies, you may need to adjust the call of the paths according to the index.

As usual, the handy Device Manager is available. Browse to your target directory and under the selection

- „.rob/.ras“

And load a Robot Assembly as if you are loading an ordinary robot.

Load and store also possible using methods of class ER_CAPI

See page 25 for more information or in the doxygen document „EASY-ROB ER_CAPI.chm“

Modular design of production lines possible

As a further application of the Robot Assemblies, the following modular design of simulations can also be implemented:
With the new Robot Assembly files, you can also create entire production stations or workstations and store them separately. These are then available to you in order to subsequently flexibly combine entire production lines into a modular design. Combine your production stations at your convenience and click together your production line quickly and individually.

Layout functions of the 3D-PDF Export incl. animation

Your complete 3D simulation with animation in only one PDF document!

Use the new EASY-ROB™ 3D-PDF Export plug-in and create a PDF document of your simulation within a few steps. Share easily and conveniently your simulation ideas with customers, partners and colleagues-
Without any additional software.

Applications

- Quick and easy presentation to third parties
- Disclosure of interactive simulation concepts
- Installation and maintenance instructions
- Documentation of difficult information
- Universal education and training material
- Interactive sales documentation for a improved understanding at customers

Layout functions for successful recognition value

In order to enable your customers, partners and employees to identify themselves better with their simulation results, the 3D-PDF Export in EASY-ROB™ version 7.0 has been extended by layout functions.

You can bring with up to 12 different pictures, among other things, logos and other static information, e.g. customer names, to the 3D-PDF page you have created. The control file "3D-PDF-Layout.3pl" not only determines the position of the images on the 3D-PDF, but also the scaling can be controlled by other parameters; these can be correspondingly compressed or stretched.

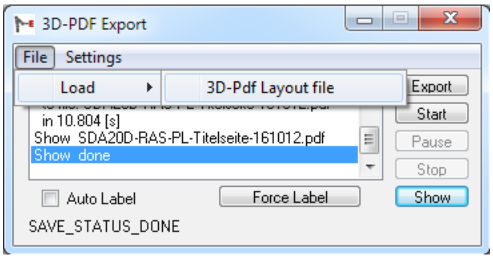
The control file as well as used images must not lie in the directory of the project file (* .cel), but can be integrated intelligently via "IMAGE_PATH". The images can be read as an independent and universal JPG format.

The navigation bar can now also be adapted, please see screenshot "Layout area and influence over control file".

In order to make your 3D-PDF a successful presentation, you have now the possibility to change the font size of the set labels quickly and easily via "LABELFONTSIZE".

Please re-use the * .3pl file

Of course, "* .3pl" files can be reloaded and re-used in the "3D-PDF Export"-dialog in the file menu after their creation, please see screenshot. This keeps your individual layout settings for future projects of your customers, etc.



in 10.804 [s]
Show SDA20D-RAS-PL-Titelseite-161012.pdf
Show done

SAVE_STATUS_DONE

Control file "3D-PDF-Layout.3pl" at a glance

The control file "3D-PDF-Layout.3pl" is an ASCII file and can be edited with a text editor e.g. Notepad.

```
! ------------------------------------------
! Date: 2016/09/14  Time: 10:43
! 3D-PDF Version v7004
! 3D-PDF Layout File "3D-PDF-Layout.3pl"
! ------------------------------------------
!
!(0,0)--> x  <-- width -->
!  |        |
!  v      height
!  y        |
!
! Important Notes:
! - Viewport-, Control-Area and all Images should be inside the Page-Area
! - Images cannot be on top of the ViewPort- or the Control-Area,
!   but can be used as background, if their size is less or equal to the Page-Area.
! - The number of possible loaded images is limited by 12
! - The "image path" and "image file" must be saved within quotes
! ------------------------------------------
!
! Page Area:  width height
PAGE_AREA            400.000  300.000
!
! View Port Area: left(x) upper(y)  width height
VIEWPORT_AREA        0.000  50.000  400.000  225.000
!
! Control Area: left(x) upper(y)  width height
CONTROL_AREA         0.000  275.000  400.000  25.000
!
! Label Font Size: Default = 10, valid values are between 1 and 50
LABELFONTSIZE        10
!
! Path where the images stored
! If no path is defined, the current working directory is valid
IMAGE_PATH           ""
!
! Images specifications for 3 images
! Parameter 1:   "Image file name"
! Parameter 2-5: left(x) upper(y)  width height
! Parameter 6:   Scaling is one of ISO_Stretch = 0 or ISO_CenterFit = 1
ADD_IMAGE            "3D-Pdf-Image-Logo.jpg"  0.000  0.000  87.920  50.000  1
ADD_IMAGE            "3D-PDF-IMAGE-TEXT.JPG"  87.920  0.000  312.080  50.000  1
!
```

Standard EASY-ROB ™ image files that can be replaced, supplemented and positioned at will.

3D-Pdf Export with Animation

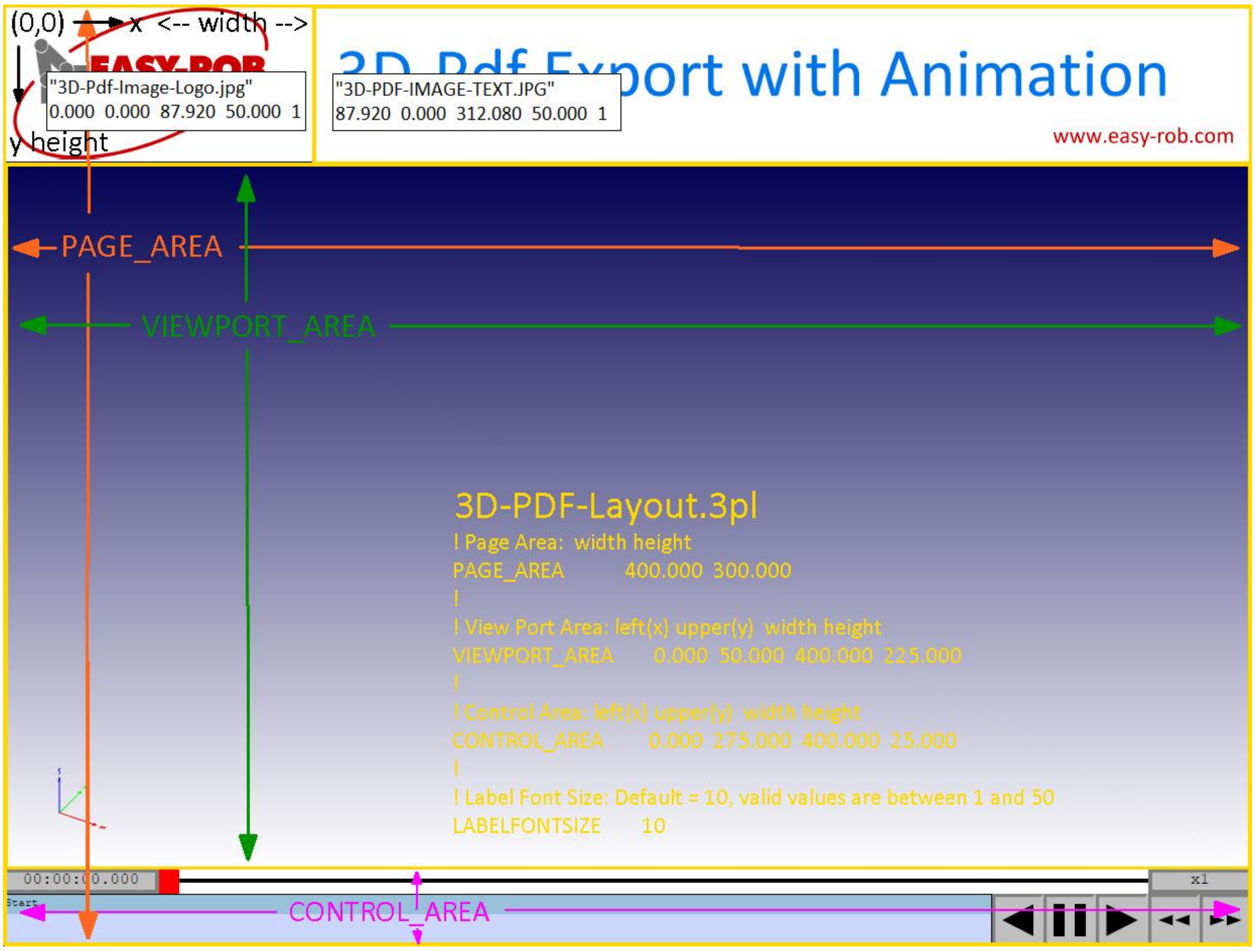
www.easy-rob.com

3D-Pdf-Image-Logo.jpg

3D-Pdf-Image-Text.jpg

Functions and parameters of the 3D PDF layout

The next screenshot shows the areas which you can influence:



"3D-Pdf-Image-Logo.jpg"
0.000 0.000 87.920 50.000 1

"3D-PDF-IMAGE-TEXT.JPG"
87.920 0.000 312.080 50.000 1

```
3D-PDF-Layout.3pl
! Page Area:  width height
PAGE_AREA         400.000  300.000
!
! View Port Area: left(x) upper(y)  width height
VIEWPORT_AREA      0.000  50.000  400.000  225.000
!
! Control Area: left(x) upper(y)  width height
CONTROL_AREA       0.000  275.000  400.000  25.000
!
! Label Font Size: Default = 10, valid values are between 1 and 50
LABELFONTSIZE      10
```

Layout area and influence over control file

- PAGE AREA = Adjust the total height and width
- VIEWPORT AREA = Manipulates height and width of 3D windows
- CONTROL AREA = Scaling of the navigation bar
- LABELFONTSIZE = Font size of labels
- IMAGE_PATH = Set path to used images; otherwise WORKDIR
- ADD IMAGE = Select, position, and scale images for layout

ERCL Commands for layout functions of 3D-PDF Export

The layout functions can be controlled via additional ERCL commands:

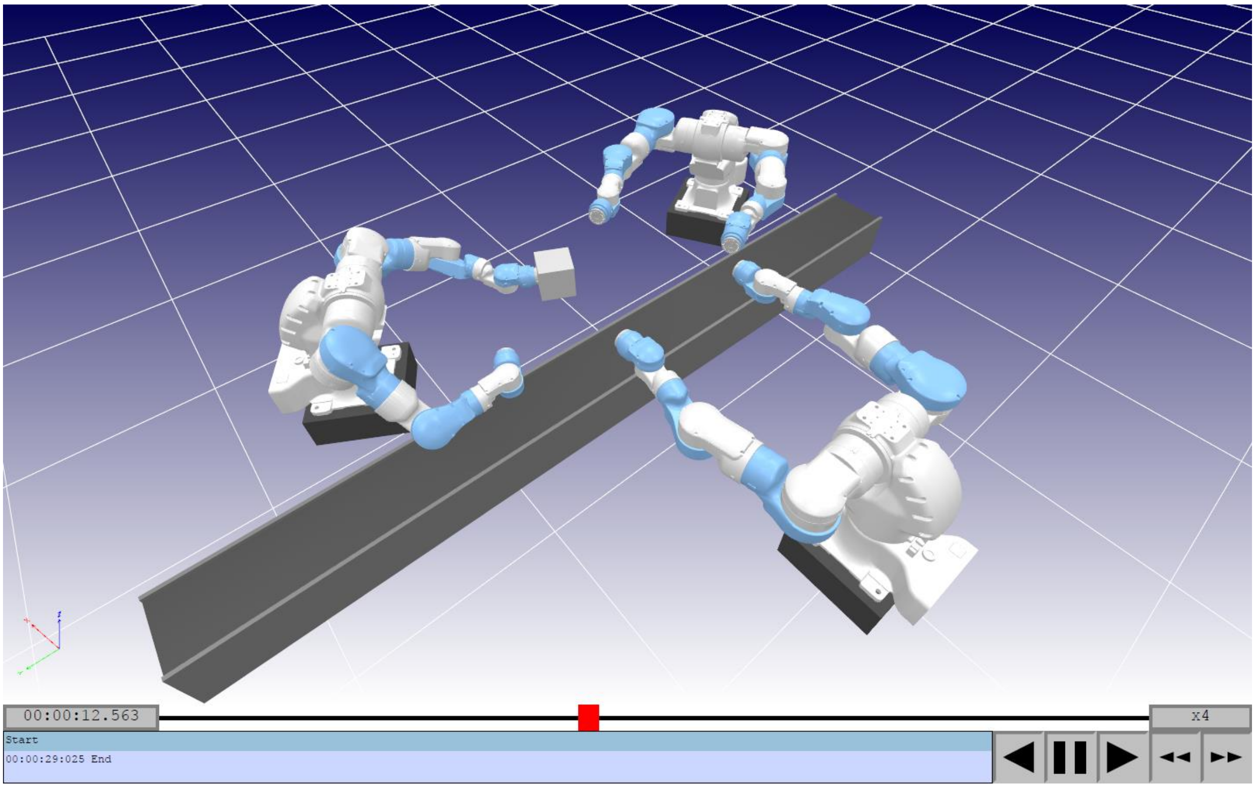
ERC _3D_PDF_EXPORT RESET_LAYOUT	
ERC _3D_PDF_EXPORT LOAD_LAYOUT	filename.3pl
ERC _3D_PDF_EXPORT SAVE_LAYOUT	filename.3pl
ERC _3D_PDF_EXPORT PAGE_AREA	width height
ERC _3D_PDF_EXPORT VIEWPORT_AREA	left(x) upper(y) width height
ERC _3D_PDF_EXPORT CONTROL_AREA	left(x) upper(y) width height
ERC _3D_PDF_EXPORT LABEL_FONT_SIZE	valid values are between 1 and 50
ERC _3D_PDF_EXPORT IMAGE_PATH	path
ERC _3D_PDF_EXPORT IMAGE_PATH	USERPROFILE
ERC _3D_PDF_EXPORT IMAGE_PATH	WORKING_DIRECTORY
ERC _3D_PDF_EXPORT IMAGE_PATH	3PL_FILE_FOLDER
ERC _3D_PDF_EXPORT IMAGE_PATH	""
ERC _3D_PDF_EXPORT ADD_IMAGE	"filename.jpg" left(x) upper(y) width height scaling

"EASY-ROB-ERPL_ENG.pdf" as a further document

A description of the commands can be found in the section "New ERCL commands" or in the corresponding document of the ERPL / ERCL programming language "EASY-ROB-ERPL_ENG.pdf"**.**

Store motion sequences with animation in 3D-PDF

In the Adobe® Reader, you can use the navigation bar to start, pause, stop, fast-forward, rewind, and change the speed (x1 / 64x to x64x). The time specifies the real process time.



Navigationsleiste im Adobe® Reader

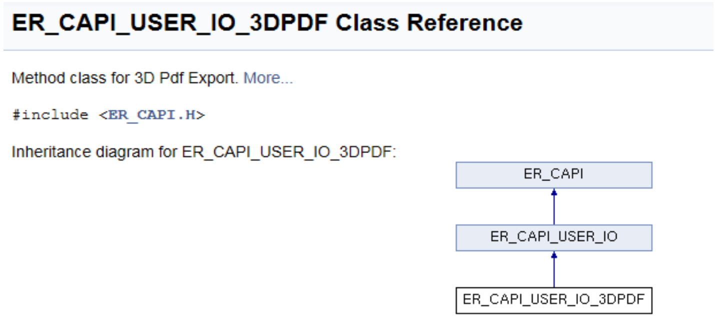
The 3D-PDF Export functionality can also be controlled via ERCL commands.
The following ERCL commands are available to the user:

ERC _3D_PDF_EXPORT SCREENSHOT [flnname]
ERC _3D_PDF_EXPORT ON / OFF [flnname]
ERC _3D_PDF_EXPORT SET_FILE flnname
ERC _3D_PDF_EXPORT SET_LABEL labelname
ERC _3D_PDF_EXPORT SET_PASSWORD passwordname
ERC _3D_PDF_EXPORT PAUSE
ERC _3D_PDF_EXPORT DEACTIVATE

A description of the commands can be found in the section "New ERCL commands" and in the document for the ERPL / ERCL programming language "EASY-ROB-ERPL_ENG.pdf".

The 3D PDF export can also be controlled via the API method class:

- ER_CAPI_USER_IO_3DPDF



ER_CAPI_USER_IO_3DPDF Class Reference

Method class for 3D Pdf Export. More...

```
#include <ER_CAPI.H>
```

Inheritance diagram for ER_CAPI_USER_IO_3DPDF:

Online Link:
http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/class_e_r___c_a_p_i___u_s_e_r___i_o__3_d_p_d_f.html

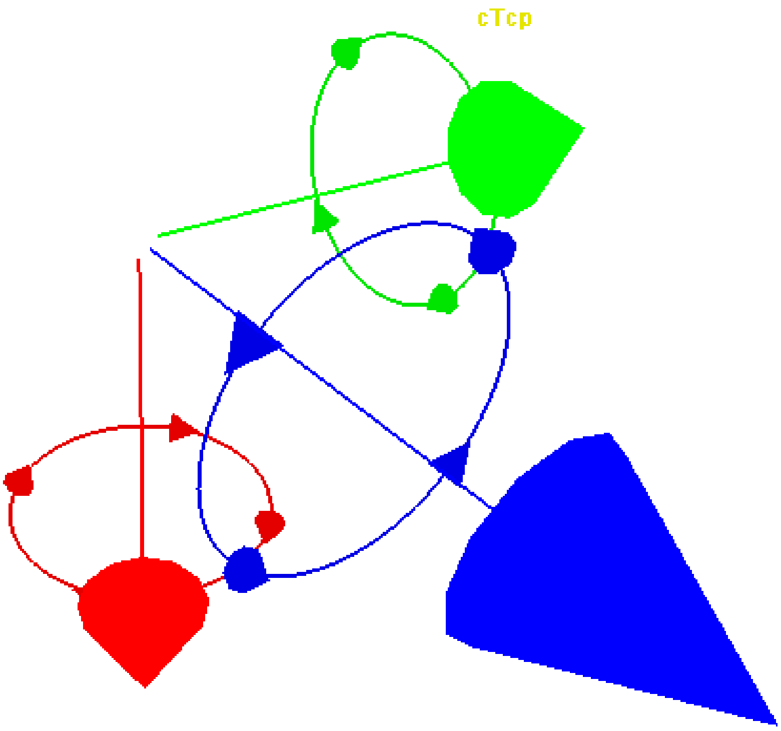
TCP JOG coordinate system

In order to keep always the right direction in mind, a smart coordinate system can be seen at the tip of the robot during the Cartesian or axial jogging, which illustrates the flange coordinate system.

Thereby the axes are shown below:

- X Achse = rot
- Y Achse = grün
- Z Achse = blau

The positive rotation directions for the rotation are drawn with concentric circles around the respective translatory axes/directions and the respective vector.



TCP JOG Coorsys TCP Tool in detail

Activation depends on the respective JOG mode

The TCP JOG coordinate system is activated via the menu

- „View → CoorSys → Show cRobot TCP JOG CoorSys “

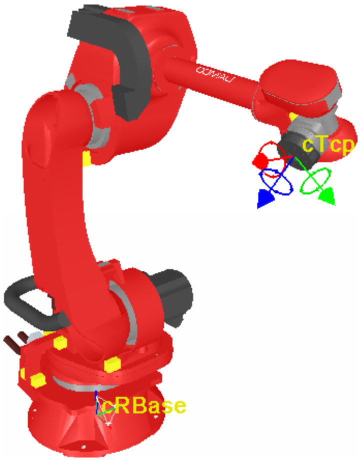
For the setting to be retained when restarting EASY-ROB ™, it can be saved in the environment file "easy-rob.env".

```
! Robot Tcp Jog coordinate system 1-Show , 0-Hide
ROBOT_TCP_JOG_COORSYS 1
```

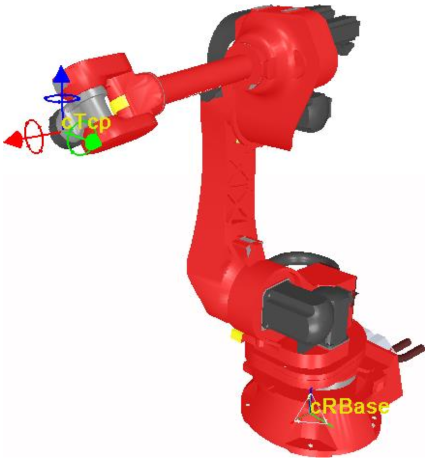
The TCP JOG coordinate system is activated after selection of the JOG mode and represents either

- the robot’s TCP by the button TCP TOOL,
- the robot’s Base by the button TCP BASE or
- world by the button TCP WORLD.

The following example shows 2 different JOG modes:



TCP Tool activated



TCP Base activated

Not displayed during simulation

During the simulation, the TCP JOG coordinate system is dimmed for a better overview.

Without inverse no TCP JOG coordinate system

If an inverse solution has not been assigned to a kinematic, e.g. positioners, feed units or individual kinematics or if these have not been selected from the reference kinematics, see page 28, the TCP JOG coordinate system is only displayed in a simplified manner.

Licensing variants are now thread-safe

EASY-ROB ™ supports several dongle variants for the USB connection as well as the computer-bound license (hardware number) and the EASY-ROB ™ license manager (floating licenses). These include the WibuKey dongle and CodeMeter CmStick from WIBU-SYSTEMS AG and the newly added MatrixLock dongle from TechnoData Interware GmbH.

WibuKey Dongle

CodeMeter, CmStick

MatrixLock-Dongle

Each of these dongles is represented by a unique ID (also referred to as hardware number or hardware ID), which is the basis for generating the licensing key.

This flexibility is especially important for our OEM customers, so the products EASY-ROB™ DLL Version and the EASY-ROB™ Robotics Simulation Kernel are compliant with the host application's licensing variants.

The dongle manufacturers provide for the control of the dongle an API, which is not necessarily Thread Safe.

To avoid errors while accessing the dongle from different threads, the dongle access is synchronized by Mutex objects and semaphores now.

Note:

The dongle access during the simulation is extremely rare, so there are no performance losses. Generally a dongle access takes place during the initialization and loading of new objects, for example robots and Robot Assemblies.

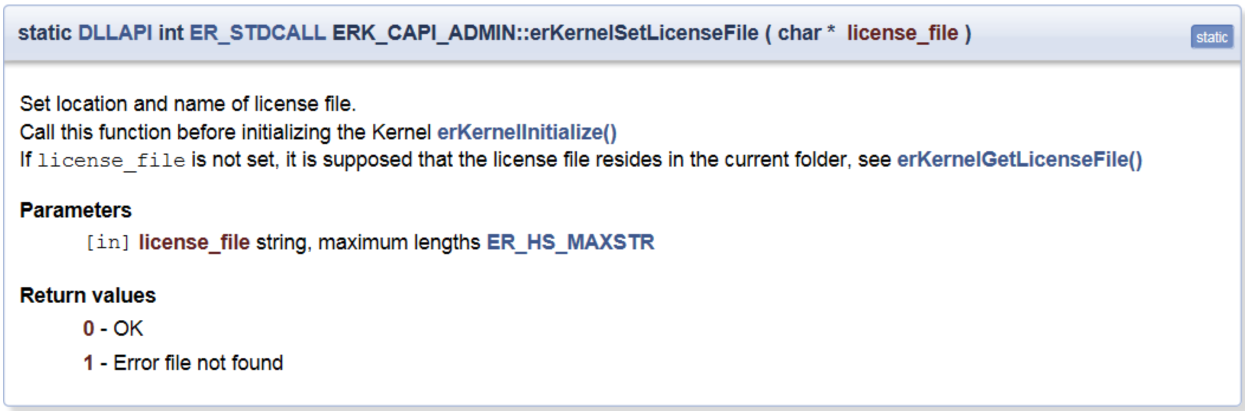
Tip:

New API function calls for DLL Version and Kernel allow the explicit dongle access to read the unique dongle ID, for license query and to determine the path for the license file "license.dat"; please see:

erKernelGetHardwareID(...), erKernelGetLicense(...), erKernelSetLicenseFile(...).

Robotics Simulation Kernel class ERK_CAPI

New methods have been added to the class **ERK_CAPI_ADMIN**, which includes the licensing, current hardware ID (dongle or computer-related) and licensed options.



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelSetLicenseFile (char * license_file) static

Set location and name of license file.
Call this function before initializing the Kernel erKernelInitialize()
If `license_file` is not set, it is supposed that the license file resides in the current folder, see erKernelGetLicenseFile()

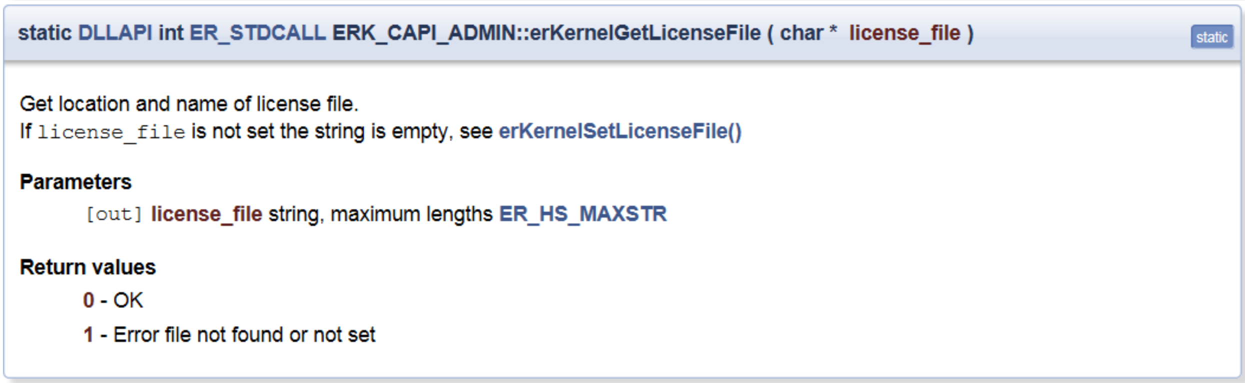
Parameters

`[in]` **license_file** string, maximum lengths **ER_HS_MAXSTR**

Return values

0 - OK

1 - Error file not found



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelGetLicenseFile (char * license_file) static

Get location and name of license file.
If `license_file` is not set the string is empty, see erKernelSetLicenseFile()

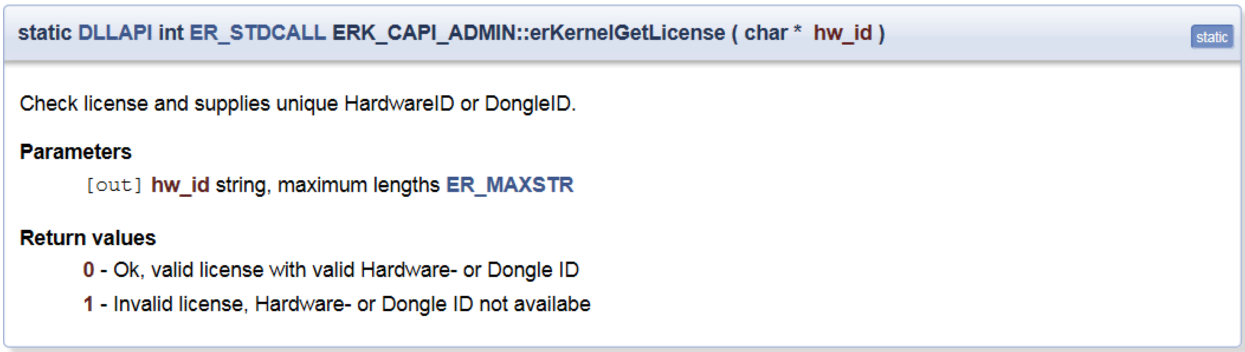
Parameters

`[out]` **license_file** string, maximum lengths **ER_HS_MAXSTR**

Return values

0 - OK

1 - Error file not found or not set



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelGetLicense (char * hw_id) static

Check license and supplies unique HardwareID or DongleID.

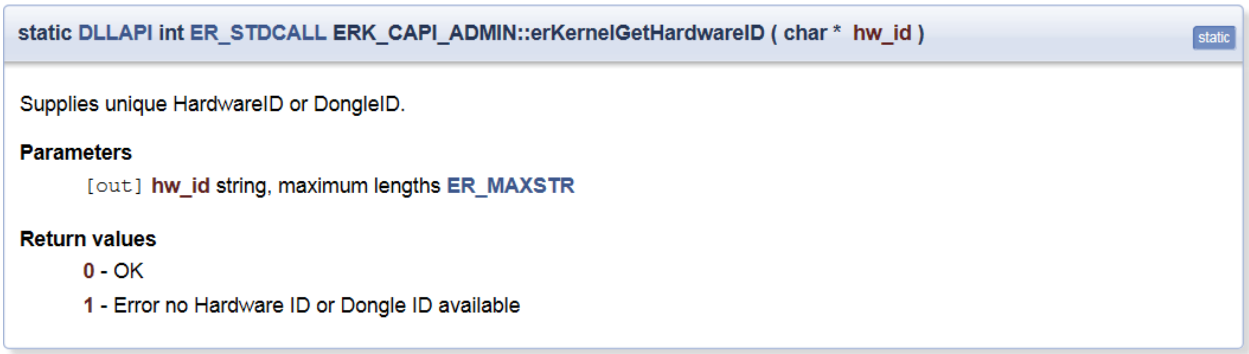
Parameters

`[out]` **hw_id** string, maximum lengths **ER_MAXSTR**

Return values

0 - Ok, valid license with valid Hardware- or Dongle ID

1 - Invalid license, Hardware- or Dongle ID not availabe



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelGetHardwareID (char * hw_id)

Supplies unique HardwareID or DongleID.

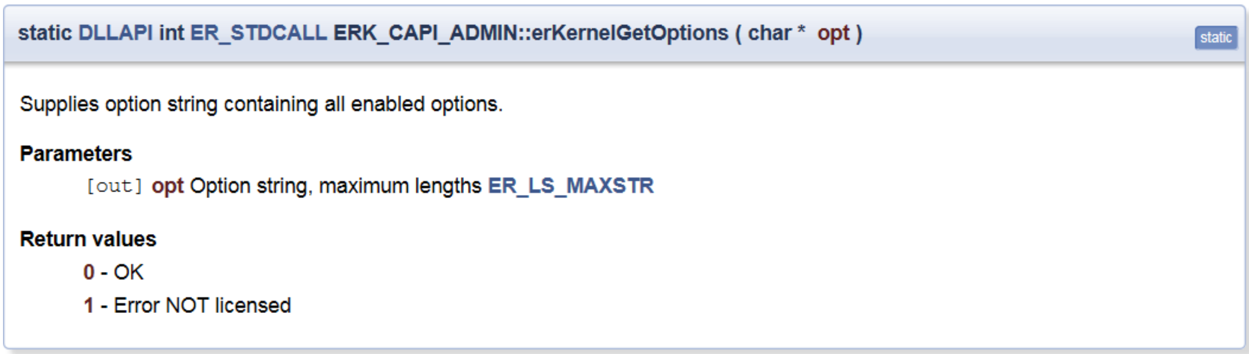
Parameters

[out] **hw_id** string, maximum lengths **ER_MAXSTR**

Return values

0 - OK

1 - Error no Hardware ID or Dongle ID available



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelGetOptions (char * opt)

Supplies option string containing all enabled options.

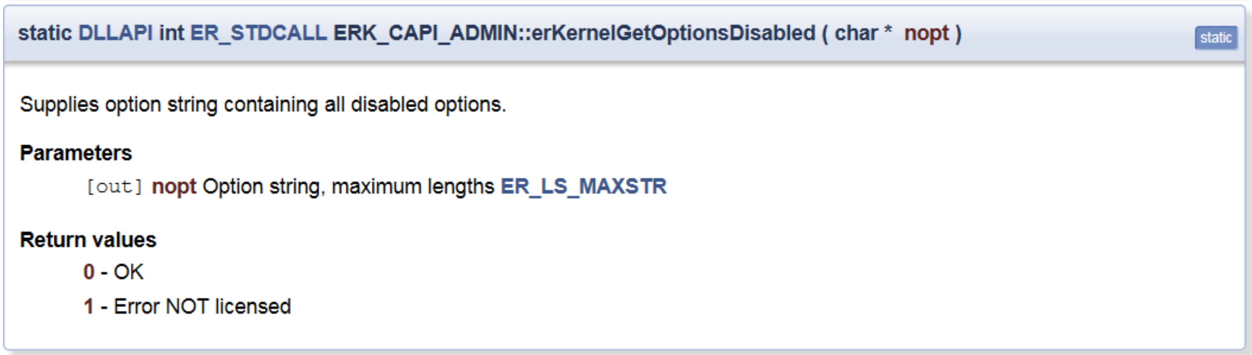
Parameters

[out] **opt** Option string, maximum lengths **ER_LS_MAXSTR**

Return values

0 - OK

1 - Error NOT licensed



static DLLAPI int ER_STDCALL ERK_CAPI_ADMIN::erKernelGetOptionsDisabled (char * nopt)

Supplies option string containing all disabled options.

Parameters

[out] **nopt** Option string, maximum lengths **ER_LS_MAXSTR**

Return values

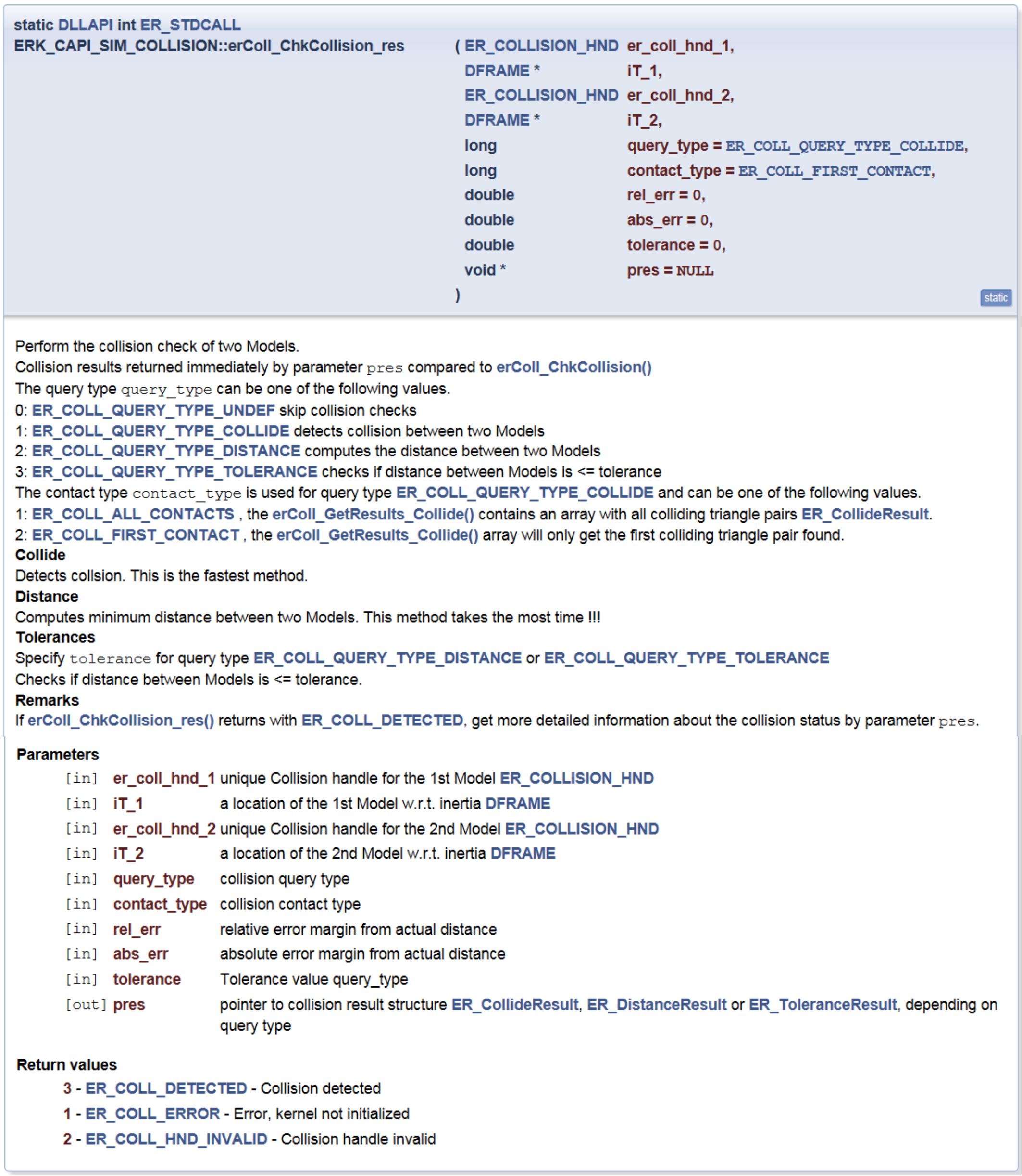
0 - OK

1 - Error NOT licensed

Robotics Simulation Kernel collision thread-safe

The Kernel supplies methods to test collisions between two triangulated 3D geometries. Class **ERK_CAPI_SIM_COLLISION** provides new methods, so that the collision test is Thread-safe.

In the old variant (not Thread-safe) the collision test was checked while calling method erColl_ChkCollision(...) and the result was retrieved with a further call of method erColl_GetResults_Collide(...).
For the new Thread-safe variant, the result is immediately delivered when erColl_ChkCollision_res (..., void * pres) is called. The memory for void * pres must be assigned in advance.



```
static DLLAPI int ER_STDCALL
ERK_CAPI_SIM_COLLISION::erColl_ChkCollision_res    ( ER_COLLISION_HND  er_coll_hnd_1,
                                                     DFRAME *          iT_1,
                                                     ER_COLLISION_HND  er_coll_hnd_2,
                                                     DFRAME *          iT_2,
                                                     long              query_type = ER_COLL_QUERY_TYPE_COLLIDE,
                                                     long              contact_type = ER_COLL_FIRST_CONTACT,
                                                     double            rel_err = 0,
                                                     double            abs_err = 0,
                                                     double            tolerance = 0,
                                                     void *            pres = NULL
                                                   )
```

Perform the collision check of two Models.
Collision results returned immediately by parameter `pres` compared to **erColl_ChkCollision()**
The query type `query_type` can be one of the following values.
0: **ER_COLL_QUERY_TYPE_UNDEF** skip collision checks
1: **ER_COLL_QUERY_TYPE_COLLIDE** detects collision between two Models
2: **ER_COLL_QUERY_TYPE_DISTANCE** computes the distance between two Models
3: **ER_COLL_QUERY_TYPE_TOLERANCE** checks if distance between Models is <= tolerance
The contact type `contact_type` is used for query type **ER_COLL_QUERY_TYPE_COLLIDE** and can be one of the following values.
1: **ER_COLL_ALL_CONTACTS** , the **erColl_GetResults_Collide()** contains an array with all colliding triangle pairs **ER_CollideResult**.
2: **ER_COLL_FIRST_CONTACT** , the **erColl_GetResults_Collide()** array will only get the first colliding triangle pair found.

Collide
Detects collsion. This is the fastest method.

Distance
Computes minimum distance between two Models. This method takes the most time !!!

Tolerances
Specify `tolerance` for query type **ER_COLL_QUERY_TYPE_DISTANCE** or **ER_COLL_QUERY_TYPE_TOLERANCE**
Checks if distance between Models is <= tolerance.

Remarks
If **erColl_ChkCollision_res()** returns with **ER_COLL_DETECTED**, get more detailed information about the collision status by parameter `pres`.

Parameters

[in]	**er_coll_hnd_1**	unique Collision handle for the 1st Model **ER_COLLISION_HND**
[in]	**iT_1**	a location of the 1st Model w.r.t. inertia **DFRAME**
[in]	**er_coll_hnd_2**	unique Collision handle for the 2nd Model **ER_COLLISION_HND**
[in]	**iT_2**	a location of the 2nd Model w.r.t. inertia **DFRAME**
[in]	**query_type**	collision query type
[in]	**contact_type**	collision contact type
[in]	**rel_err**	relative error margin from actual distance
[in]	**abs_err**	absolute error margin from actual distance
[in]	**tolerance**	Tolerance value query_type
[out]	**pres**	pointer to collision result structure **ER_CollideResult**, **ER_DistanceResult** or **ER_ToleranceResult**, depending on query type

Return values

3 - **ER_COLL_DETECTED** - Collision detected
1 - **ER_COLL_ERROR** - Error, kernel not initialized
2 - **ER_COLL_HND_INVALID** - Collision handle invalid

Class ER_CAPI EASY-ROB™ DLL- and Multi-Robot Version

New methods have been added to class **ER_CAPI**,
see header files: er_CAPI.h und er_CAPI_Types.h

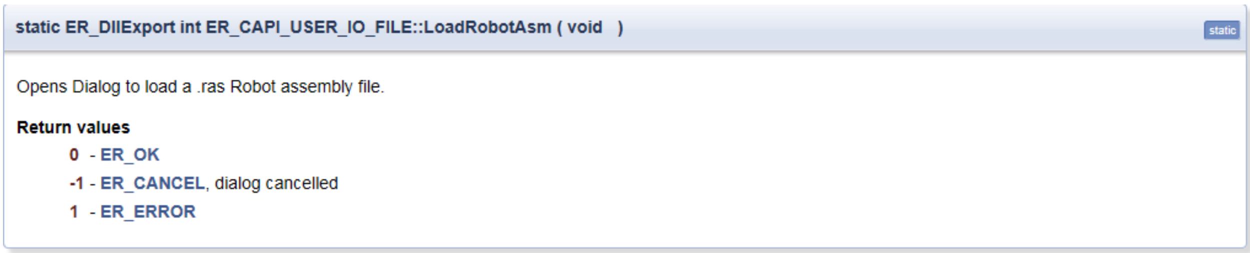
New constants in **er_CAPI_Types.h**

```
const int KIN_USER_DATA       = 12; ///< Number of Sets of Kin User Data
const int KIN_PRGS            = 12; ///< max. number of available Programs

// ER_CAPI_SIM_ERPL::erc_onoff(...)
const int ERC_ONOFF_DISABLE   = 0; ///< disable flag
const int ERC_ONOFF_ENABLE    = 1; ///< enable flag
const int ERC_ONOFF_STATUS    = 2; ///< request current flag status
const int ERC_VIEW_CHOREOGRAPHY = 53; ///< Enable/disable View choreography
const int ERC_3D_PDF_EXPORT_ERCL = 54;///< Enable/disable 3D Pdf Export ERCL
const int ERC_TCP_JOG_COORSYS  = 55; ///< Enable/disable Show tcp jog coorsys
          for cRobot with inverse kinematics
const int ERC_DISPLAY_CTAG = 56; ///< Enable/disable Display selected Tag

// AuxUpdate(int idx) feedback
const int AUX_UPDATE_IDX_LOAD_ROBOT_ASM = 56;///< Robot Assembly is loaded
const int AUX_UPDATE_IDX_SAVE_ROBOT_ASM = 57;///< Robot Assembly saved (.ras)
const int AUX_UPDATE_IDX_SWE_PASSIVE_EXCEEDED= 58; ///< At least one passive
          axis exceeds its maximum allowed travel ranges

// 3D PDF Export, Layout definitions
const int _3DPDF_EXPORT_RESET_LAYOUT = 0x100000; ///< 3D_Pdf_Export reset
          current 3D_Pdf Layout settings, set to default data
const int _3DPDF_EXPORT_LOAD_LAYOUT_FILE = 0x200000; ///< 3D_Pdf_Export load
          3D_Pdf Layout from *.3pl file
const int _3DPDF_EXPORT_SAVE_LAYOUT_FILE = 0x400000; ///< 3D_Pdf_Export load
          3D_Pdf Layout to *.3pl file
```

New and extended methods in er_CAPI.h

Load and save Robot Assemblies

`int ER_CAPI_USER_IO_FILE::LoadRobotAsm ()`



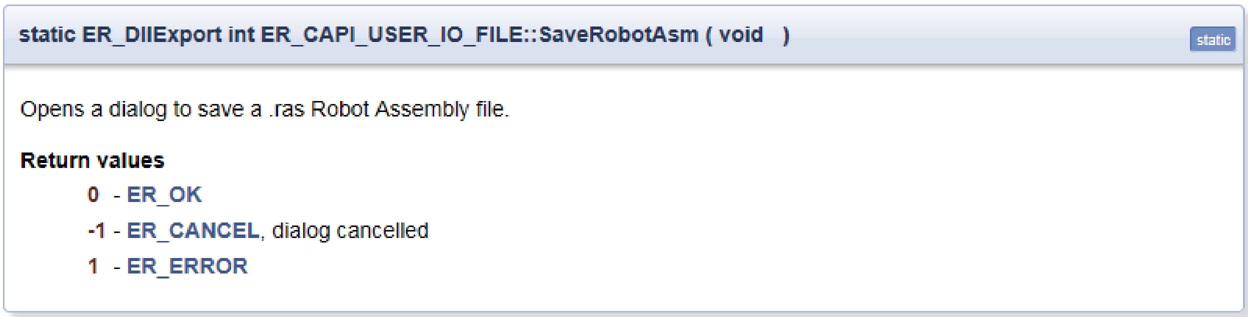
static ER_DllExport int ER_CAPI_USER_IO_FILE::LoadRobotAsm (void) static

Opens Dialog to load a .ras Robot assembly file.

Return values

0 - **ER_OK**
-1 - **ER_CANCEL**, dialog cancelled
1 - **ER_ERROR**

`int ER_CAPI_USER_IO_FILE::SaveRobotAsm ()`
`int ER_CAPI_USER_IO_FILE::SaveRobotAsm_ext ()`

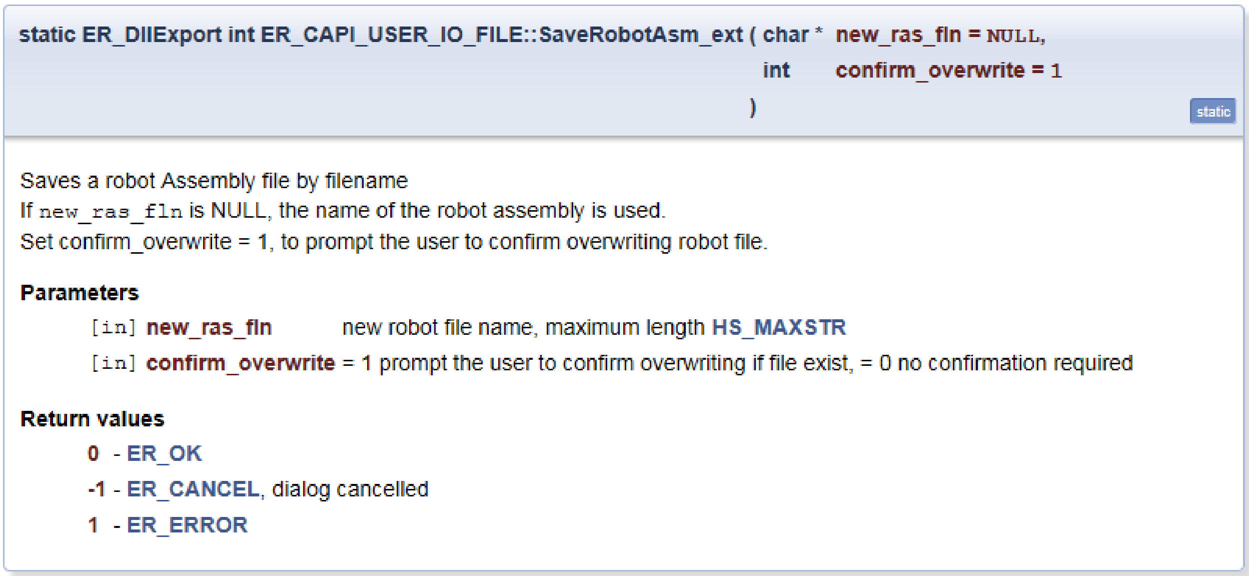


static ER_DllExport int ER_CAPI_USER_IO_FILE::SaveRobotAsm (void) static

Opens a dialog to save a .ras Robot Assembly file.

Return values

0 - **ER_OK**
-1 - **ER_CANCEL**, dialog cancelled
1 - **ER_ERROR**



```
static ER_DllExport int ER_CAPI_USER_IO_FILE::SaveRobotAsm_ext ( char * new_ras_fln = NULL,
                                                                 int    confirm_overwrite = 1
                                                               )
```
static

Saves a robot Assembly file by filename
If `new_ras_fln` is NULL, the name of the robot assembly is used.
Set confirm_overwrite = 1, to prompt the user to confirm overwriting robot file.

Parameters

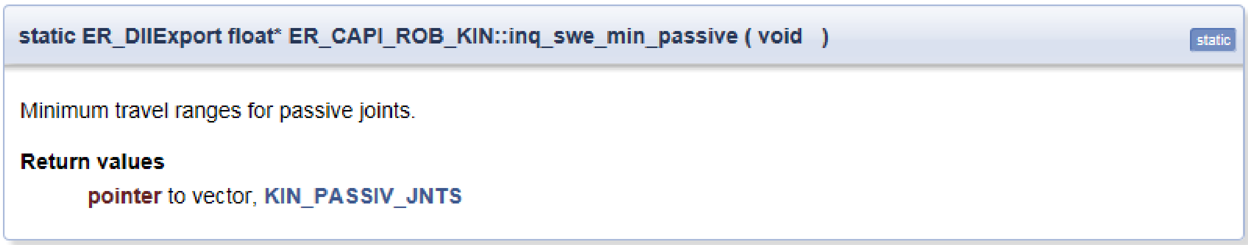
[in] **new_ras_fln** new robot file name, maximum length **HS_MAXSTR**
[in] **confirm_overwrite** = 1 prompt the user to confirm overwriting if file exist, = 0 no confirmation required

Return values

0 - **ER_OK**
-1 - **ER_CANCEL**, dialog cancelled
1 - **ER_ERROR**

Travel ranges for passive joints

float *ER_CAPI_ROB_KIN::inq_swe_max_passive ()



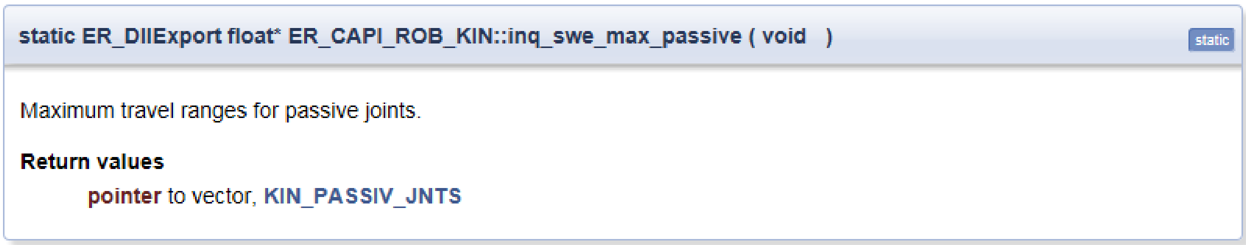
static ER_DllExport float* ER_CAPI_ROB_KIN::inq_swe_min_passive (void) static

Minimum travel ranges for passive joints.

Return values

pointer to vector, **KIN_PASSIV_JNTS**

float *ER_CAPI_ROB_KIN::inq_swe_min_passive ()



static ER_DllExport float* ER_CAPI_ROB_KIN::inq_swe_max_passive (void) static

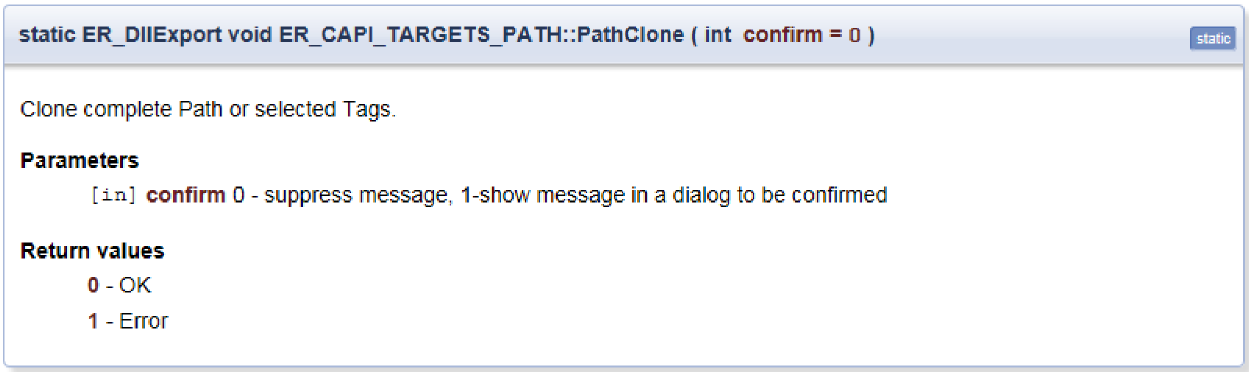
Maximum travel ranges for passive joints.

Return values

pointer to vector, **KIN_PASSIV_JNTS**

Clone paths and Tags

```
void ER_CAPI_TARGETS_PATH::PathClone(int confirm=0)
```



static ER_DllExport void ER_CAPI_TARGETS_PATH::PathClone (int confirm = 0) static

Clone complete Path or selected Tags.

Parameters

[in] **confirm** 0 - suppress message, 1-show message in a dialog to be confirmed

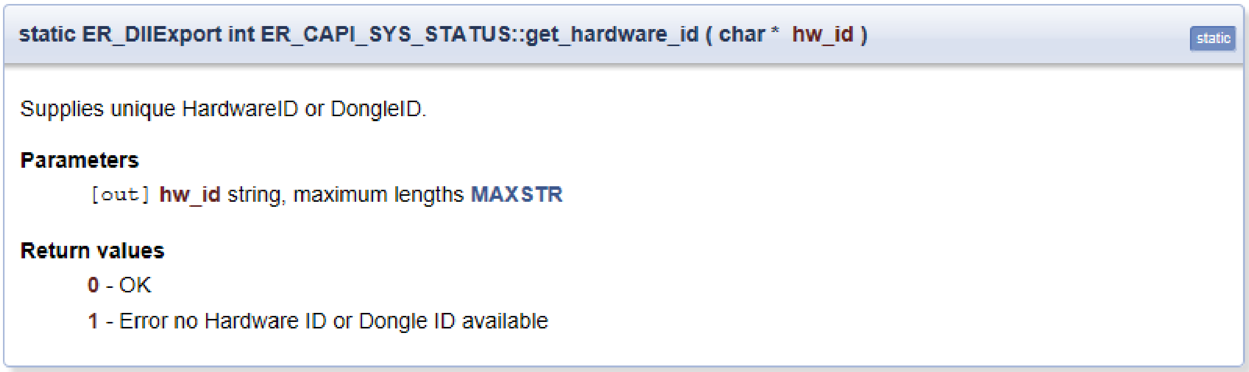
Return values

0 - OK

1 - Error

Hardware ID

```
int ER_CAPI_SYS_STATUS::get_hardware_id (char *hw_id)
```



static ER_DllExport int ER_CAPI_SYS_STATUS::get_hardware_id (char * hw_id) static

Supplies unique HardwareID or DongleID.

Parameters

[out] **hw_id** string, maximum lengths **MAXSTR**

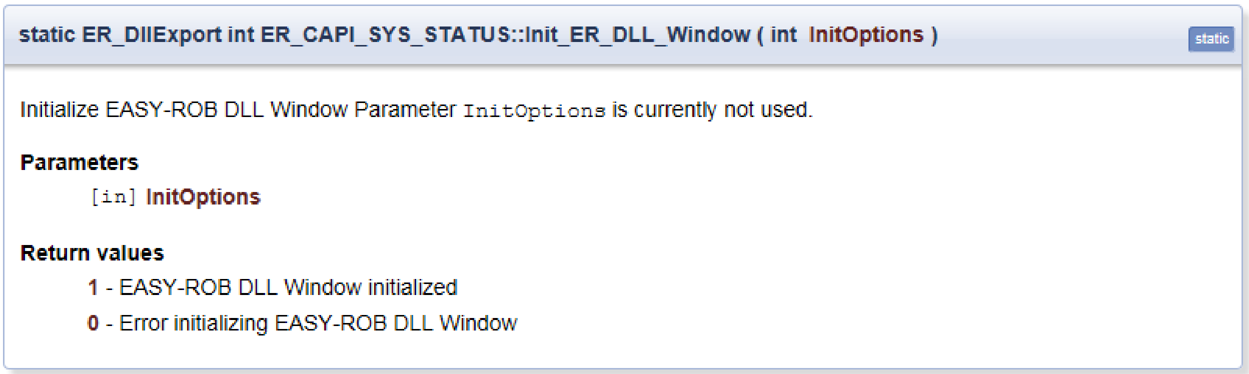
Return values

0 - OK

1 - Error no Hardware ID or Dongle ID available

Start/Initialize the EASY-ROB™ DLL Window

```
int ER_CAPI_SYS_STATUS::Init_ER_DLL_Window (int InitOptions)
Ausnahme:
         Nur EASY-ROB™ DLL Version
```



static ER_DllExport int ER_CAPI_SYS_STATUS::Init_ER_DLL_Window (int InitOptions) static

Initialize EASY-ROB DLL Window Parameter `InitOptions` is currently not used.

Parameters

[in] **InitOptions**

Return values

1 - EASY-ROB DLL Window initialized

0 - Error initializing EASY-ROB DLL Window

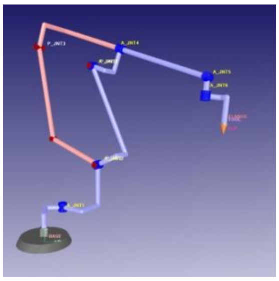
RefKin ID – Kinematic structures at a glance

For each kinematic model for which an analytical inverse transformation (IK) exists in EASY-ROB ™, you will find the Inverse Kin-ID and an associated Sub-ID. Together they result in the reference kinematic ID = RefKin ID.
For example, in EASY-ROB™ Kinematics, you can look up the appropriate IDs in "Kin-IDs-Parameters_ENG.pdf" and then assign them so that your model has a correct analytic inverse transformation.

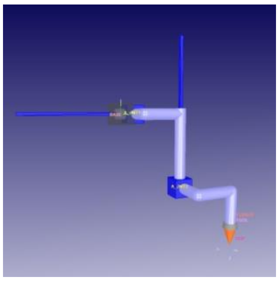
Overview of kinematic models

The following kinematic structures are described as RefKins:

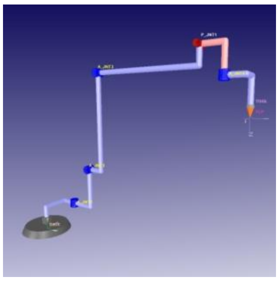
- Vertical articulated robot with central wrist (if necessary with coupling and backlink; 6 axes)
- Jet robots (if necessary with coupling and backlink; 6 axes)
- Palletizing robots (if necessary with backlink; 4 axes)
- SCARA kinematic (4 or 6 axes)
- Portal robots (1 up to 6 axes)



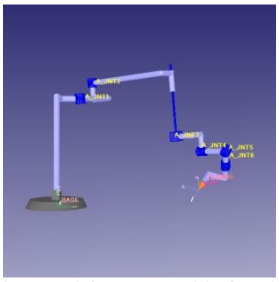
V-articulated (BL) ID 111, SID 0



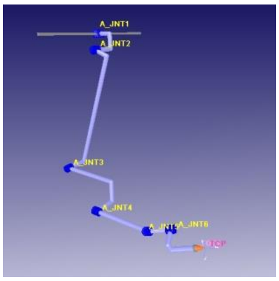
2-axes-portal ID 133, SID 13,23



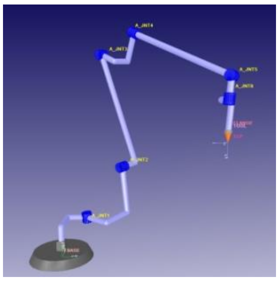
Palletizer ID 129, SID 0



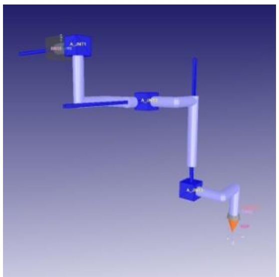
6-axes-SCARA ID 139, SID 0



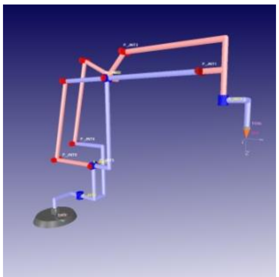
Jet robot ID 127, SID 0



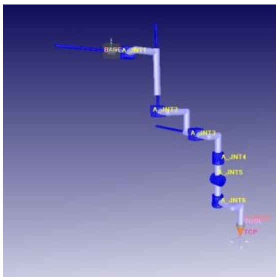
V-articulated ID 110, SID 1



4-axes-portal ID 134, SID 0,123



Palletizer ID 129, SID 1



6-axes-portal ID 136, SID 0

Examples of different reference kinematics as idealized robot models

Examples of Kin-ID and Sub-ID

Thus, a **palletizing robot** has the following IDs:

- Kin-ID 129 – Can also be understood as the main class
- Sub-ID 0 – Is a subgroup for further classification

However, a **palletizer with backlink** (BL) has the following IDs:

- Kin-ID 129 – Like the normal palletizing robot
- Sub-ID 1 – Distinction due to backlink kinematic.

Transformation table

Using this table, you can see which transformation lengths and rotations are allowed for each active axis to the next active axis "Geometric Data to next", so that the inverse transform (IK) with the specified Kin-ID and Sub-ID can calculate a result for each possible configuration. Green hooks indicate that a corresponding transformation is allowed.

Example of a transformation table:

"Geometric Data to next"	x	y	z	Rx	Ry	Rz
Robot Base → 1st Joint (aJnt1)	✓	✓	✓	✓	✓	✓
aJnt1 → aJnt2	✓	✓	✓	✗	✗	✗
aJnt2 → aJnt3	✓	✓	✓	✗	✗	✗
aJnt3 → aJnt4	✓	✓	✓	✗	✗	✗
aJnt4 → aJnt5	✓	✗	✗	✗	✗	✗
aJnt5 → aJnt6	✓	✗	✗	✗	✗	✗
aJnt6 → Tip (Flange)	✓	✓	✓	✓	✓	✓
Tip (Flange)→ TCP (Tool data)	✓	✓	✓	✓	✓	✓

Transformation table of a vertical articulated robot with central wrist: Kin-ID 110, Sub-ID 0

Rob-files as an example of Kin-IDs

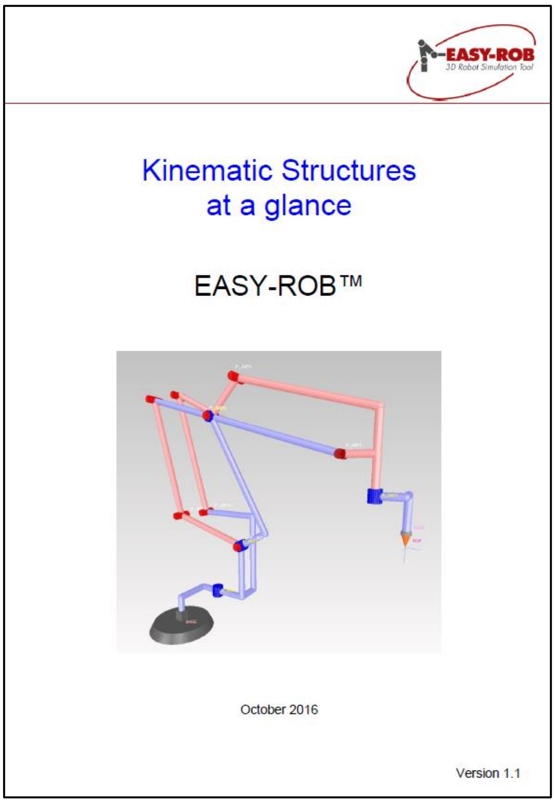
For each kinematic example you will find a corresponding ROB-file in directory "KinIDs". The representation of these example models was reduced to the essential. Thus, active rotation axes are visualized as blue cylinders and active translation axes as blue cubes. Passive axes, on the other hand, are correspondingly red. The connection between the individual axes and thus the kinematic chain is represented by thin cylinders.

In the document "TrainLib-Tutorial", chapter 6 "Setting up kinematics", you will find a detailed step-by-step guide of how to create simple kinematics from standard CAD components on your own.

"Kin-IDs-Parameters_ENG.pdf" as a further document

For further information please refer to the document "Kin-IDs-Parameters_ENG.pdf". This document provides a compact, but also a detailed overview of the most important kinematic structures in EASY-ROB™.

In addition, a description of the "Numerical Solution" Kin-ID = 100, Sub-ID = 0/1 can be found on the last page of the document. This provides a solution for calculating the inverse of kinematic models for which no analytical solution is available.



EASY-ROB

3D Robot Simulation Tool

Kinematic Structures
at a glance

EASY-ROB™

October 2016

Version 1.1

Kin-IDs-Parameters_ENG.pdf

Complete robot libraries

EASY-ROB™ provides complete libraries for the integration of all major types of robots of the market. These include ABB, b + m, Comau, Denso, Eisenmann, Fanuc, Guedel, igm, Kawasaki, KUKA, Mitsubishi, OTC-Daihen, Rice, Stäubli, Tricept, Unimation, Universal Robots and Yaskawa.

The robot libraries of ABB, KUKA, Comau, Fanuc, Stäubli and Yaskawa are almost complete and constantly maintained by us.

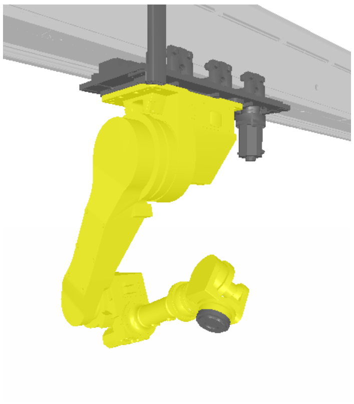
Currently more than 1000 robots, positioners and external tracking axes are available of various manufacturers.

Further information:

http://www.easy-rob.com/en/product/extensions/robot-libraries.html

New models of Fanuc for example M-20iA-35M and R-2000iB-200T portal, or from Stäubli, e.g. TX-340SH join the numerous robots.

Fanuc M-20IA-35M



Fanuc R-2000iB-200T-Portal

Stäubli TX-340SH

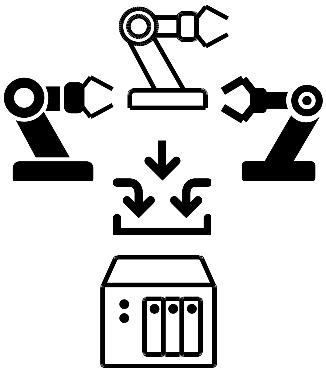
Important notice:

Non-existent robots, handling systems, machines, tools or even special kinematics can be easily and quickly "reconstructed virtually" in EASY-ROB™.

Multi-Robot – Multi-Kin and Multi-Program are now united!

With increasing simulation complexity, a separation of the Multi-Kin and Multi-Program version was hardly possible. EASY-ROB therefore decided to combine both versions in the new Multi-Robot version as a powerful simulation tool for multi-robot applications.

Options overview



The Multi-Robot (EXE, DLL) is based on the Single-Robot version.

- The number of robots*) and programs for each work cell is not limited. The number of kinematics like grippers, turntables and/or conveyors is not limited in loading, linking and simulating.
- With the Multi-Robot version you can move several robots and kinematics synchronized and simultaneous. The different programs communicate by I/O-signals.

  *) Robots have at least 4 axes and can approach Cartesian target points (Tags).

Update possible

For customers of older versions it is possible to purchase an update. Please contact our sales department at +49 6192 921 70 79 or sales@easy-rob.com and provide your hardware / dongle number and the version number of your license. To determine your hardware number (HwNr): Start EASY-ROB ™ and open the menu

- „? → License Info“.

If necessary, repeat this step for your additional licenses..

New ERCL commands

ERCL - 3D-PDF Export Layout definition commands

ERC _3D_PDF_EXPORT RESET_LAYOUT

3D-Pdf Layout reset to default values

ERC _3D_PDF_EXPORT LOAD_LAYOUT filename.3pl

Load 3D-Pdf Layout from .3pl file "filename.3pl"

ERC _3D_PDF_EXPORT SAVE_LAYOUT filename.3pl

Save 3D-Pdf Layout to .3pl file "filename.3pl"

ERC _3D_PDF_EXPORT PAGE_AREA width height

Set Page Area for 3D-Pdf Layout
Parameter: width height, see image page 14

ERC _3D_PDF_EXPORT VIEWPORT_AREA left(x) upper(y) width height

Set Viewport Area for 3D-Pdf Layout;
Parameter: left(x) upper(y) width height, see image page 14

ERC _3D_PDF_EXPORT CONTROL_AREA left(x) upper(y) width height

Set Control Area for 3D-Pdf Layout;
Parameter: left(x) upper(y) width height, see image page 14

ERC _3D_PDF_EXPORT LABEL_FONT_SIZE size

Set Label Font Size for Control Area of 3D-Pdf Layout;
Parameter: size in range [1-50], default value: size = 10

Add images to a 3D-PDF Layout

The path must be set in advance with command IMAGE_PATH. Paths and images information must be set in "".

ERC _3D_PDF_EXPORT IMAGE_PATH "path"

Path information to load images with ADD_IMAGE

ERC _3D_PDF_EXPORT IMAGE_PATH USERPROFILE

Key "USERPROFILE"
Set the path to load images with ADD_IMAGE, i.e. c:\Users\MyLoginName

ERC _3D_PDF_EXPORT IMAGE_PATH WORKING_DIRECTORY

Key "WORKING_DIRECTORY" ,
current working directory determine the path to load images with ADD_IMAGE

ERC _3D_PDF_EXPORT IMAGE_PATH ""

Same as Key "WORKING_DIRECTORY" current working directory

ERC _3D_PDF_EXPORT IMAGE_PATH 3PL_FILE_FOLDER

Key "3PL_FILE_FOLDER "
The path defined in 3D-Pdf Layout file .3pl determine the path to load images with ADD_IMAGE.

ERC _3D_PDF_EXPORT ADD_IMAGE "filename.jpg" left(x) upper(y) width height Scaling

Adds a jpg image to a defined position
Parameter 1: "Image file name"
Parameter 2-5: left(x) upper(y) width height
Parameter 6: Scaling is one of ISO_Stretch = 0 or ISO_CenterFit = 1

Contact data

EASY-ROB 3D Robot Simulation Tool

Address:	Hauptstr. 42 65719 Hofheim am Taunus Germany
Owner:	Stefan Anton
Phone:	+49 (0) 6192 921 70 77
FAX:	+49 (0) 6192 921 70 66
Email:	contact@easy-rob.com sales@easy-rob.com
Web:	www.easy-rob.com
Online Shop:	http://www.easy-rob.com/produkt/shop.html/

EASY-ROB customer area

Content:	Program updates and robot libraries
Web:	http://www.easy-rob.com/en/special/customer-area/

Log in details:

User name:	customer
Password:	**********

Own notes