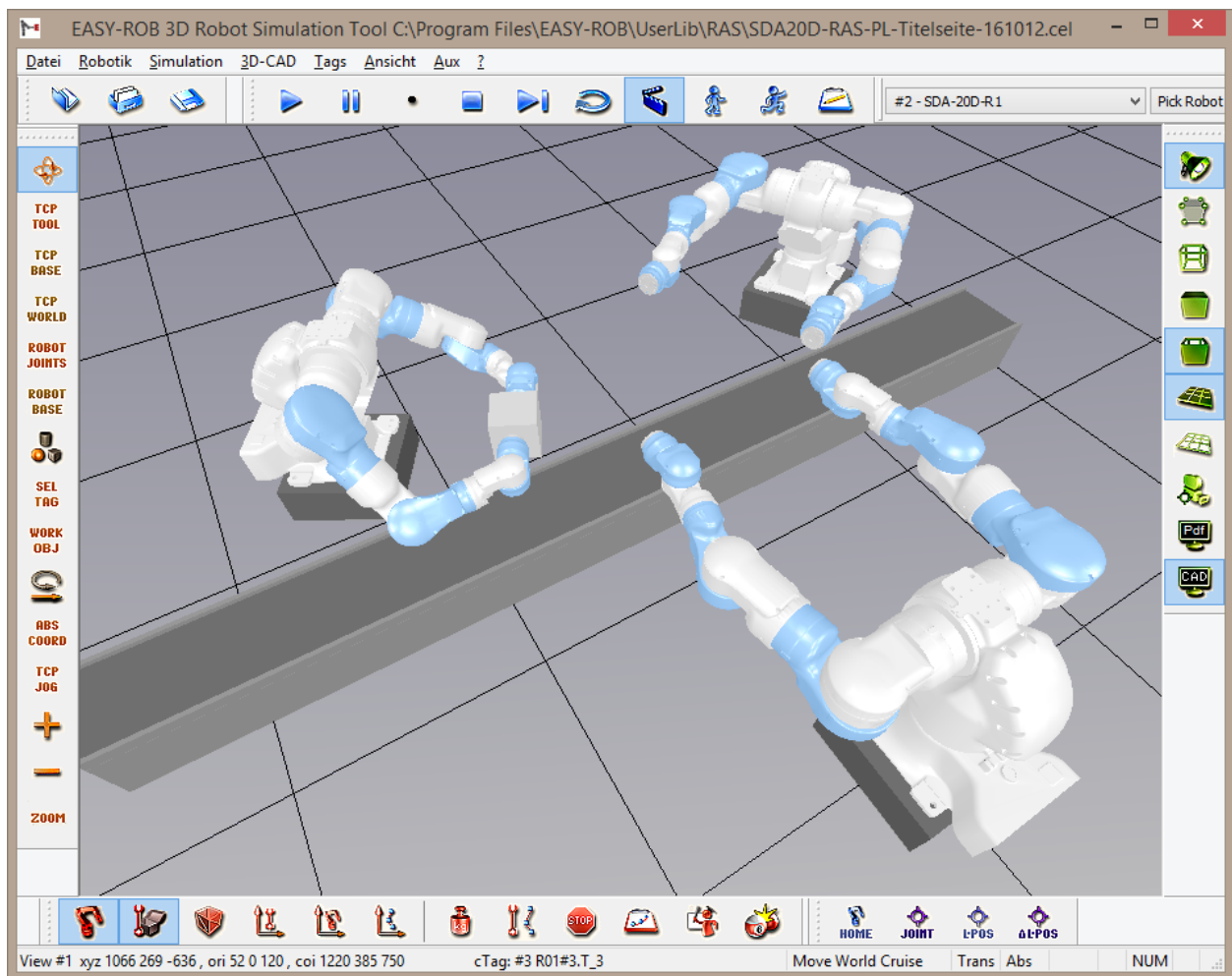


# Die neue Version

## EASY-ROB™ V7.0



Oktober 2016

Version 1.0



# EASY-ROB™

## Inhaltsverzeichnis

<b>EASY-ROB™ V7.0 Update .....</b>	<b>4</b>
<b>Sprachumstellung (ENGLISCH / DEUTSCH / CHINESISCH) .....</b>	<b>6</b>
Wie kann die Sprache des GUI umgestellt werden? .....	7
Lokalisationsdatei "easy-rob-localizationx64.ini" m Überblick .....	8
<b>Robot Assembly .....</b>	<b>9</b>
Einfach speichern und laden wie Roboter .....	10
Modularer Aufbau von Produktionslinien möglich .....	11
<b>Layout Funktionen für 3D-PDF Export mit Animation .....</b>	<b>12</b>
Layout Funktionen für erfolgreichen Wiedererkennungswert .....	12
Erneute Benutzung der *.3pl-Datei erwünscht .....	12
Kontrolldatei "3D-PDF-Layout.3pl" im Überblick .....	13
Funktionen und Parameter des 3D-PDF Layouts .....	14
ERCL Befehle für Layout Funktionen des 3D-PDF Exports .....	15
<b>TCP JOG Koordinatensystem .....</b>	<b>18</b>
Aktivierung abhängig von jeweiligen JOG Modus .....	19
<b>Lizenzierungs-Varianten nun Thread-Safe .....</b>	<b>20</b>
<b>Robotics Simulation Kernel Klasse ERK_CAPI .....</b>	<b>21</b>
<b>Robotics Simulation Kernel Kollision Thread-Safe .....</b>	<b>23</b>
<b>Klasse ER_CAPI EASY-ROB™ DLL- und Multi-Robot Version .....</b>	<b>24</b>
Neue Konstanten in er_CAPI_Types.h .....	24
Neue und erweiterte Methoden in er_CAPI.h .....	25
<b>RefKin ID – Kinematische Strukturen im Überblick .....</b>	<b>28</b>
Übersicht der kinematischen Modelle .....	28
Beispiel Kin-ID und Sub-ID .....	29
Transformations-Tabelle .....	29
<b>Vollständige Roboterbibliotheken .....</b>	<b>31</b>
<b>Multi-Robot – Multi-Kin und Multi-Program jetzt vereint! .....</b>	<b>32</b>
Übersicht Optionen .....	32
<b>Neue ERCL-Befehle .....</b>	<b>33</b>
ERCL - 3D-PDF Export Layout Definition Kommandos .....	33
Bilder zum 3D-PDF Layout hinzufügen .....	34
<b>Kontakt .....</b>	<b>35</b>
<b>Eigene Notizen .....</b>	<b>36</b>

## EASY-ROB™ V7.0 Update

Hallo liebe EASY-ROB™ Community!

Endlich ist es so weit –

EASY-ROB™ Version 7.0 wird vorgestellt und folgende Highlights bietet die neue Version in der Schnellübersicht:

- **Sprachumstellung**  
Neben der englischen Sprache als Standardeinstellung kann das GUI nun in Deutsch und Chinesisch dargestellt werden. Über die „easy-rob-localizationx64.ini“ kann die Sprache schnell und bequem umgeschaltet werden.
- **Robot Assembly \*.ras**  
Komplexe kinematische Robotermodelle lassen sich nun als Robot Assembly einfach und bequem erstellen und in Arbeitszellen \*.cel-Dateien beliebig oft verwenden. Damit lassen sich u.a. auch Dual-Arm Roboter wie der Motoman/Yaskawa SDA20D inkl. Programmen, Pfade und Tags modular verwenden.
- **Layout Funktionen für 3D-PDF Export**  
Erstmalig gibt es die Möglichkeit ein personalisiertes Layout für das 3D-PDF zu benutzen. Das Layout des 3D-PDF Exports kann nun durch eigenen Bilder z.B. Logos und statische Informationen zur Beschriftung ergänzt werden. Damit erstellen Sie für Simulationsergebnisse Ihrer Kunden individualisierte 3D-PDF Dateien zwecks besserer Identifizierung etc.
- **TCP JOG Koordinatensystem**  
Damit man stets die richtige Richtung im Auge behält ist nun während des kartesischen oder achsweisen Joggens am Tip des Roboters ein schickes Koordinatensystem zu sehen, welches das Flansch Koordinatensystem verdeutlicht.
- **Dongle Zugriff nun Thread-Safe**  
Der Dongle-Zugriff wurde durch Mutex-Objekte und Semaphoren synchronisiert, so dass Zugriffe aus Threads sicher erfolgen, was durch die API der Hersteller nur bedingt gegeben ist.
- **Kollisionstest nun Thread-Safe im Robotics Kernel**  
Die Klasse ERK\_CAPI\_SIM\_COLLISION liefert neue Methoden, so dass der Kollisionstest Thread-Safe erfolgt.
- **Neue Robotermodelle**  
Die Roboter Bibliothek inkl. Verfahrachsen wächst dank Ihrer Anregung stetig weiter und Modelle der Hersteller Motoman/Yaskawa, Fanuc, KUKA, ABB, Stäubli, Comau und UniversalRobots sind dazugekommen.
- **Multi-Robot**  
Mit zunehmender Simulationskomplexität war eine Trennung der Multi-Kin und Multi-Program Version kaum noch möglich. EASY-ROB entschloss sich deshalb, beide Versionen in der Multi-Roboter Version als mächtiges Simulationstool für Mehr-Roboter Anwendungen zu vereinen.

In eigener Sache teilen wir Ihnen folgendes mit:

Aufgrund der verschwindend kleinen Nachfrage nach x86 Anwendungen von EASY-ROB™ wurde mit Version 7.0 die 32bit Entwicklung eingestellt. EASY-ROB kann sich nun mit seiner gesamten Entwicklungsleistung auf die Weiterentwicklung der 64bit Version konzentrieren. Sollten Sie dennoch einen x86 Anwendungsfall haben, nehmen Sie bitte mit unserem Vertrieb Kontakt auf.

Ab sofort steht allen Kunden mit einer gültigen v7.0 Lizenz oder einem Softwarepflegevertrag die neue EASY-ROB™ Version 7.0 kostenfrei zur Verfügung.

Für Kunden älterer Versionen besteht die Möglichkeit ein Update zu erwerben. Nehmen Sie dazu bitte mit unserem Vertrieb unter +49 6192 921 70 79 oder [sales@easy-rob.com](mailto:sales@easy-rob.com) Kontakt auf und geben Ihre Hardware-/Dongle-Number und die Versionsnummer Ihrer Lizenz an. Ihre Hardware Number (HwNr) finden Sie folgt: Starten Sie EASY-ROB™ und öffnen das Menü

- „? → License Info“.

Wiederholen Sie ggfs. diesen Schritt für Ihre weiteren Lizenzen.

Für Ihre Anregungen und Verbesserungsvorschläge bedanken wir uns schon jetzt bei Ihnen.

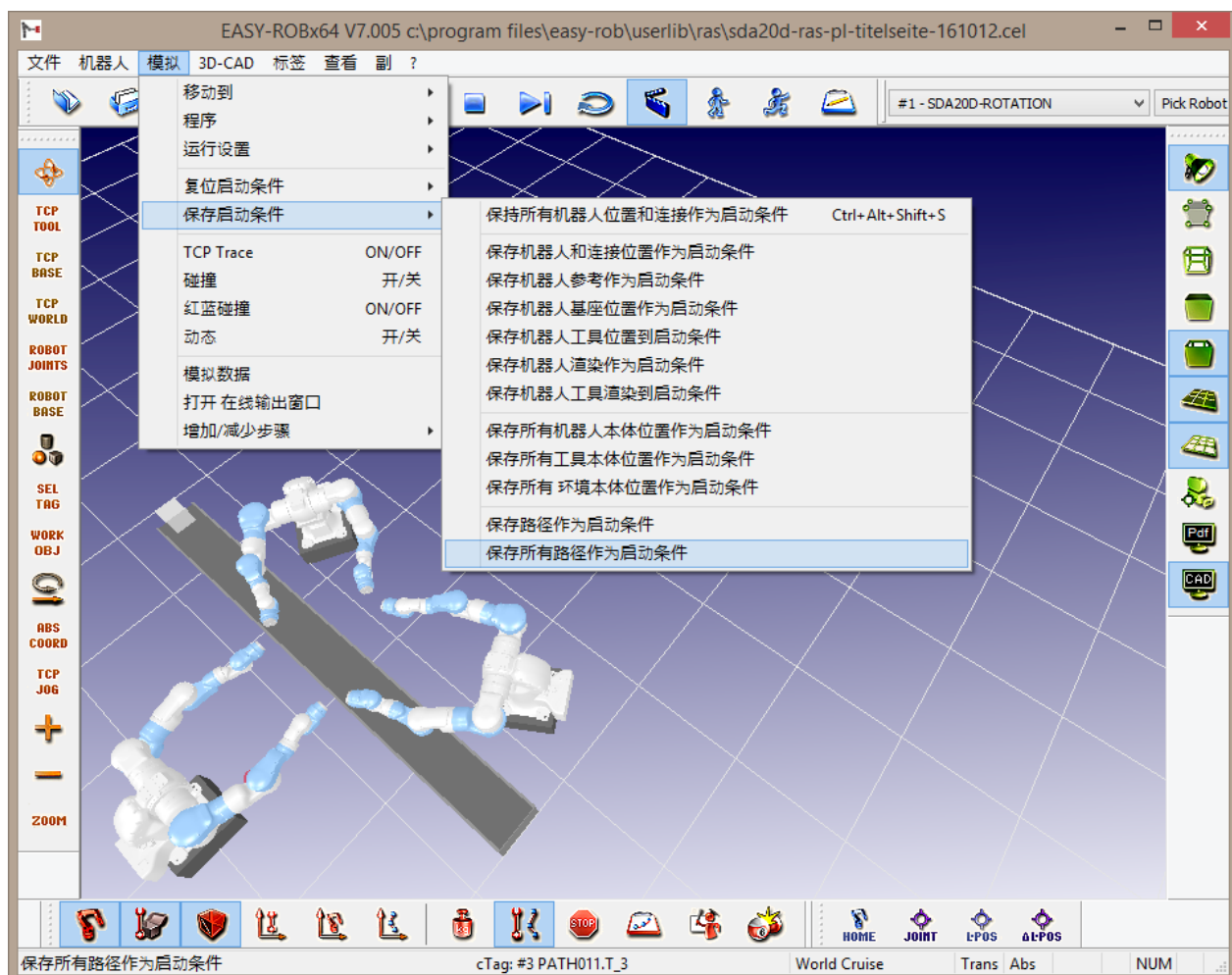
Vielen Dank

EASY-ROB  
3D Robot Simulation Tool

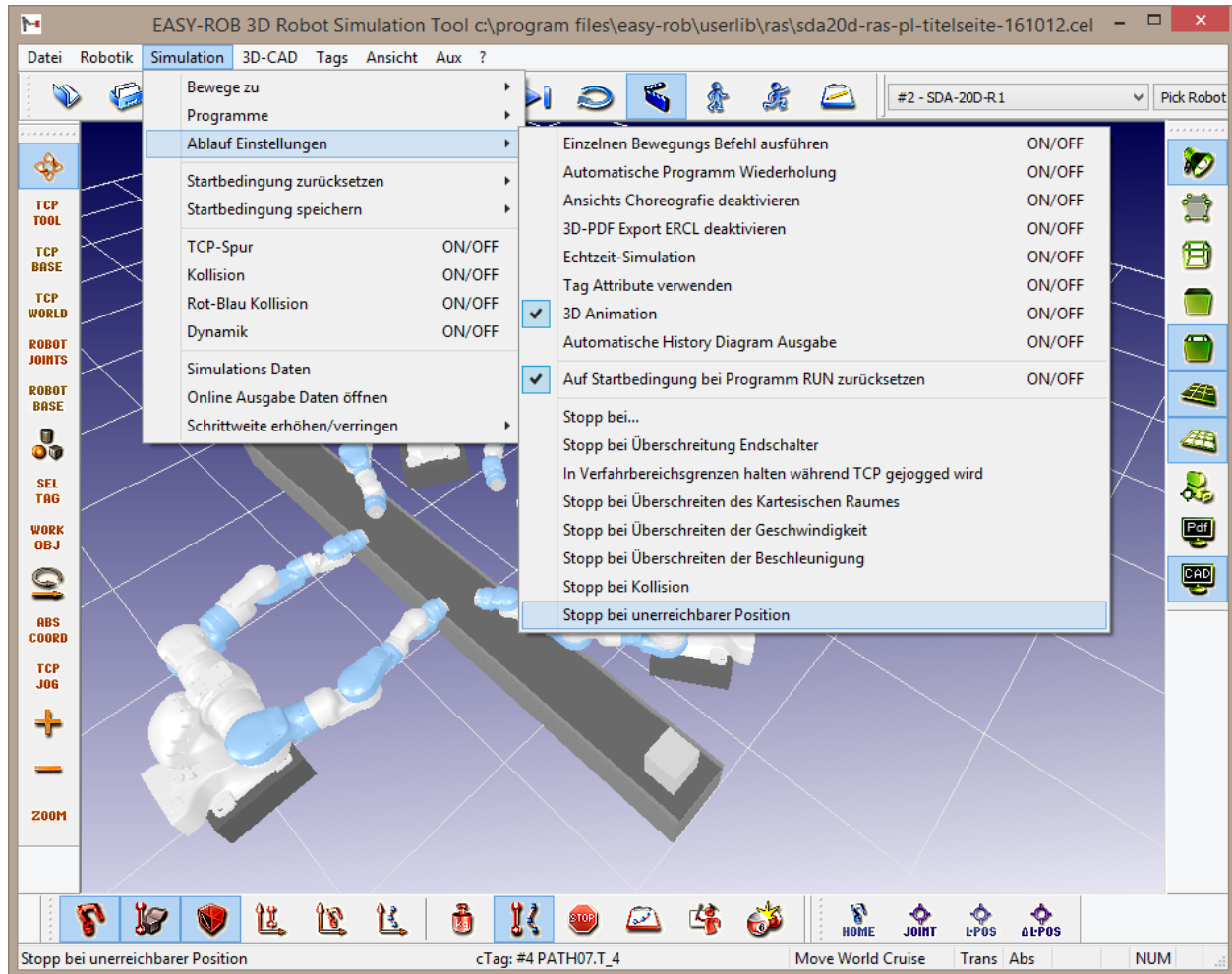
## Sprachumstellung (ENGLISCH / DEUTSCH / CHINESISCH)

Erstmalig in der Geschichte von EASY-ROB™ können Sie das Graphical User Interface (GUI) wahlweise zwischen folgenden Sprachen umstellen:

- Englisch
- Deutsch oder
- Chinesisch.



GUI auf Chinesisch



GUI auf Deutsch

## Wie kann die Sprache des GUI umgestellt werden?

Die Lokalisation erfolgt unter dem Menüpunkt

„Datei → Edit → EASY-ROB Systemfiles → Localization Datei“

und wird durch ausklammern der *nichtgewünschten* Sprachen mit einem „!“ vor den Zeilen

- en-US für Englisch
- de-DE für Deutsch
- zh-Hans für Chinesisch (traditionell)

erreicht. Auf der nachfolgenden Seite wurde in der Lokalisationsdatei die deutsche Sprache aktiviert.

## Neustart von EASY-ROB™ zur Aktivierung nötig

Damit ein Wechsel der GUI Sprache erfolgen kann, muss nach der Anpassung der Lokalisationsdatei diese gespeichert und EASY-ROB™ neu gestartet werden.

EASY-ROB™ sucht dabei nach folgenden Lokalisation-DLLs im Installationsverzeichnis von Version 7.0:

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

Sollten die Lokalisation-DLLs nicht vorhanden sein, startet EASY-ROB™ mit Englischem GUI, was auch die Standardeinstellung ist.

## Lokalisationsdatei "easy-rob-localizationx64.ini" m Überblick

```
! 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
!
```

Die Lokalisationsdatei easy-rob-localizationx64.ini liegt im Installationsverzeichnis von EASY-ROB™.

## In eigener Sache

Mit ca. 2200 Zeilen an Übersetzungsleistung je neuer Sprache des GUIs ist EASY-ROB stolz auf die geschaffene Arbeit und freut sich, wenn die wachsenden EASY-ROB™ Community davon profitiert.

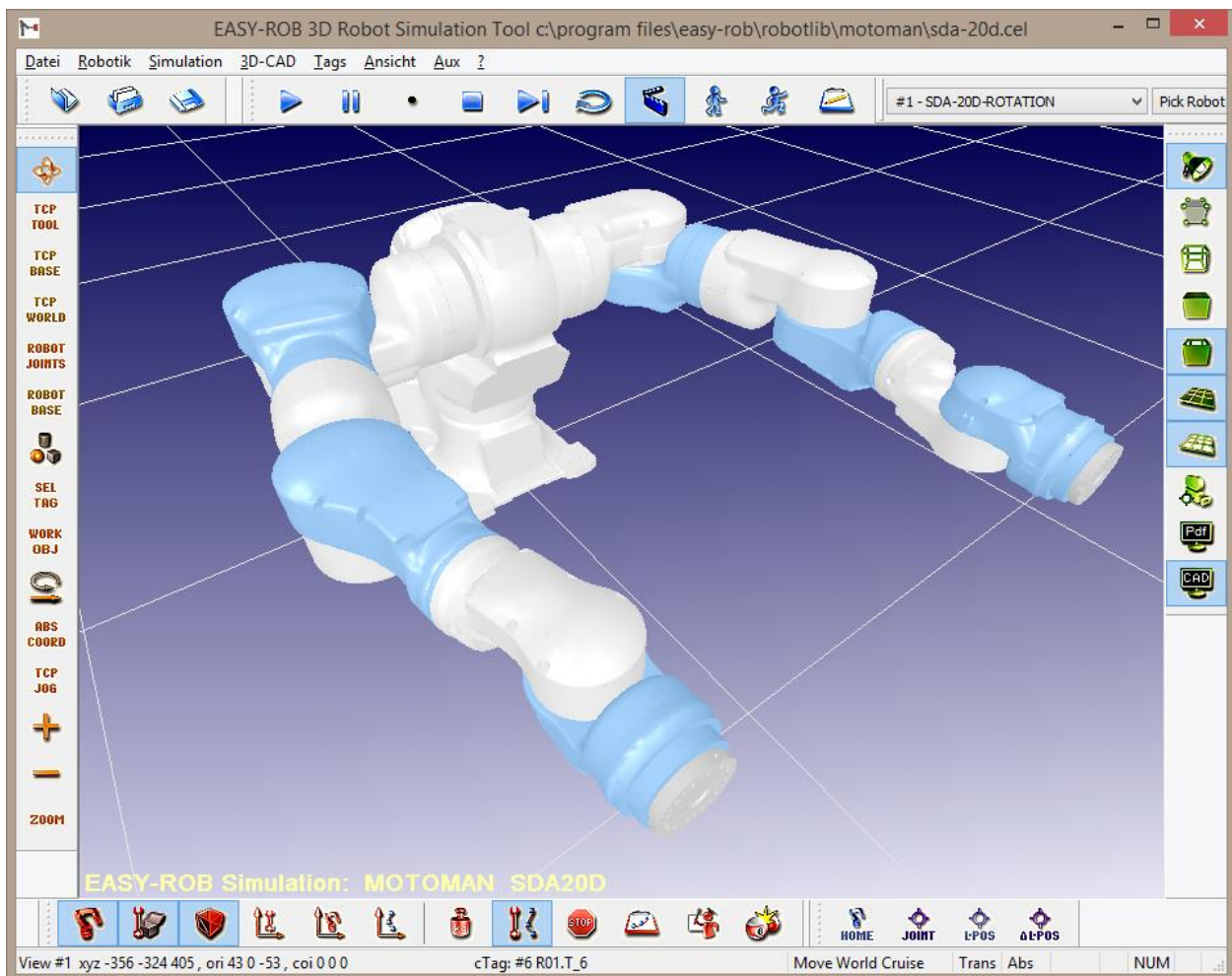


## Robot Assembly

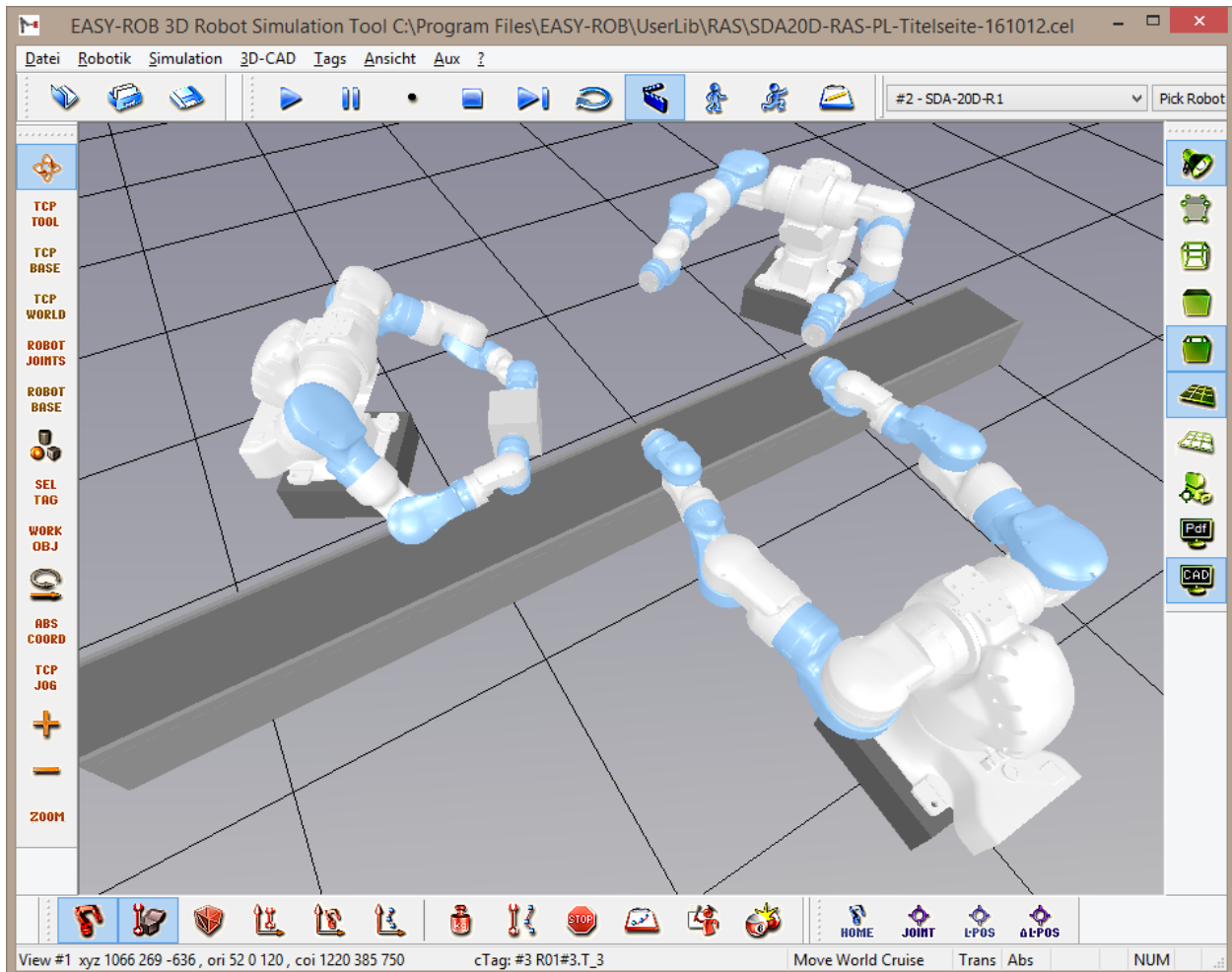
Komplexe kinematische Modelle können nun einfach und bequem als Robot Assembly aus einzelnen Robotern zusammengefügt, gespeichert und arbeitszellen-unabhängig wieder verwendet werden. Die Bezeichnung Assembly ist an Baugruppen aus dem CAD angelehnt. In diesen Assemblies können Modelle aus einzelnen Robotern/Devices/Geräten zusammengesetzt und aufgebaut werden. Optional können im Robot Assembly auch Programme, Pfade und Tags gespeichert werden.

Ihr EASY-ROB™ V7.0 erzeugt für Robot Assemblies eine neue Datei Erweiterung: \*.ras; die Robot Assembly Datei.

Damit muss z.B. der Dual-Arm Roboter Typ „SDA20D“ von Motoman/Yaskawa bestehend aus 3 Robotern mit insgesamt 15 Achsen nicht mehr als \*.cel-Datei gespeichert werden, sondern kann nun als Robot Assembly beliebig oft in \*.cel-Dateien geladen und simuliert werden, siehe bitte Screenshot:



Motoman/Yaskawa SDA20D als einzelne Robot Assembly Datei "SDA20D-RAS-PL.ras"



Mehrere Robot Assemblies vom Typ SDA20D in Kooperation in einer Arbeitszelle  
"SDA20D-RAS-PL-Titelseite-161012.cel"

## Einfach speichern und laden wie Roboter

Robot Assemblies werden wie Roboter behandelt und mit Ihnen können nicht nur Attribute wie z.B. Home Positionen, sondern auch erstellte Pfade bzw. Tag Punkte und sogar Programme gespeichert und geladen werden.

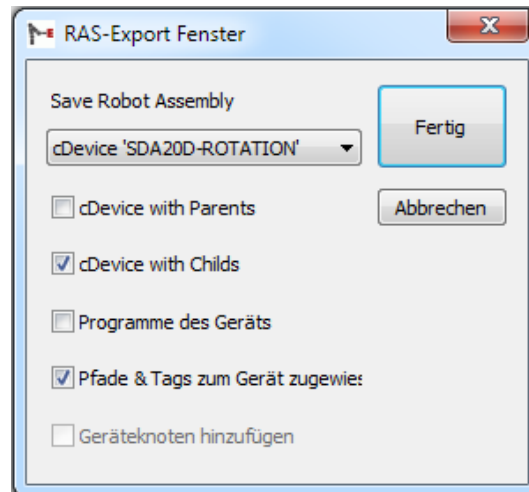
Unter dem Menü

- „Datei → Speichern → Roboter Assembly Datei“

kann Ihre individuelle Roboter Kinematik gespeichert und unter

- „Datei → Laden → Roboter Assembly Datei“

geladen werden.



Robot Assembly Speichern Dialog

Beim Speichern eines Robot Assemblies können optional alle "Child-Devices", die zugehörigen ERPL-Programme und die zugewiesenen Pfade & Tags gespeichert werden.

Beim erneuten Laden eines Robot Assemblies in ein und dieselbe Arbeitszelle \*.cel-Datei werden die Pfad-Namen im Tag Window mit einem „#“ + Index eindeutig gekennzeichnet. So wird zum einen eine Verwechslung der Tag Points ausgeschlossen und zum anderen wird ermöglicht, dass sogar ein simulieren ohne zusätzliche Modifizierung, selbst wenn das Roboter Assembly bzgl. Welt Koordinatensystem verschoben wurde, erfolgreich ausgeführt werden kann. Hier müssen Sie ggfs. in den ERPL Programmen der diversen Robot Assemblies den Aufruf der Pfade entsprechend des Index anpassen.

Wie gewohnt steht Ihnen der praktische Device Manager zur Verfügung. Browsen Sie in Ihr Zielverzeichnis und unter der Auswahl

■ „.rob/.ras“

laden Sie ein Robot Assembly, als ob Sie einen gewöhnlichen Roboter laden.

## Laden und speichern auch über Methoden der Klasse ER\_CAPI möglich

Mehr dazu bitte ab Seite 25 oder im doxygen-Dokument „EASY-ROB ER\_CAPI.chm“

## Modularer Aufbau von Produktionslinien möglich

Als weitere Anwendung der Robot Assemblies lässt sich auch folgender modularer Aufbau von Simulationen realisieren:

Mit den neuen Robot Assembly Dateien können Sie auch ganze Fertigungsstationen oder Arbeitsinseln erstellen und separat speichern. Diese wiederum stehen Ihnen dann zur Verfügung um anschließend flexibel kombinierbar ganze Produktionslinien modular zu gestalten. Kombinieren Sie Ihre Fertigungsstationen nach Belieben und klicken Sie Ihre Produktionslinie kundenspezifisch schnell und individuell zusammen.

## Layout Funktionen für 3D-PDF Export mit Animation

Die gesamte 3D-Simulation mit Animation- in nur einem PDF-Dokument!

Nutzen Sie das EASY-ROB™ 3D-PDF Export Plug-In und erstellen Sie in nur wenigen Schritten ein animiertes PDF-Dokument von Ihrem Simulationsablauf. Das EASY-ROB™ 3D-PDF Export Plug-In ermöglicht Ihnen, Ihre Simulation als animiertes PDF Dokument zu exportieren und mit Kunden, Projektpartnern und Kollegen zu teilen.



### Anwendungsbereiche

- Schnelle und einfache Präsentation-auch gegenüber Dritten
- Weitergabe von interaktiven Simulations-Konzepten
- Montage- und Wartungsanleitungen
- Dokumentation von erklärungsintensiven Inhalten
- Universelles Schulungs- und Trainingsmaterial
- Interaktive Vertriebsmaterialien für ein verbessertes Produktverständnis bei Kunden

### Layout Funktionen für erfolgreichen Wiedererkennungswert

Damit Ihre Kunden, Partner und Mitarbeiter sich noch besser mit Ihren Simulationsergebnissen identifizieren können, wurde der 3D-PDF Export in EASY-ROB™ Version 7.0 um Layout Funktionen erweitert.

Mit bis zu 12 unterschiedlichen Bildern können Sie u.a. Logos und weitere statische Informationen, z.B. Kundennamen, auf die von Ihnen erstellte 3D-PDF Seite bringen. Über die Kontrolldatei "3D-PDF-Layout.3pl" kann nicht nur die Lage der Bilder auf dem 3D-PDF bestimmt, sondern auch die Skalierung über weitere Parameter kontrolliert werden; diese können entsprechend gestaucht bzw. gedehnt werden.

Die Kontrolldatei sowie die verwendeten Bilder müssen dabei nicht im Verzeichnis der Projektdatei (\*.cel) liegen, sondern können über „IMAGE\_PATH“ intelligent eingebunden werden. Die Bilder sind als unabhängiges und universelles JPG Format lesbar.

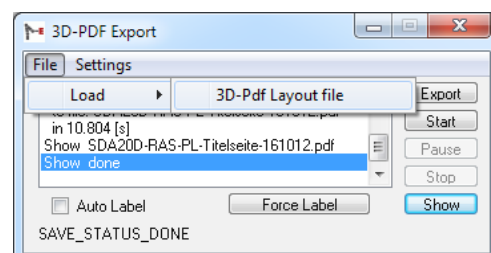
Auch die Navigationsleiste kann jetzt angepasst werden, siehe bitte Screenshot „Layout Bereich und Einflussnahme über Kontrolldatei“.

Damit Ihr 3D-PDF eine gelungene Präsentation ergibt, besteht jetzt die Möglichkeit, die Schriftgröße der gesetzten Labels über „LABELFONTSIZE“ schnell und einfach zu variieren.

### Erneute Benutzung der \*.3pl-Datei erwünscht

Natürlich können „\*.3pl“ Dateien nach der Erstellung im „3D-PDF Export“-Dialog im File-Menü erneut geladen und verwendet werden, siehe bitte Screenshot.

So bleiben Ihre individuellen Layout-Einstellungen auch für spätere Projekte Ihrer Kunden etc. erhalten.



## Kontrolldatei "3D-PDF-Layout.3pl" im Überblick

Die Kontrolldatei 3D-PDF-Layout.3pl ist eine ASCII Datei und kann mit einem Texteditor z.B. Notepad editiert werden.

```
! -----
! 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™ Bild-Dateien, die beliebig ersetzt, ergänzt und positioniert werden können.



## 3D-Pdf Export with Animation

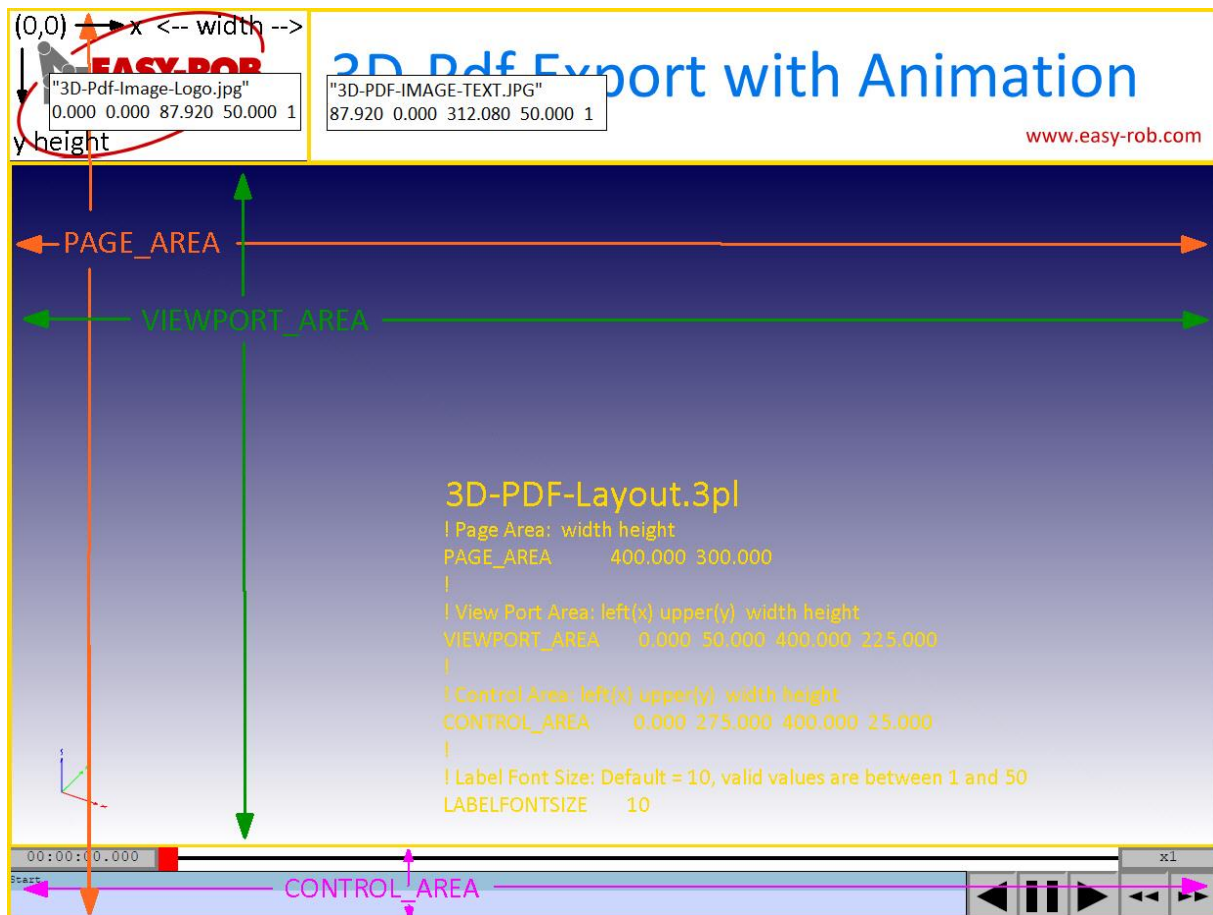
[www.easy-rob.com](http://www.easy-rob.com)

3D-Pdf-Image-Logo.jpg

3D-Pdf-Image-Text.jpg

## Funktionen und Parameter des 3D-PDF Layouts

Auf dem nächsten Screenshot sind die Bereiche dargestellt, auf welche man Einfluss nehmen kann:



Layout Bereich und Einflussnahme über Kontrolldatei

- PAGE AREA = gesamte Seitenhöhe und Breite einstellen
- VIEWPORT AREA = Höhe und Breite 3D Fenster manipulieren
- CONTROL AREA = Skalierung der Navigationsleiste vornehmen
- LABELFONTSIZE = Schriftgröße von Labels beeinflussen
- IMAGE\_PATH = Pfad zu verwendeten Bildern festlegen; sonst WORKDIR
- ADD IMAGE = Bilder für Layout auswählen, positionieren und skalieren



## ERCL Befehle für Layout Funktionen des 3D-PDF Exports

Über zusätzliche ERCL Befehle können die Layout Funktionen angesteuert werden:

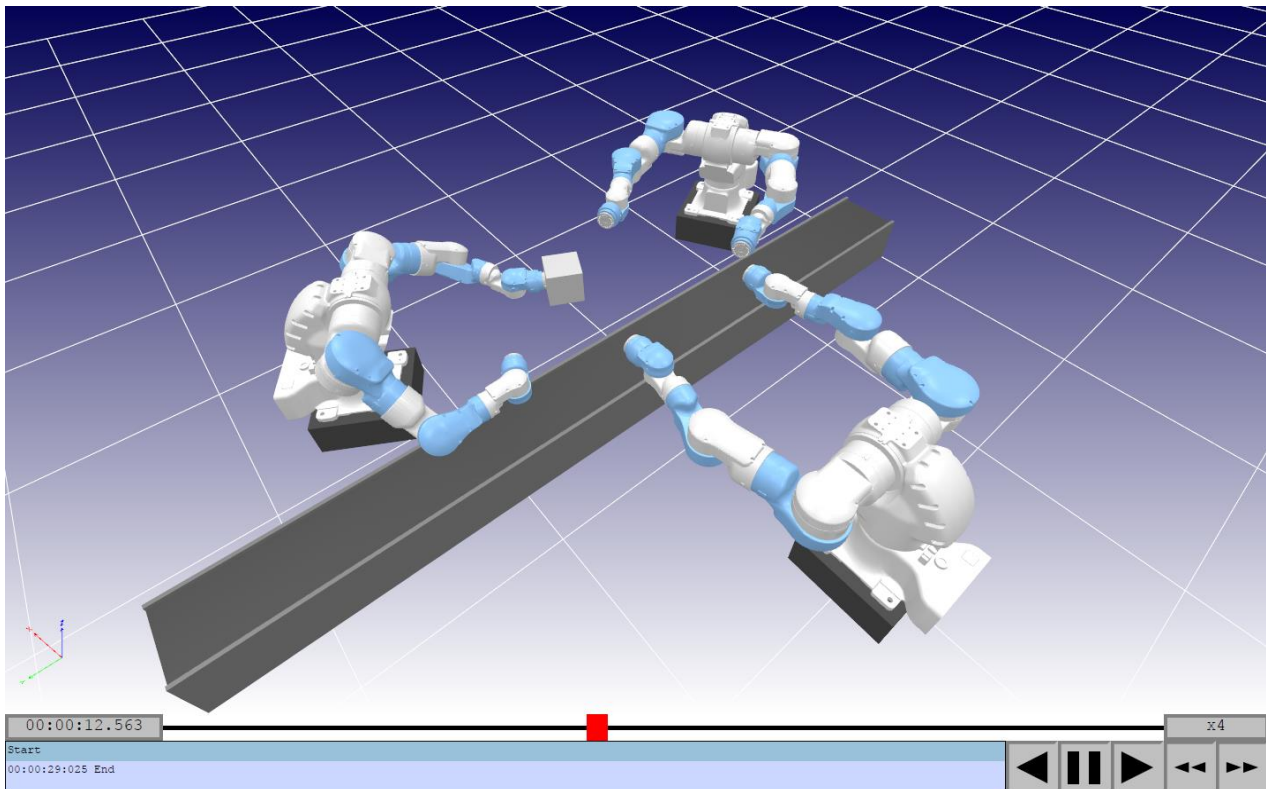
ERC _3D_PDF_EXPORT RESET_LAYOUT	
ERC _3D_PDF_EXPORT LOAD_LAYOUT	filename.3pl
ERC _3D_PDF_EXPORT SAVE_LAYOUT	filename.3pl
ERC _3D_PDF_EXPORT PAGE_AREA	width height
ERC _3D_PDF_EXPORT VIEWPORT_AREA	left(x) upper(y) width height
ERC _3D_PDF_EXPORT CONTROL_AREA	left(x) upper(y) width height
ERC _3D_PDF_EXPORT 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\_DE.pdf“ als weiterführendes Dokument

Eine Beschreibung der Kommandos finden Sie im Abschnitt “Neue ERCL-Befehle” oder im entsprechenden Dokument zur ERPL- / ERCL- Programmiersprache „EASY-ROB-ERPL\_DE.pdf“.

## Bewegungsablauf mit Animation in 3D-PDF speichern

Im Adobe® Reader können Sie mit der Navigationsleiste den aufgezeichneten Bewegungsablauf starten, pausieren, stoppen, vor- und zurückspulen, sowie die Geschwindigkeit (x1/64-fache bis x64-fache) ändern. Die Zeitangabe gibt die dabei die reale Prozesszeit an.



Navigationsleiste im Adobe® Reader

Die 3D-PDF-Export-Funktionalität kann alternativ auch über ERCL-Kommandos gesteuert werden. Dabei stehen dem Benutzer die folgenden ERCL-Befehle zur Verfügung:

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

Eine Beschreibung der Kommandos finden Sie im Abschnitt "Neue ERCL-Befehle" und im Dokument zur ERPL- / ERCL- Programmiersprache „EASY-ROB-ERPL\_DE.pdf“.



Der 3D-PDF Export lässt sich auch über die API Methodenklasse:

- ER\_CAPI\_USER\_IO\_3DPDF

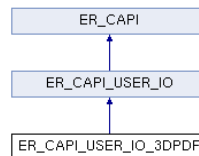
ansteuern.

### 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](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)

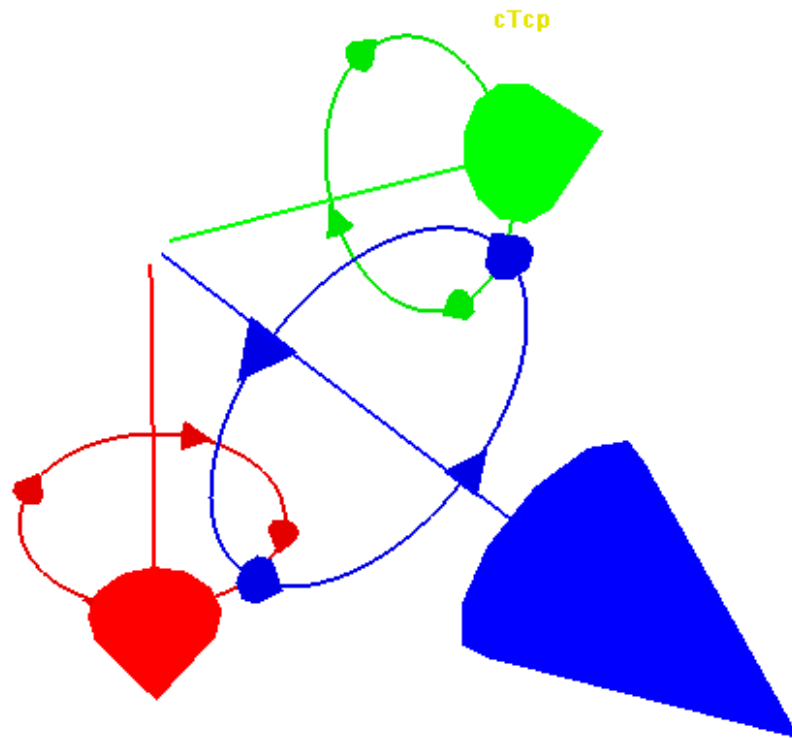
## TCP JOG Koordinatensystem

Damit sie stets die richtige Richtung im Auge behält, ist nun während des kartesischen oder achsweisen Joggens am Tip bzw. Flansch des Roboters ein schickes Koordinatensystem zu sehen, welches die Richtung verdeutlicht.

Dabei werden die Achsen folgt dargestellt:

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

Die positiven Drehrichtungen für die Rotation sind mit konzentrischen Kreisen um die jeweilige translatorischen Achsen/Richtungen und den jeweiligen Vektor gezeichnet.



TCP JOG Coorsys TCP Tool im Detail

## Aktivierung abhängig von jeweiligen JOG Modus

Die generelle Aktivierung des TCP JOG Koordinatensystem erfolgt über das Menü

- „Ansicht → KoorSys → cRobot TCP JOG Koordinatensystem anzeigen“

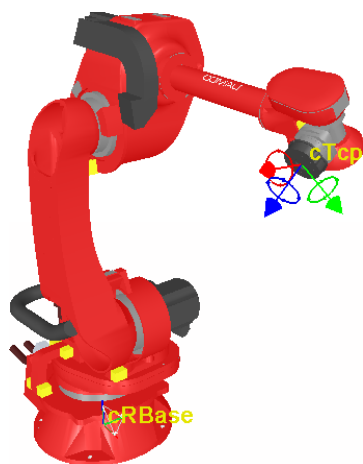
Damit die Einstellung beim erneuten Start von EASY-ROB™ erhalten bleibt, kann sie in der Umgebungs-Datei "easy-rob.env" gespeichert werden.

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

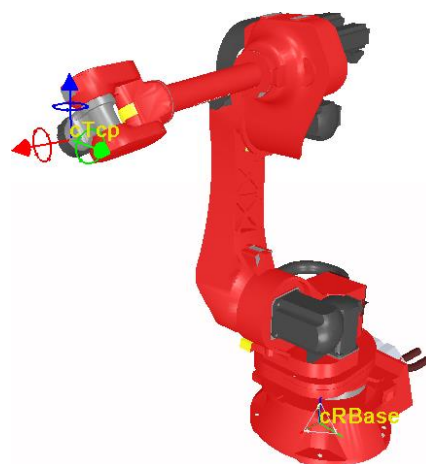
Das TCP JOG Koordinatensystem wird nach Auswahl des JOG Modus aktiviert und stellt entweder

- den Robot TCP mit Button TCP TOOL,
- nach Robot Base mit Button TCP BASE oder
- nach Welt mit Button TCP WORLD dar.

Im nachfolgenden Beispiel sind 2 unterschiedliche JOG Modi zu sehen:



TCP Tool aktiviert



TCP Base aktiviert

## Während Simulation ausgeblendet

Während der Simulation wird das TCP JOG Koordinatensystem, zwecks besserer Übersicht, ausgeblendet.

## Ohne Inverse kein TCP JOG Koordinatensystem

Wurde einer Kinematik keine Inverse Lösung zugewiesen, wie z.B. Positionierer, Zuführeinheiten oder individuelle Kinematiken bzw. wurde diese nicht aus den Referenz Kinematiken ausgewählt, siehe Seite 28, wird das TCP JOG Koordinatensystem nur vereinfacht dargestellt.

## Lizenzierungs-Varianten nun Thread-Safe

EASY-ROB™ unterstützt neben der rechnergebundenen Lizenz (Hardware-Nummer) und dem EASY-ROB™ Lizenz Manager (floating licenses) mehrer Dongle-Varianten für den USB Anschluss. Dazu zählen WibuKey-Dongle und CodeMeter CmStick von WIBU-SYSTEMS AG und der neu hinzugekommene MatrixLock-Dongle von TechnoData Interware GmbH.



WibuKey Dongle



CodeMeter, CmStick



MatrixLock-Dongle

Jeder dieser Dongle repräsentiert sich durch eine eindeutige ID (auch als Hardware-Nummer oder Hardware ID bezeichnet), welche die Grundlage zum Erzeugen des Lizenzierungsschlüssels darstellt.

Insbesondere für unsere OEM Kunden ist diese Flexibilität sehr wichtig, so dass die Produkte EASY-ROB™ DLL Version und EASY-ROB™ Robotics Simulation Kernel mit der Lizenzierungsvariante der Host-Anwendung konform ist.

Die Dongle-Hersteller stellen zur Ansteuerung des Dongles eine API zur Verfügung die nicht unbedingt Thread-Safe (Threadsicher) ist.

Damit es auch aus verschiedenen Threads nicht zu Fehlern kommt wurde der Dongle-Zugriff durch Mutex-Objekte und Semaphoren synchronisiert.

### Bemerkung:

Der Dongle-Zugriff während der Simulation ist extrem selten, so dass es zu keinerlei Performance-Einbußen kommt. In der Regel erfolgt ein Dongle-Zugriff beim Initialisieren und beim Laden von neuen Objekten, beispielsweise Roboter und Robot Assemblies (Roboter-Gruppen).

### Hinweis:

Neue API Funktionen für DLL Version und Kernel erlauben den expliziten Dongle-Zugriff zum Auslesen der Dongle-ID, zur Lizenzabfrage und zur Bestimmung des Pfades für die Lizenzdatei "license.dat".

siehe

- `erKernelGetHardwareID(...)`, `erKernelGetLicense(...)`, `erKernelSetLicenseFile(...)`

## Robotics Simulation Kernel Klasse ERK\_CAPI

Neue Methoden sind zur Klasse **ERK\_CAPI\_ADMIN** hinzugekommen, die die Lizenzierung, aktuelle Hardware-ID (Dongle oder rechnerbezogen) und lizenzierte Optionen betreffen.

```
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 available

**static DLLAPI int ER\_STDCALL ERK\_CAPI\_ADMIN::erKernelGetHardwareID ( char \* hw\_id )** static

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 )** static

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 )** static

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 Kollision Thread-Safe

Der Kernel liefert auch Methoden zum Prüfen von Kollisionen zwischen zwei triangulierten 3D-Geometrien. Die Klasse **ERK\_CAPI\_SIM\_COLLISION** liefert neue Methoden so dass der Kollisionstest Thread-Safe erfolgt

In der alten Variante (nicht Thread-Safe) wurde die Kollision mit dem Aufruf von `erColl_ChkCollision(...)` geprüft und die Ergebnisse mit einem weiteren Aufruf `erColl_GetResults_Collide(...)` abgeholt. Bei der neuen Thread-Safe-Variante wird das Ergebnis beim Aufruf von `erColl_ChkCollision_res(..., void *pres)` sofort geliefert. Der Speicher für `void *pres` muss vorab entsprechend zugewiesen werden.

```
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
)
static
```

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

## Klasse ER\_CAPI EASY-ROB™ DLL- und Multi-Robot Version

Neue Methoden sind zur Klasse **ER\_CAPI** hinzugekommen,  
siehe Header-Dateien: er\_CAPI.h und er\_CAPI\_Types.h

Neue Konstanten 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
```



## Neue und erweiterte Methoden in er\_CAPI.h

### Robot-Assemblies laden und speichern

```
int ER_CAPI_USER_IO_FILE::LoadRobotAsm ( )
```

```
static ER_DllExport int ER_CAPI_USER_IO_FILE::LoadRobotAsm ( void )
```

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 )
```

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_flm = NULL,
                                                                int    confirm_overwrite = 1
                                                                )
```

Saves a robot Assembly file by filename

If new\_ras\_flm 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\_flm**          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**

## Verfahrenbereiche für passive Achsen

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**

## Pfade mit Tags klonen

```
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

## EASY-ROB™ DLL Window starten/Initialisieren

```
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 – Kinematische Strukturen im Überblick

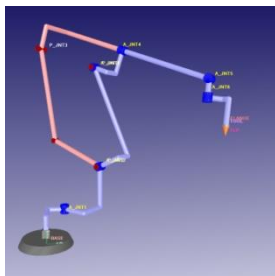
Zu jedem kinematischen Modell, für das in EASY-ROB™ eine analytische inverse Transformation (IK) existiert, finden Sie die Inverse Kin-ID und eine dazugehörige Sub-ID vor. Zusammen ergeben diese die Referenz Kinematik ID = RefKin ID.

Wenn Sie z.B. in EASY-ROB™ Kinematics einen neuen Roboter erstellen, können Sie die passenden IDs in „Kin-IDs-Parameters\_DE.pdf“ nachschlagen und anschließend vergeben, damit Ihr Modell eine korrekte analytische inverse Transformation besitzt.

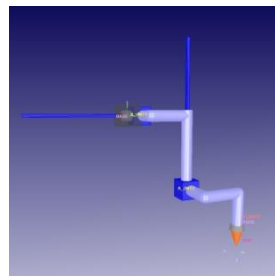
### Übersicht der kinematischen Modelle

Folgende kinematischen Strukturen werden als RefKins näher beschrieben:

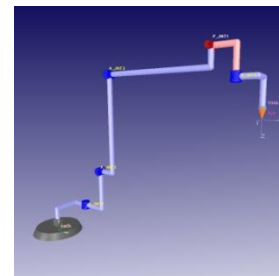
- Vertikal-Knickarmroboter mit Zentralhand (ggfs. mit Kopplung und Backlink; 6 Achsen)
- Jet-Roboter (ggfs. mit Kopplung und Backlink; 6 Achsen)
- Palettier-Roboter (ggfs. mit Backlink; 4 Achsen)
- SCARA-Kinematiken (4 bis 6 Achsen)
- Portal-Roboter (1 bis 6 Achsen)



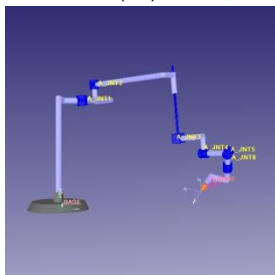
V-Knickarm (BL) ID 111, SID 0



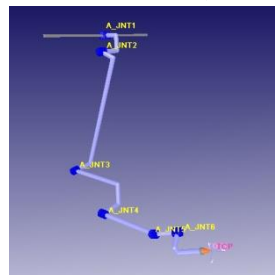
2-Achs-Portal ID 133, SID 13,23



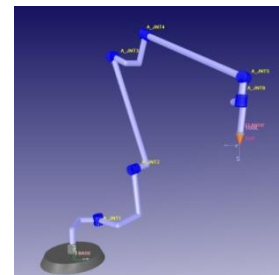
Palettier-Rob. ID 129, SID 0



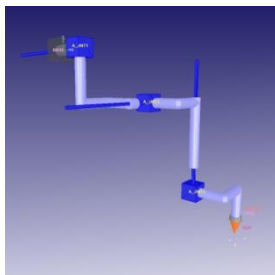
6-Achs-SCARA ID 139, SID 0



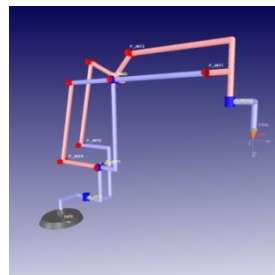
Jet-Roboter ID 127, SID 0



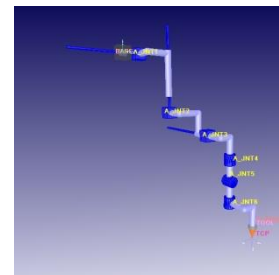
V-Knickarm ID 110, SID 1



4-Achs-Portal ID 134, SID 0,123



Palettier-Rob. ID 129, SID 1



6-Achs-Portal ID 136, SID 0

Beispiele einiger Referenz-Kinematiken als **parametrisierte** Roboter Modelle.

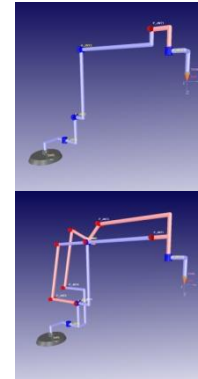
## Beispiel Kin-ID und Sub-ID

So hat ein **Palettier-Roboter** folgende IDs:

- Kin-ID 129 – kann auch als Hauptklasse verstanden werden
- Sub-ID 0 – ist eine Untergruppe zur weiteren Klassifizierung

Ein **Palettier-Roboter mit Backlink** weist jedoch folgende IDs auf:

- Kin-ID 129 – gleich dem normalen Palettier-Roboter
- Sub-ID 1 – Unterscheidung aufgrund der Backlink Kinematik.



## Transformations-Tabelle

An Hand dieser Tabelle können Sie ablesen, welche Transformations-Längen und -Rotationen für jede aktive Achse zur nächsten aktiven Achse "Geometric Data to next" erlaubt sind, damit die analytische inverse Transformation (IK) mit der angegebenen Kin-ID und Sub-ID für jede mögliche Konfiguration eine Lösung berechnet. Grüne Häkchen deuten an, dass eine entsprechende Transformation zugelassen ist.

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

Transformationstabelle für Palettier-Roboter: Kin-ID 129, Sub-ID 0

## Rob-Dateien als Beispiele für Kin-IDs

Zu jedem Kinematik-Beispiel finden Sie eine entsprechende Rob-Datei im Verzeichnis „RefKin-RobID“ vor. Die Darstellung dieser Beispiel-Modelle wurde auf das wesentliche reduziert. So sind aktive Rotationsachsen als blaue Zylinder und aktive Translationsachsen als blaue Würfel visualisiert. Passive Achsen dagegen sind entsprechend rot. Die Verbindung zwischen den einzelnen Achsen und damit die kinematische Kette wird über dünne Zylinder (primitive Geometrien) dargestellt.

Im Dokument „TrainLib-Tutorial“ finden Sie im Kapitel 6 „Aufsetzen von Kinematiken“ eine detaillierte Schritt-für-Schritt-Anleitung wie Sie einfache Kinematiken aus Standard CAD-Komponenten selbst erstellen können.

## Kin-IDs-Parameters\_DE.pdf als weiterführendes Dokument

Weitere Informationen erhalten Sie im Dokument „Kin-IDs-Parameters\_DE.pdf“. Dieses Dokument liefert Ihnen eine kompakte, aber auch detaillierte Übersicht über die wichtigsten kinematischen Strukturen in EASY-ROB™.

Zusätzlich finden Sie auf der letzten Seite des Dokuments eine Beschreibung der „Numerischen Lösung“ Kin-ID = 100, Sub-ID = 0/1. Diese liefert eine Lösung zur Berechnung der Inversen von kinematischen Modellen, bei denen keine analytische Lösung zur Verfügung steht.



Kin-IDs-Parameters\_DE.pdf

## Vollständige Roboterbibliotheken

In EASY-ROB™ stehen vollständige Bibliotheken zur Einbindung aller bedeutenden Robotertypen des Marktes bereit. Dazu zählen ABB, b+m, Comau, Denso, Eisenmann, Fanuc, Guedel, igm, Kawasaki, KUKA, Mitsubishi, OTC-Daihen, Reis, Stäubli, Tricept, Unimation, Universal Robots und Yaskawa.

Die Roboterbibliotheken von ABB, KUKA, Comau, Fanuc, Stäubli und Yaskawa sind nahezu vollständig und werden permanent von uns gepflegt.

Derzeit sind mehr als 1000 Roboter, Positionierer und externe Tracking-Achsen verschiedener Hersteller verfügbar.

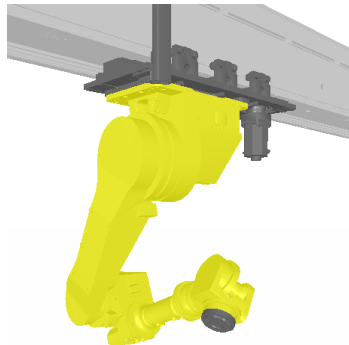
Weitere Informationen:

<http://www.easy-rob.com/produkt/erweiterungen/roboter-bibliotheken.html>

Neue Modelle von Fanuc z.B. M-20iA-35M und R-2000iB-200T-Portal oder von Stäubli z.B. TX-340SH gesellen sich zu den zahlreichen Robotern hinzu.



Fanuc M-20iA-35M



Fanuc R-2000iB-200T-Portal



Stäubli TX-340SH

### Wichtig:

Nicht vorhandene Roboter, Handlingsysteme, Maschinen, Werkzeuge oder auch spezielle Kinematiken lassen sich in EASY-ROB™ einfach und schnell „virtuell nachbauen“.

## Multi-Robot – Multi-Kin und Multi-Program jetzt vereint!

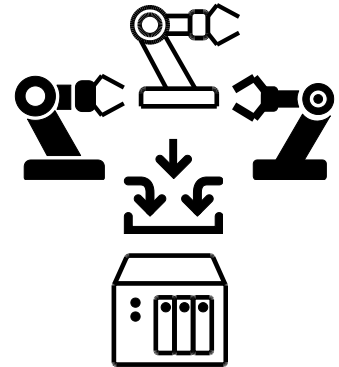
Mit zunehmender Simulationskomplexität war eine Trennung der Multi-Kin und Multi-Program Version kaum noch möglich. EASY-ROB entschloss sich deshalb, beide Versionen in der neuen Multi-Roboter Version als mächtiges Simulationstool für Mehr-Roboter Anwendungen zu vereinen.

### Übersicht Optionen

Die **Multi-Robot** (EXE,DLL) baut auf der **Single-Robot** Version auf.

- Die Anzahl der Roboter\*) *und* Programme pro Arbeitszelle ist nicht limitiert. Kinematiken wie Greifer, Positionierer und/oder Zuführeinheiten können beliebig oft geladen, verkettet und simuliert werden.
- Mit der **Multi-Robot** Version lassen sich mehrere Roboter und Kinematiken synchronisiert und gleichzeitig bewegen, wobei die Programme über I/O-Signale kommunizieren.

\*) Roboter haben mindestens 4 Achsen und können kartesische Zielpunkte (Tags) anfahren.



### Update möglich

Für Kunden älterer Versionen besteht die Möglichkeit ein Update zu erwerben. Nehmen Sie dazu bitte mit unserem Vertrieb unter +49 6192 921 70 79 oder [sales@easy-rob.com](mailto:sales@easy-rob.com) Kontakt auf und geben Ihre Hardware-/Dongle-Number und die Versionsnummer Ihrer Lizenz an. Ihre Hardware Number (HwNr) finden Sie folgt: Starten Sie EASY-ROB™ und öffnen das Menü

- „? → License Info“.

Wiederholen Sie ggfs. diesen Schritt für weitere Lizenzen.



## Neue ERCL-Befehle

### ERCL - 3D-PDF Export Layout Definition Kommandos

**ERC \_3D\_PDF\_EXPORT RESET\_LAYOUT**

3D-Pdf Layout auf Standard-Werte zurücksetzen

**ERC \_3D\_PDF\_EXPORT LOAD\_LAYOUT** filename.3pl

3D-Pdf Layout aus .3pl Datei "filename.3pl" laden

**ERC \_3D\_PDF\_EXPORT SAVE\_LAYOUT** filename.3pl

3D-Pdf Layout in .3pl Datei "filename.3pl" speichern

**ERC \_3D\_PDF\_EXPORT PAGE\_AREA** width height

Page Area des 3D-Pdf Layout festlegen;  
Parameter: width height, siehe Bild Seite 14

**ERC \_3D\_PDF\_EXPORT VIEWPORT\_AREA** left(x) upper(y) width height

Viewport Area des 3D-Pdf Layout festlegen;  
Parameter: left(x) upper(y) width height, siehe Bild Seite 14

**ERC \_3D\_PDF\_EXPORT CONTROL\_AREA** left(x) upper(y) width height

Control Area des 3D-Pdf Layout festlegen;  
Parameter: left(x) upper(y) width height, siehe Bild Seite 14

**ERC \_3D\_PDF\_EXPORT LABEL\_FONT\_SIZE** size

Label Font Size im Control Area des 3D-Pdf Layout festlegen;  
Parameter: size im Bereich [1-50], Default-Wert: size = 10

## Bilder zum 3D-PDF Layout hinzufügen

Der Pfad muss vorab mit dem Befehl IMAGE\_PATH festgelegt werden. Pfad- und Bildangaben müssen in "" erfolgen.

**ERC \_3D\_PDF\_EXPORT IMAGE\_PATH "path"**

Pfadangabe aus denen die Bilder mit ADD\_IMAGE geladen werden.

**ERC \_3D\_PDF\_EXPORT IMAGE\_PATH USERPROFILE**

Der Key "USERPROFILE"  
bestimmt den Pfad aus denen die Bilder mit ADD\_IMAGE geladen werden z. B: c:\Users\MyLoginName

**ERC \_3D\_PDF\_EXPORT IMAGE\_PATH WORKING\_DIRECTORY**

Der Key "WORKING\_DIRECTORY",  
current working directory bestimmt den Pfad aus denen die Bilder mit ADD\_IMAGE geladen werden

**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 "  
Der Pfad der in der geladenen 3D-Pdf Layout Datei .3pl definiert ist bestimmt den Pfad aus denen die Bilder mit ADD\_IMAGE geladen werden

**ERC \_3D\_PDF\_EXPORT ADD\_IMAGE "filename.jpg" left(x) upper(y) width height Scaling**

Fügt ein jpg Bild an definierte Position ein  
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

## Kontakt

### EASY-ROB 3D Robot Simulation Tool

Adresse: Hauptstr. 42  
65719 Hofheim am Taunus  
Germany

Inhaber: Stefan Anton

Tel.: +49 (0) 6192 921 70-77 / -79  
Fax: +49 (0) 6192 921 70 66

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

Web: [www.easy-rob.com](http://www.easy-rob.com)

Online Shop: <http://www.easy-rob.com/produkt/shop.html/>

### EASY-ROB Kundenbereich

Inhalte: Programm-Updates und Roboterbibliotheken

Web: [www.easy-rob.com/special/kundenbereich](http://www.easy-rob.com/special/kundenbereich)

Zugangsdaten:

Benutzer:	customer
Passwort:	*****

## Eigene Notizen