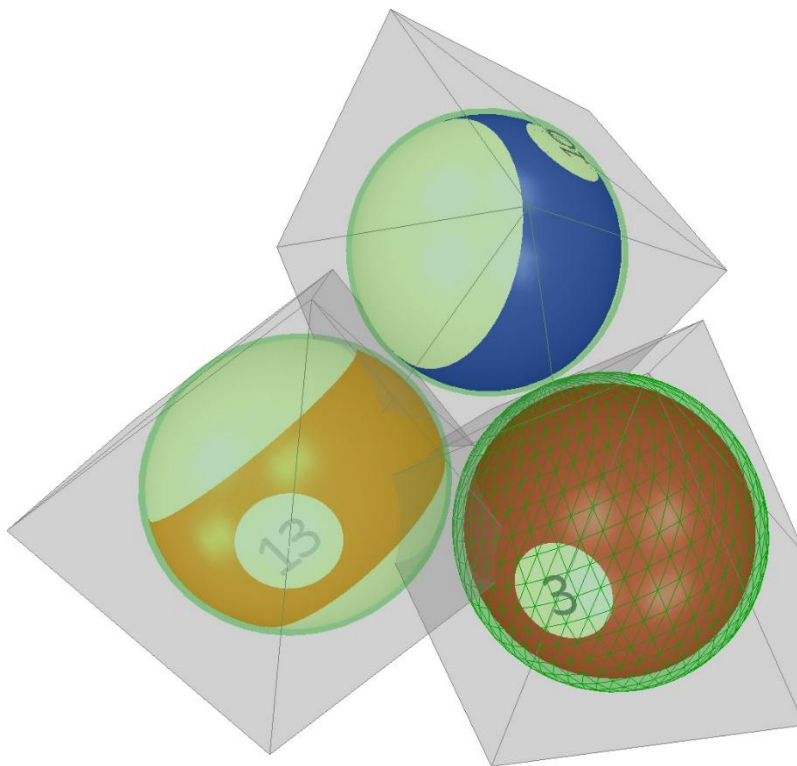


# Die neue Version

## EASY-ROB™ V7.3



Oktober 2017

Version 1.5



# EASY-ROB™

## Inhaltsverzeichnis

<b>EASY-ROB™ V7.3 Update .....</b>	<b>5</b>
<b>Individuelle Pfade für temporäre und Benutzerdateien konfigurieren .....</b>	<b>6</b>
Temporäre Dateien.....	6
Benutzerdateien .....	6
Fehlende Einträge in der config.dat .....	6
easy-rob.pth - Neue Datei zur Verwaltung von Arbeits- und Geometriedateien .....	7
easy-rob.pth direkt editieren .....	7
Vollständiger Pfad im Message Window einsehbar .....	7
easy-rob.pth-Datei im Überblick für Dateipfade .....	8
config.dat-Datei im Überblick .....	8
<b>Wo werden die eingestellten Pfade angezeigt? .....</b>	<b>10</b>
Aux Menü in EASY-ROB™ .....	10
<b>EASY-ROB™ VISU .....</b>	<b>11</b>
Übersicht über Vorteile und Anwendungsmöglichkeiten.....	12
Feature-Übersicht.....	12
<b>EASY-ROB™ Collision .....</b>	<b>13</b>
Übersicht über Vorteile und Anwendungsmöglichkeiten.....	13
Feature-Übersicht.....	14
<b>EASY-ROB™ 3D-PDF SDK .....</b>	<b>15</b>
Programmierschnittstelle .....	15
Feature-Übersicht.....	16
<b>Linkage ersetzt Synchronized .....</b>	<b>17</b>
ERC LINKAGE Befehle .....	18
Schematische Darstellung des Achsen-Mappings.....	19
<b>Re-Design von Dialogen.....</b>	<b>20</b>
Neuer Style .....	20
<b>Ignore Limits.....</b>	<b>21</b>
<b>Verfeinerte Werteingabe.....</b>	<b>22</b>
<b>CheckBoxen für Eingabe Dialoge .....</b>	<b>23</b>
„Check All“ wählt alle Optionen an oder ab .....	23
<b>Robotics Simulation Kernel Klasse ERK_CAPI .....</b>	<b>24</b>
Satzvorlauf Methode "erSET_NEXT_TARGET_ADVANCE()" .....	24
Satzvorlauf Struktur "NEXT_TARGET_DATA_ADVANCE" .....	25
<b>Klasse ER_CAPI EASY-ROB™ DLL- und Multi-Robot Version .....</b>	<b>26</b>
Neue Konstanten in er_CAPI_Types.h .....	26
Neue und erweiterte Methoden in er_CAPI.h .....	27
<b>Vollständige Roboterbibliotheken.....</b>	<b>34</b>
<b>Dual-Arm Robots.....</b>	<b>35</b>
<b>Neue ERPL- und ERCL-Befehle.....</b>	<b>37</b>
ERCL - Linkage Device .....	37
ERCL - Display Device .....	37
Parser Funktionen .....	37
<b>Kontakt.....</b>	<b>38</b>
<b>Eigene Notizen .....</b>	<b>39</b>



## EASY-ROB™ V7.3 Update

Hallo liebe EASY-ROB™ Community!

Wir freuen uns, die neue EASY-ROB™ Version 7.3 vorstellen zu dürfen und wie immer findet Ihr die Highlights hier in der Schnellübersicht:

- **Individuelle Pfade für EASY-ROB™ Systemdateien**  
Es können nun Windows-benutzerabhängig die Pfade für Systemdateien eingestellt werden.
- **EASY-ROB™ VISU**  
3D Visualisierung für die Industrie  
Das leistungsstarke EASY-ROB™ VISU dient zur Integration in technologiebasierte Softwareanwendungen. Damit kann man Prozesse visualisieren, Problemstellungen aufzeigen oder Kunden und Kollegen einfach das neue Projekt sehen lassen.
- **EASY-ROB™ Collision**  
Mehr als 1000-fach erprobt–  
jetzt separat verfügbar. Dank der intelligenten Softwarearchitektur EROSA steht Ihnen ab sofort EASY ROB™ Collision zur Verfügung, ein hochperformantes Softwaremodul zur Kollisionserkennung von 3D Objekten.
- **EASY-ROB™ 3D-PDF SDK**  
Der belastbare 3D-PDF Export für Ihre eigene Softwareanwendung nun als SDK verfügbar mit vielen Features u.a. Animation.
- **Linkage ersetzt Synchronized**  
Aber weiter als das kann zum einen ein Master-Slave Wechsel programmiert werden und zum anderen können sogar die Achsen von Devices gemappt werden.
- **Re-Design**  
Größere Buttons in angepassten Dialogen
- **Ignore Limits**  
Damit kann man schnell ein bestehendes Wertelimit kurzweilig aushebeln um noch freier zu simulieren.
- **CheckBoxen für Eingabe Dialoge**  
Bequeme Mehrfachauswahl durch Einsatz von CheckBoxes
- **Dual Arm Robots**  
EASY-ROB™ unterstützt Dual Arm Robots. Diese werden als RAS-Dateien abgelegt.

Ab sofort steht allen Kunden mit einer gültigen v7.3 Lizenz oder einem Softwarepflegevertrag die neue EASY-ROB™ Version 7.3 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.

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

Vielen Dank

Ihr EASY-ROB Team

## Individuelle Pfade für temporäre und Benutzerdateien konfigurieren

Endlich kann man individuelle Pfade für EASY-ROB™ Systemdateien konfigurieren.

Damit können mehrere Benutzer an einem Rechner sicher und effizient arbeiten und entsprechend verwaltet werden. EASY-ROB™ greift dabei auf die Windows User Verwaltung zu. Die verschiedenen Ordnerplätze sind frei wählbar, müssen aber Schreibrechte aufweisen.

Die Ordnerpfade werden in der Konfigurationsdatei "config.dat" und in der neuen easy-rob.pth-Datei konfiguriert. Die Konfigurationsdatei befindet sich immer EASY-ROB™ Installationsverzeichnis und wird vom Administrator konfiguriert.

### Temporäre Dateien

Bei der Arbeit mit EASY-ROB™ werden temporäre Dateien erzeugt und diese werden nach Beendigung unter Umständen nicht gelöscht. Die Pfade für die temporäre Dateien werden in der „config.dat“ mit "TMPDIR=" festgelegt. In diesem Verzeichnis müssen Schreibrechte vorhanden sein.

Zu den temporären Dateien gehören z.B.:

- moni\_msg.txt
- HardwareNumber.dat

Beispiel:       TMPDIR=%USERPROFILE%\EASY-ROB\tmp

### Benutzerdateien

Zu den Benutzerdateien gehören Dateien, die der EASY-ROB™ Bediener für seine Bedürfnisse individuell anpassen kann. Die Pfade für die Benutzerdateien werden in der config.dat mit "USRDIR=" festgelegt. In diesem Verzeichnis müssen Schreibrechte vorhanden sein.

Zu den Benutzerdateien gehören:

- easy-rob.pth (neue Datei)
- easy-rob.env
- easy-rob-localization.ini
- er\_LoadFromLibPb.ini
- er\_LoadFromLibPb\_prefered.ini

Beispiel:       USRDIR=%USERPROFILE%\EASY-ROB

### Fehlende Einträge in der config.dat

Werden in der config.dat keine Pfade für "TMPDIR=" und "USRDIR=" festgelegt, wird standardmäßig das EASY-ROB™ Installations-Verzeichnis gesetzt. Das entspricht dem alten Zustand bis Version 7.0.

## easy-rob.pth - Neue Datei zur Verwaltung von Arbeits- und Geometriedateien

Damit Anwender Ihre eigenen Arbeits- und Geometrieordner festlegen können, welche ggfs. von den Benutzerordnern abweichen, wurden die Pfade für diese Ordner aus der config.dat herausgetrennt und werden zukünftig in der neuen Datei **“easy-rob.pth”** verwaltet.

Hier können Pfade für die Arbeits- und Geometrieverzeichnis mit

- WORKDIR=
- IGPDIR=

festgesetzt werden.

## easy-rob.pth direkt editieren

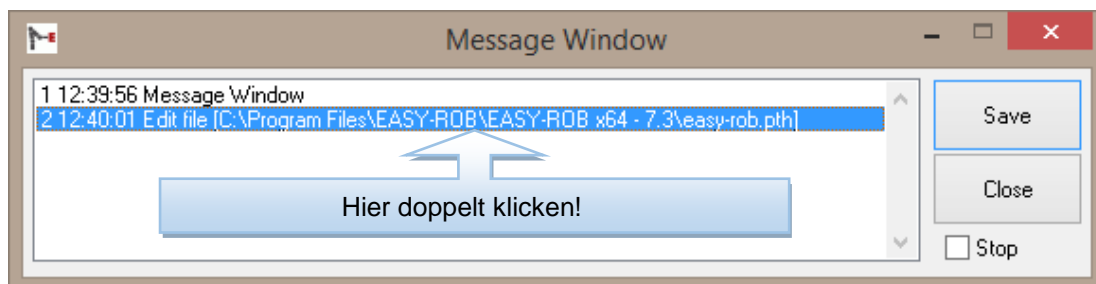
Aus der EASY-ROB™-Bedienoberfläche können Sie die Datei easy-rob.pth direkt editieren und sich anzeigen lassen:

- Menu File → Edit → EASY-ROB System Files → Working pathes file

## Vollständiger Pfad im Message Window einsehbar

Ist das Message Window geöffnet (CTRL+M) wird der Ort der Datei angezeigt.

Weiterhin kann die easy-rob.pth-Datei durch einen Doppelklick aus dem Message Window geöffnet werden.



Vollständiger Pfad im Message Window

## easy-rob.pth-Datei im Überblick für Dateipfade

### Beispieldatei für 'easy-rob.pth':

```
!-----  
! Working pathes file  
!  
! File name:      "easy-rob.pth"  
! Location:       User folder  
!  
!-----  
!  
WORKDIR= ..\Projects\Proj  
IGPDIR=  ..\Projects\Proj\igp  
!  
WORKDIR= ..\Projects_2\Proj  
IGPDIR=  ..\Projects_2\igp  
!
```

## config.dat-Datei im Überblick

### Beispieldatei für 'config.dat':

```
!-----  
! Configuration file  
!  
! File name:      "config.dat"  
! Location:       Working directory  
!  
!-----  
!  
! Files located in 'TMPDIR' Folder are  
! - moni_msg.txt  
! - HardwareNumber.dat  
!  
TMPDIR= .\  
!TMPDIR=%USERPROFILE%\EASY-ROB\tmp  
!TMPDIR= C:\Users\MyUser Name\AppData\Local\Temp  
!  
!-----  
!  
! Files located in 'USRDIR' Folder are  
! - easy-rob.pth  
! - easy-rob.env  
! - easy-rob-localization.ini  
! - er_LoadFromLibPb.ini  
! - er_LoadFromLibPb_prefered.ini  
!
```



```
USRDIR= .\
!USRDIR= %USERPROFILE%\EASY-ROB
!USRDIR= C:\Users\MyUser Name
!
!-----
! License file
!-----
!
WIBUKEY_USE = 1
CODEMETER_USE= 1
MATRIXLOCK_USE= 1
LMNGR_USE= 1
!
LICENSE= ..\
!
!-----
! API UserDll
!-----
!
USER_DLL= api_user_01x64.dll 1 api_user_01 (x64)
!
!-----
! Preferred Editor
!-----
!EDIT= my_editor -> your preferred editor
EDIT= notepad
!
```

## Wo werden die eingestellten Pfade angezeigt?

Aus EASY-ROB™ heraus können Sie die aktuell eingestellten Systempfade einzusehen.

### Aux Menü in EASY-ROB™

Innerhalb von EASY-ROB™ können Sie die Systempfade und –dateien editieren und sich anzeigen lassen:

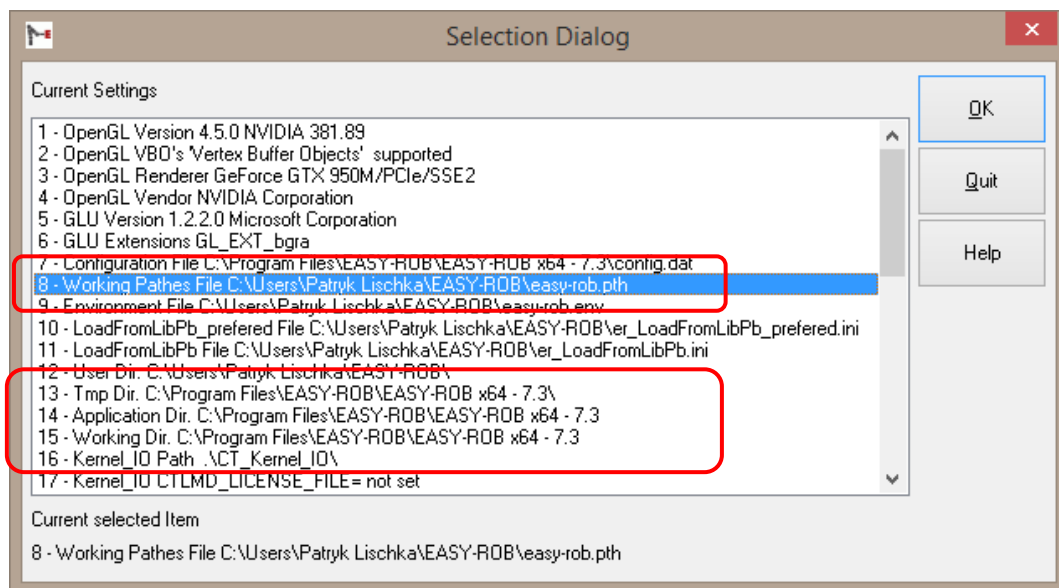
- Menü Aux → Einstellungen → Aktuelle Einstellungen anzeigen

Bei Anwahl der einzelnen Dateien per Doppelklick können diese auch direkt editiert werden. Im genannten Menü haben die relevanten Dateien folgende Nummern vorne weg stehen:

- 7 - Configuration File
- 8 - Working Pathes File

Weiter sind die aktuell konfigurierten Pfade bzw. Benutzer- und Arbeitsverzeichnisse aufgelistet:

- 12 - User Dir.
- 13 - Tmp Dir.
- 14 - Application Dir.
- 15 - Working Dir.



Beispiel eines Selection Dialogs mit markierten Einträgen

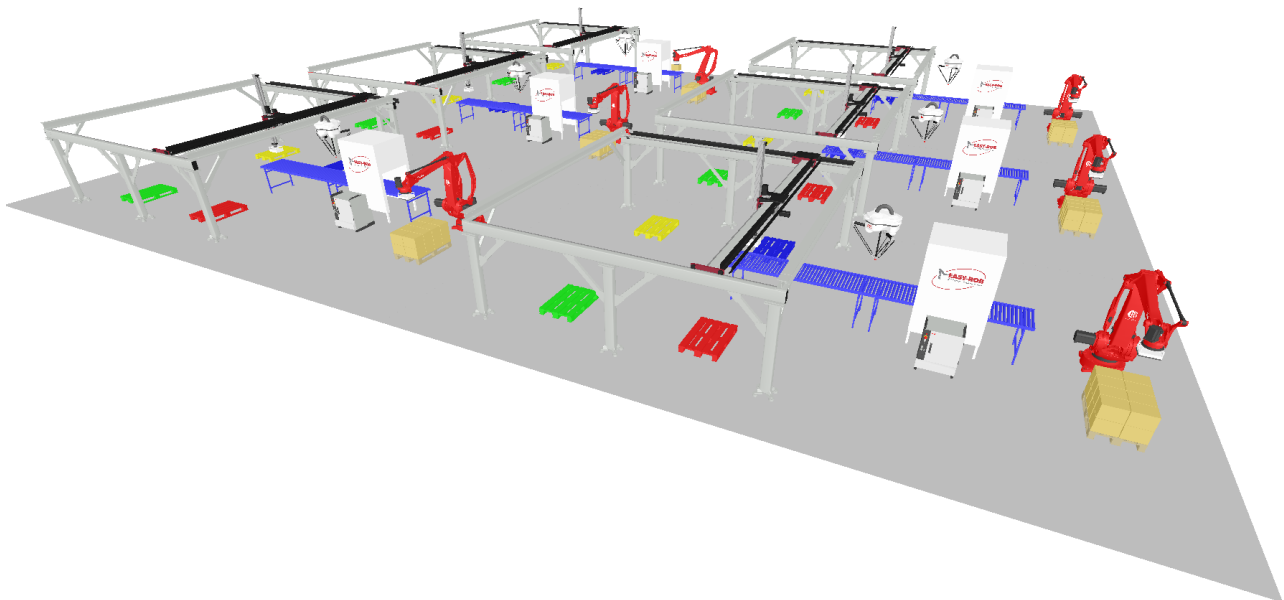
Hinweis: Die Pfade werden auch in der Datei moni\_msg.txt gespeichert.

## EASY-ROB™ VISU

Aufgrund der starken Nachfrage nach einer immer feineren Aufteilung der Anwendungsbereiche bei Software Modulen bietet nun EASY-ROB ab Version 7.3 ein reines Visualisierungsmodul als DLL/API an. Das neue Mitglied in der Product Family: EASY-ROB™ VISU.

Dieses eigenständige Modul ist als 3D Visualisierung für die Industrie gedacht und spricht alle Firmen an, welche technologiebasierte Softwareanwendungen entwickeln, aber keine eigene Visualisierung haben oder sich mit sehr eingeschränkte Hardware Anforderungen konfrontiert sehen.

Damit lassen sich aber auch einfach nur Prozesse visualisieren, Problemstellungen aufzeigen oder Kunden und Kollegen neue Projekte präsentieren.



Komplexe Simulation mehrerer Arbeitszellen

Für Rückfragen oder weitere benötigte Informationen steht Ihnen unser Vertrieb unter:

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

gerne zur Verfügung!

## Übersicht über Vorteile und Anwendungsmöglichkeiten

### Vorteile

- Preisleistungsvorteil durch Eigenentwicklung
- Hochperformant bei geringen Hardwareanforderungen
- Integration in technologiebasierte Softwarelösungen
- Kein dedizierter Speicher nötig
- Freie Platzierung des OpenGL™ Fensters
- Verfügbar unter Windows® 32- und 64-Bit

### Anwendungsmöglichkeiten

- Animation und Simulation
- Werbung, Verkauf und Vertrieb
- Darstellung von Sensordaten
- Echtzeitanbindung
- Positionierung von Geometrien
- Branchenunabhängig

## Feature-Übersicht

### OpenGL™ Engine

- Robuste Grafik Engine auf Basis der OpenGL™ Grafikbibliothek



### AVI Recorder

- Integrierte Erzeugung von Video-Dateien in vielen Auflösungen



### Integration

- Detaillierte Doxygen Dokumentation
- Programmierbeispiele für MS Visual Studio® C/C++ und C#



### Mathematik

- Methodenklasse für math. Berechnungen, u.a. von homogenen Matrizen
- Umwandlung von Winkeln, Dreiecksberechnung und Formel Parser etc.



### Messwerkzeuge

- Position, Entfernung, Durchmesser etc.



### Kamerafunktion

- Statisch/bewegbar und beliebig oft frei positionierbar



### API

- C/C++ und C# Methodenklasse ERVisu\_CAPI



### CAD Import

- CAD Daten – Import und Export
- Modellierung einfacher parametrisierbarer 3D Geometrien

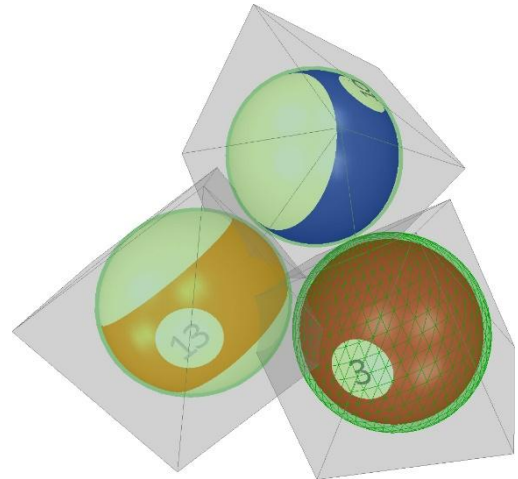


## EASY-ROB™ Collision

Auch bei dieser Entwicklung stand der Diversifikationsbedarf des Marktes im Fokus. EASY-ROB bietet ab Version 7.3 ein eigenständiges Softwaremodul zur Kollisionsprüfung als DLL/API an: EASY-ROB™ Collision.

Zielgruppe sind Unternehmen, die ein Softwaremodul zur hochperformanten Kollisionsprüfung suchen und die kostenintensive Eigenentwicklung gescheut haben.

EASY-ROB™ Collision hat sich tausendfach durch den Einsatz in den EASY-ROB™ Produkten bewährt und ist dabei absolut präzise und verlässlich. Gerade bei großen und komplexen 3D Modellen spielt EASY-ROB™ Collision seine beeindruckende Performance aus-  
auf EASY-ROB™ Collision ist immer Verlass!



Collision Logo symbolisiert durch  
Billard Kugeln.

## Übersicht über Vorteile und Anwendungsmöglichkeiten

### Vorteile

- Inklusive diverser Berechnungsmethoden
- Hochperformant bei geringen Hardwareanforderungen
- Threadsafe
- Leichte Integration in technologiebasierte Softwarelösungen
- Verfügbar unter Windows® 32- und 64-Bit

### Anwendungsmöglichkeiten

- Animation und Simulation
- Bewegungsplanung
- Montagetests
- Offline Programmierung
- Messprotokolle
- Virtual Prototyping
- Chirurgische Simulation
- Haptisches Rendern
- Molekulardesign
- Branchenunabhängig

Für Rückfragen oder weitere benötigte Informationen steht Ihnen unser Vertrieb unter:

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

gerne zur Verfügung!

## Feature-Übersicht

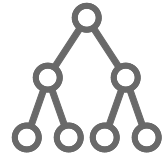
### Diverse Berechnungsmethoden

- Collision Detection
- Toleranz
- Distanz Berechnung



### Hierarchische Repräsentation

- Intelligent und schnell mit OBBTree-Tight-Fitting



### Collision Detection

- Methode, ob zwei triangulierte Modelle kollidieren
- Kollisionserkennung nach dem ersten kollidierenden Dreieckspaar
- Berechnung aller kollidierenden Dreieckspaare



### Threadsafe

- Paralleler Kollisionstest sicher auf Multi-Core CPU



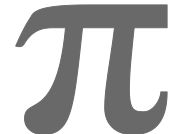
### Toleranz

- Prüft, ob zwei Modelle näher oder weiter als ein definierter Toleranzabstand sind



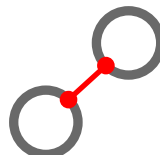
### Mathematik

- Methodenklasse für math. Berechnungen, u.a. von homogenen Matrizen



### Distanz Berechnung

- Berechnet den kleinsten euklidischen Abstand zwischen zwei nicht kollidierenden Modellen und
- Die 2 nächstliegenden Punkte auf den Modellen



### Integration

- Detaillierte Doxygen Dokumentation
- Programmierbeispiele für MS Visual Studio® C/C++ und C#



### API

- C/C++ und C# Methodenklasse ERColl\_CAPI



## EASY-ROB™ 3D-PDF SDK

Suchen Sie für Ihre Software einen belastbaren 3D-PDF Export mit Animation und sind bisher nicht fündig geworden?

Dann haben wir für Sie genau die richtige Lösung:  
Erweitern Sie Ihr Produkt um unseren erfolgreichen 3D-PDF Export!



### Anwendungsbereiche 3D-PDF Export

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

Das EASY-ROB™ 3D-PDF SDK ermöglicht Ihnen, für oben genannten Anwendungsbereiche benötigte Funktionen einfach in Ihre Software zu implementieren.

### 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

### Programmierschnittstelle

- Für das 3D-PDF SDK wird eine C/C++ und C# Methodenklasse zur Verfügung gestellt: `ER3DPDF_CAPI`

Für Rückfragen oder weitere benötigte Informationen steht Ihnen unser Vertrieb unter:

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

gerne zur Verfügung!

## Feature-Übersicht

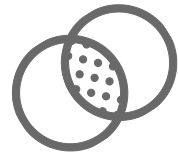
### Animation

- Exportiert die 3D-Szene als interaktives 3D-PDF
- Simulationslauf vor und zurückspulbar, und sogar rückwärts abspielbar!
- Control-Panel frei platzierbar
- Mit Animation



### GeoRending

- Lässt Sie Geometrien unterschiedlich darstellen z.B. zur Visualisierung bei Kollisionsprüfung
- Farbe
- Rendertyp z.B. Drahtmodell
- Sichtbarkeit (Visible/invisible)



### Statisch

- Exportiert die 3D-Szene als statisches 3D-PDF
- Ohne Animation



### Floordesign

- Parametrisierbarer Boden
- Länge und Breite
- Farbe



### Layoutfunktion

- Parametrisierbare Erscheinung
- Mehrere Bilder für Logodarstellung für Ihr CI



### Background

- Hintergrundfarbe einstellbar



### Label

- Ergänzt Ihr 3D-PDF um beliebige Labels
- Mit diesen lassen sich Simulationsabschnitte gezielt anspringen



### Integration

- Detaillierte Doxygen Dokumentation
- Programmierbeispiel für MS Visual Studio® C/C++ und C#



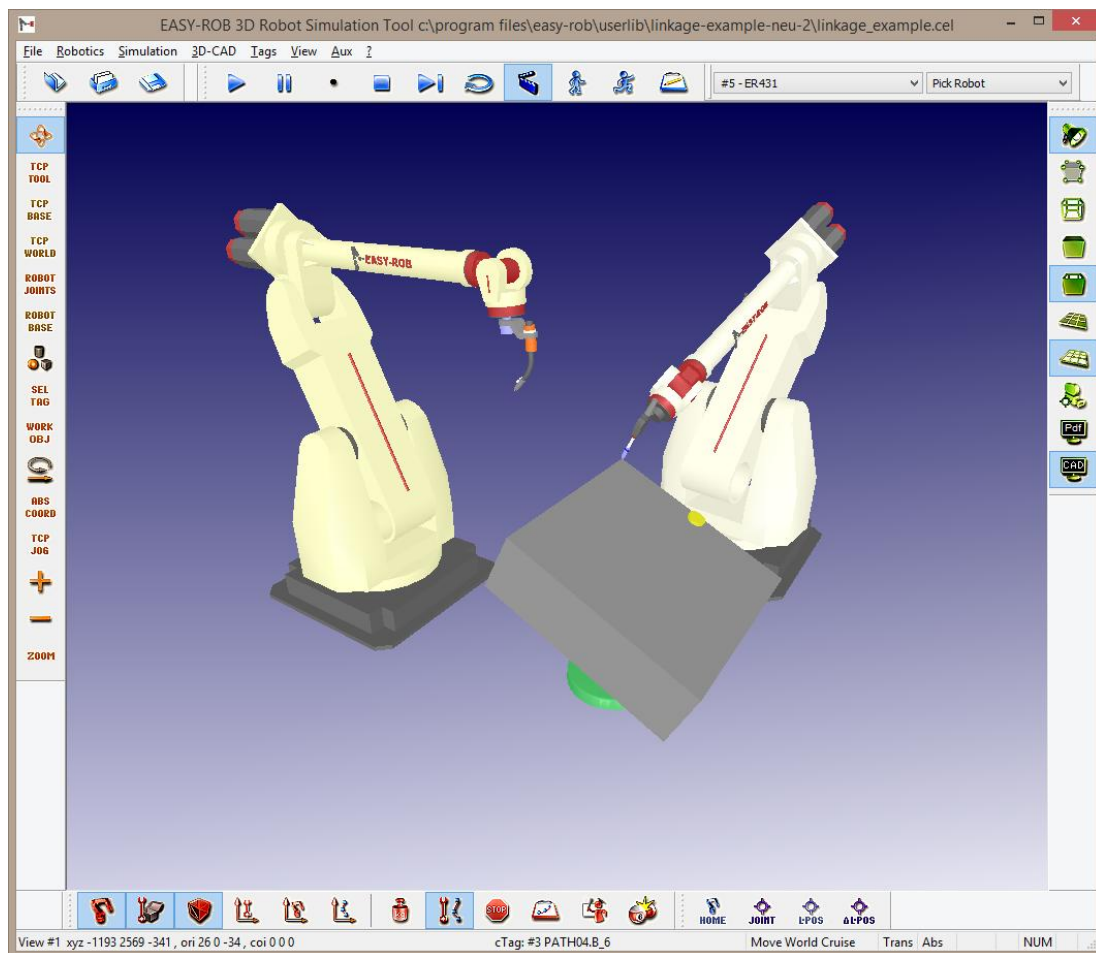


## Linkage ersetzt Synchronized

EASY-ROB™ bietet die Möglichkeit, dass Geräte Ihre Achsen mit Achsen von anderen von Geräten koppeln können z.B. kann ein Positionierer als 7. Achse eines Roboters eingebunden werden.

Dabei wurde der Begriff „Synchronized with“ in den jeweiligen Dialogen verwendet, was aber teilweise zu Missverständnissen führte.

Daher entschied man sich bei EASY-ROB, diese Funktion von „Synchronized with“ in „Linked with“ umzubenennen. Das Thema wird als Linkage behandelt u.a. spiegelt sich das auch in neuen ERCL-Befehlen wider.



Abwechselnde Kopplung zwischen Geräten

## ERC LINKAGE Befehle

Einhergehend mit dieser Umstellung wurden neue ERCL-Befehle implementiert, welche z.B. aus dem Simulationslauf einen Wechsel für solche Kopplungen zulassen. Damit kann z.B. ein Wechsel von Master und Slave elegant umgesetzt werden, d.h. zuerst steuert der erste Master-Roboter den Positionierer und später in der Simulation übernimmt dann der zweite Roboter (Slave) die Steuerung von diesem.

- ERC LINKAGE DEVICE SET 'DeviceName' AxIdx(1) .. AxIdx(n)  
Bewirkt Kopplung von Geräten zum Simulationslauf
- ERC LINKAGE DEVICE UNSET ['DeviceName']  
Hebt Kopplung wieder auf  
(A. d. R.: Wird ein weiterer DEVICE SET Befehl verwendet, ist UNSET nicht nötig)

Nachfolgende sind zwei Beispiele als ERCL-Befehle zu finden:

- ERC LINKAGE DEVICE SET TurnTable 0 7

Hier wird das aktuelle Gerät (Roboter) mit dem Gerät "TurnTable" (Positionierer als externe Achse des Roboters) gekoppelt.

Da dieser Roboter nur eine zusätzliche 7. Achse hat, der Positionierer "TurnTable" aber zwei Achsen, muss ein Mapping stattfinden. Das erfolgt mit den nachgestellten Indizes hinter dem Device-Namen "TurnTable". Hier ist die 2. Achse des "TurnTables" mit der 7. Achse des Roboters (aktuelles Gerät) gekoppelt.

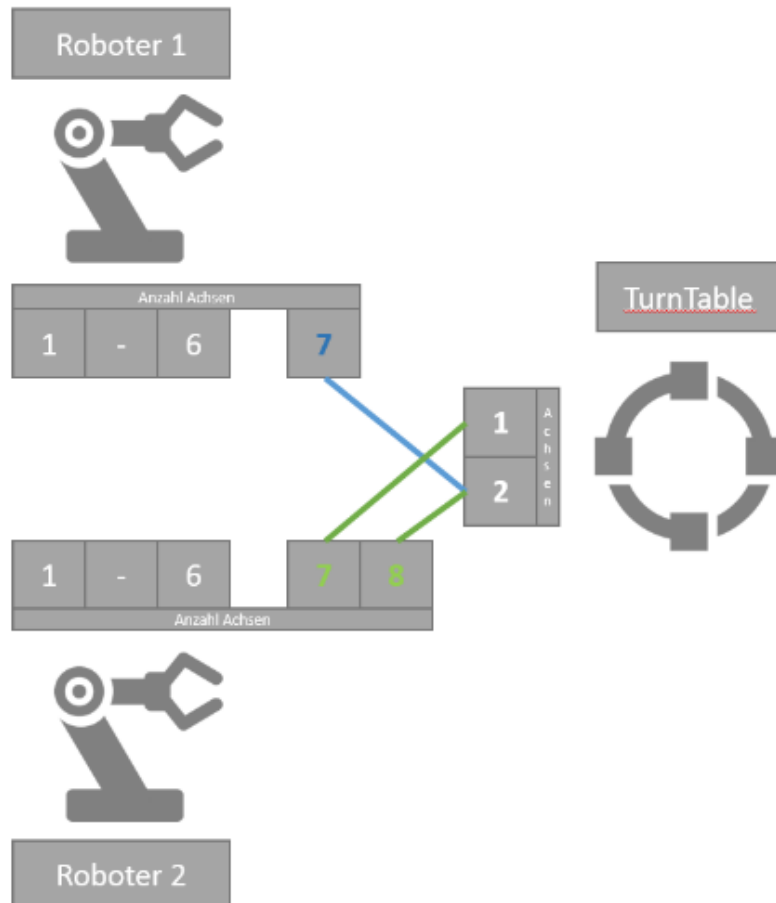
Mit der 0 wird die erste Achse des "TurnTables" vom Roboter ignoriert und nicht gekoppelt.

- ERC LINKAGE DEVICE SET TurnTable 7 8

Wie beim vorangegangenen ERCL-Befehl-Beispiel wird das aktuelle Gerät (Roboter) mit einem Gerät "TurnTable" gekoppelt. Der Unterschied liegt in der Anzahl der zusätzlichen Achsen, die der Roboter aufweist. Dieser hat wie der "TurnTable" ebenfalls zwei Achsen und das Mapping kann über Index 7, für die erste Achse, und Index 8, für die zweite Achse des "TurnTables" erfolgen.

Auf der nächsten Seite wird das oben genannte Beispiel gemappter Achsen wie folgt dargestellt:

## Schematische Darstellung des Achsen-Mappings



Schematische Darstellung des Achsen-Mappings

**blau** = ERC LINKAGE DEVICE SET TurnTable 0 7

**grün** = ERC LINKAGE DEVICE SET TurnTable 7 8

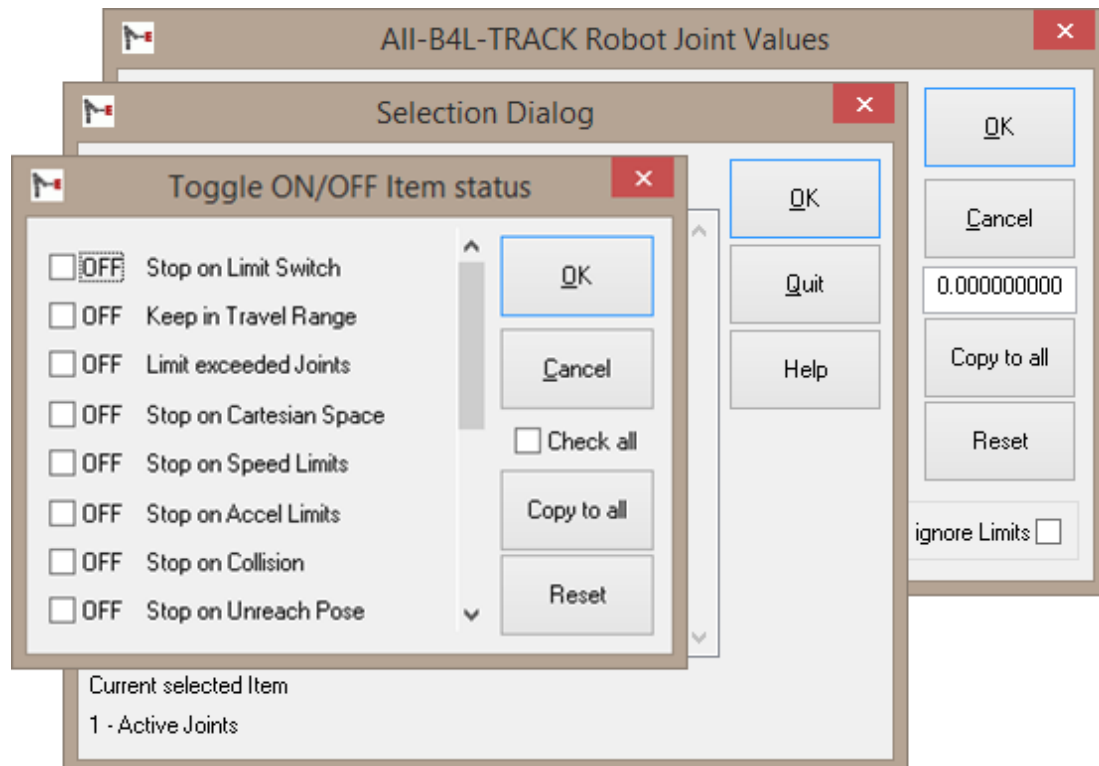
In „Neue ERCL-Befehle“ auf Seite **30**, später in diesem Dokument, sind noch weitere Linkage-Befehle zu finden.

## Re-Design von Dialogen

Viele Dialoge wurden im Rahmen des Re-Designs überarbeitet. Dabei konzentrierte man sich auf ein noch besseres Simulationserlebnis. Auf den folgenden Seiten werden die Neuheiten vorgestellt.

### Neuer Style

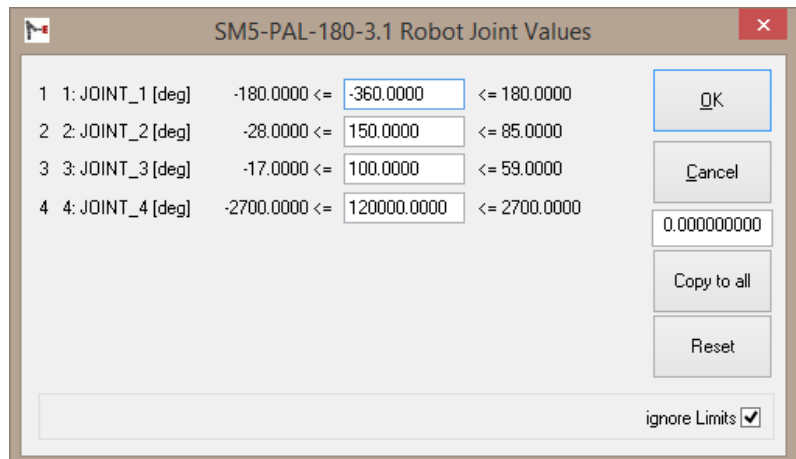
Große Buttons im chichen Style sind praktisch und fördern die User-Freundlichkeit.



Auswahl an überarbeiteten Dialogen

## Ignore Limits

Ab Version 7.3 können Zahlenwerte, in Dialogen mit Werteeingaben, ohne das Überprüfen von Limits (gültige Zahlenbereiche) eingegeben werden. Dies fördert ein noch besseres Simulationserlebnis. Man kann Werte völlig frei ausprobieren und ist in seinem eigenen Vorgehen nicht mehr eingeschränkt. Dieses kann u.a. sinnvoll sein wenn ein Roboter außerhalb seiner Verfahrbereichsgrenzen getestet werden soll.

Joint	Value	Lower Limit	Upper Limit
1: JOINT_1 [deg]	-360.0000	-180.0000	180.0000
2: JOINT_2 [deg]	150.0000	-28.0000	85.0000
3: JOINT_3 [deg]	100.0000	-17.0000	59.0000
4: JOINT_4 [deg]	120000.0000	-2700.0000	2700.0000

ignore Limits ☒

Werte außerhalb des gültigen Wertebereiches im Achswinkeldialog

Durch die praktische CheckBox „ignore Limits“ kann ad hoc das Überwachen von Limits ohne große Konfiguration umgangen werden.

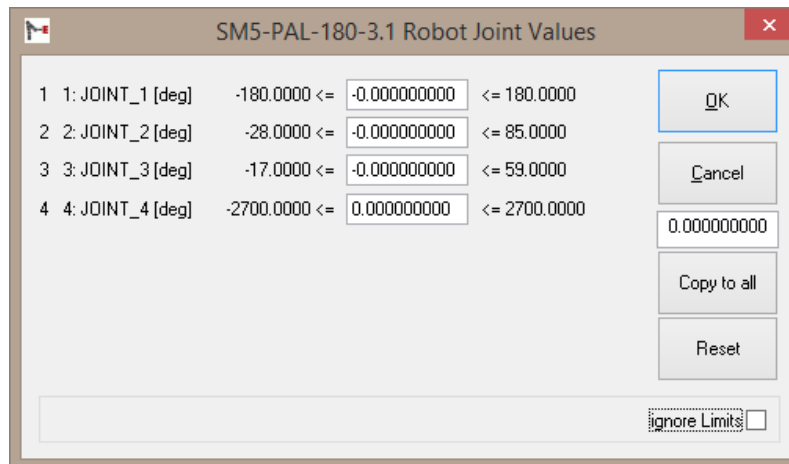
Das Setzen der CheckBox „ignore Limits“ wird auf die gesamte Werteeingabe im Dialog angewandt. Wird der Dialog geschlossen, so wird automatisch das Überwachen der Limits wieder aktiviert. Mit diesem Mechanismus will man ein bewusstes Ausschalten der Überwachung bei den Benutzern sicherstellen.

## Verfeinerte Werteingabe

Für ein verbessertes Simulationsergebnis wurden die Werteeingaben verfeinert.

Durch die Anhebung der Werteeingabe auf standardmäßig 8 Stellen hinter dem Komma bzw. sogar 9 Stellen, falls der Wert kleiner als 0.001 ist soll eine genauere Simulation ermöglicht werden.

Natürlich können Werte noch realistischer in die Simulation übernommen werden



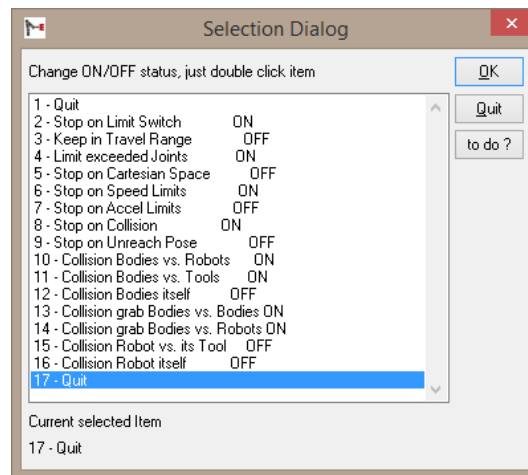
Joint	Min	Current	Max
1: JOINT_1 [deg]	-180.0000	-0.000000000	180.0000
2: JOINT_2 [deg]	-28.0000	-0.000000000	85.0000
3: JOINT_3 [deg]	-17.0000	-0.000000000	59.0000
4: JOINT_4 [deg]	-2700.0000	0.000000000	2700.0000

Neue Werteeingabe mit 9 Nachkommastellen

## CheckBoxen für Eingabe Dialoge

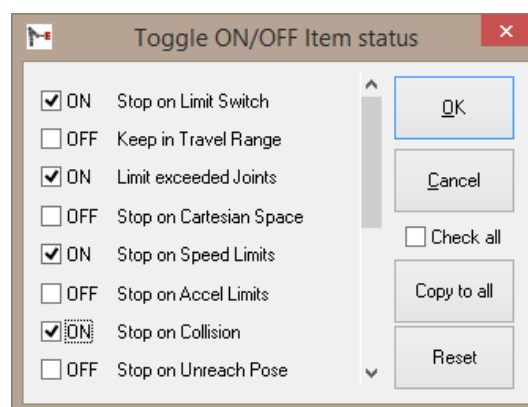
Auch hier stand bei während der Entwicklung ein verbessertes Simulationserlebnis im Fokus.

Bei Dialogen, die eine Mehrfachauswahl zulassen, z.B. der STOP On/Off Dialog, musste bisher der Anwender stets eine Option per Doppelklick oder mit dem „OK“ Button aktivieren. Hatte man seine Auswahl getroffen musste der Dialog dann noch geschlossen werden. Alles in allem war dies recht umständlich.



Alter STOP On/Off Dialog

Durch den Einsatz von CheckBoxen kann nun bequem und schnell eine Mehrfachauswahl getroffen und mit dem „OK“ Button direkt umgesetzt werden. Gleichzeitig wird der Dialog nach der Bestätigung durch „OK“ geschlossen, was auch an dieser Stelle erwartet wurde. Zusätzlich wird der Status der Auswahl angezeigt und gibt Benutzern unmittelbar ein Feedback zu Ihren Einstellungen.



Neuer optimierter STOP On/Off Dialog

### „Check All“ wählt alle Optionen an oder ab

Als absolut praktisch erweist sich der „Check all“ Button mit „Copy to all“. Hier können alle Optionen gesetzt bzw. neutralisiert werden.

## Robotics Simulation Kernel Klasse ERK\_CAPI

### Satzvorlauf Methode "erSET\_NEXT\_TARGET\_ADVANCE()"

Ein Satzvorlauf kann durch Aufruf der Methode "erSET\_NEXT\_TARGET\_ADVANCE()" erreicht werden. Entscheidend ist, dass die Methode vor dem Aufruf von "erSET\_NEXT\_TARGET()" aufgerufen werden muss.

Mit dem Satzvorlauf wird z.B. ein frühzeitiges Abbremsen auf der Bahn erreicht, wenn die programmierte Bahngeschwindigkeit des Folgesatzes geringer ist. Als Effekt ergibt sich eine längere Taktzeit, die der realen Bewegungsausführung des Roboters näher kommt.

#### ◆ erSET\_NEXT\_TARGET\_ADVANCE()

```
static DLLAPI int ER_STDCALL
ERK_CAPI_MOP_PREP::erSET_NEXT_TARGET_ADVANCE ( ER_HND er_hnd,
NEXT_TARGET_DATA_ADVANCE * p_next_target_data_advance
) static
```

Sends about next target data

The function gives information about the about next target, stored in **NEXT\_TARGET\_DATA\_ADVANCE**

#### Remarks

This function should be called once, just before calling **erSET\_NEXT\_TARGET()**

Parameter **reset** sets all data in **NEXT\_TARGET\_DATA\_ADVANCE** to default values.

#### Parameters

[in] **er\_hnd** unique kinematics handle **ER\_HND**  
[in] **p\_next\_target\_data\_advance** about next target data **NEXT\_TARGET\_DATA\_ADVANCE**

#### Return values

0 - OK, valid about next target data  
1 - Error, in valid about next target data

Zusätzlich muss der Satzvorlauf mit der Methode "erSET\_ADVANCE\_MOTION()" aktiviert werden.

#### ◆ erSET\_ADVANCE\_MOTION()

```
DLLAPI long ER_STDCALL erSET_ADVANCE_MOTION ( ER_HND er_hnd,
long Number_of_motion
)
```

Defines the number of motions, the motion planner may run in advance of the interpolator (look\_ahead).

Opcode 127, Chapter 3.4.4, Page 3-67.

#### Parameters

[in] **er\_hnd** unique kinematics handle **ER\_HND**  
[in] **Number\_of\_motion**

#### Return values

0 - OK  
1 - Error



## Satzvorlauf Struktur "NEXT\_TARGET\_DATA\_ADVANCE"

Die Daten in der NEXT\_TARGET\_DATA\_ADVANCE Struktur müssen entsprechend gesetzt werden.

Contains target data for about next move "advance move" This structure contains all required data for the about next target "advance target" Usage with `erSET_NEXT_TARGET()` [More...](#)

```
#include <erk_capi_types.h>
```

### Public Attributes

long	<b>AdvanceParam</b>	1 valid data specified, 0 no data specified, 2 initializes and set default data <a href="#">More...</a>
TErTargetID	<b>TargetID</b>	Identifier of the target, 0 no identifier given, >0 identifier of this target. <a href="#">More...</a>
long	<b>MotionType</b>	Motion Type <code>ER_JOINT</code> = <code>ER_PTP</code> = 1, <code>ER_LIN</code> = 2, <code>ER_SLEW</code> = 3, <code>ER_CIRC</code> = 4. <a href="#">More...</a>
long	<b>target_type</b>	Target Type <code>ER_TARGET_TYPE_ABS</code> or <code>ER_TARGET_TYPE_ABSJOINT</code> . <a href="#">More...</a>
double	<b>speed_value</b>	Speed for cartesian motion [m/sec]. <a href="#">More...</a>
double	<b>speed_ori_value</b>	Speed for the orientation during cartesian motion [rad/sec]. <a href="#">More...</a>
double	<b>joint_speed_percent</b>	Joint speed percentage [1%..200%] for joint motion. <a href="#">More...</a>
double	<b>segment_length</b>	Segment length to Target location [m]. <a href="#">More...</a>
long	<b>flyby_type</b>	Flyby Type <code>ER_FLYBY_TYPE_UNDEF</code> = 0, <code>ER_FLYBY_TYPE_SPEED</code> = 1, <code>ER_FLYBY_TYPE_DISTANCE</code> = 2. <a href="#">More...</a>
double	<b>flyby_value</b>	Flyby or zone value, depending on flyby type as percentage value [0%..100%], or distance [mm]. <a href="#">More...</a>
double	<b>bending_angle_value</b>	Bending angle [rad]. <a href="#">More...</a>
long	<b>TargetPosSet</b>	0x0 - not set, 0x1 - CartPos, 0x2 - CartPosVia, 0x4 - JointPos <a href="#">More...</a>
DFRAME	<b>CartPos</b>	Target cartesian position for cartesian motion or joint motion with target type <code>ER_TARGET_TYPE_ABS</code> . <a href="#">More...</a>
DFRAME	<b>CartPosVia</b>	Target cartesian via position for cartesian motion <code>ER_CIRC</code> . <a href="#">More...</a>
double	<b>JointPos [ER_KIN_DOFS]</b>	Target joint position for joint motion and target type <code>ER_TARGET_TYPE_ABSJOINT</code> . <a href="#">More...</a>

## 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

```
// Constants
const long DS_MAXSTR          = 2048;    ///< Maximum length of a double huge
        message

const int DOF12                = 12;      ///< 12 Degrees-of-Freedom

// class ER_CAPI_SYS_STATUS;
const int SYS_SYSTEM_FOLDER_DIRECTORY_UNDEF    = 0; ///< Request Idx
        undefined
const int SYS_SYSTEM_FOLDER_DIRECTORY_LIST      = 1; ///< List all system
        folder and directories in a message window
const int SYS_SYSTEM_FOLDER_DIRECTORY_CONFIG_FILE = 10; ///< configuration
        file 'config.dat', resides in Application directory
const int SYS_SYSTEM_FOLDER_DIRECTORY_LICENSE_FILE = 11; ///< license file
        'license.dat', resides in 'LICENSE=' definition in config.dat
const int SYS_SYSTEM_FOLDER_DIRECTORY_MONITORING_FILE = 12; ///< monitoring
        file 'moni_msg.txt', resides in TMPDIR
const int SYS_SYSTEM_FOLDER_DIRECTORY_ENVIRONMENT_FILE= 13; ///< environment
        file 'easy-rob.env', resides in USRDIR
const int SYS_SYSTEM_FOLDER_DIRECTORY_WORKING_PATHES_FILE= 14; ///< working
        pathes file 'easy-rob.pth', resides in USRDIR
const int SYS_SYSTEM_FOLDER_DIRECTORY_LOADFROMLIB_FILE= 15; ///< Device
        Manager: Load from Library file 'er_LoadFromLibPb.ini', resides in
        USRDIR
const int SYS_SYSTEM_FOLDER_DIRECTORY_LOADFROMLIB_PREFERRED_FILE= 16; ///<
        Device Manager: Load from Library Preferred file
        'er_LoadFromLibPb_prefered.ini', resides in USRDIR
const int SYS_SYSTEM_FOLDER_DIRECTORY_APPLICATION_DIR= 101; ///< Application
        directory
const int SYS_SYSTEM_FOLDER_DIRECTORY_WORKING_DIR    = 102; ///< current
        working directory
const int SYS_SYSTEM_FOLDER_DIRECTORY_USR_DIR        = 103; ///< User
        directory, defined in config.dat, individual or depending on
        environment variable %USERPROFILE%
const int SYS_SYSTEM_FOLDER_DIRECTORY_TMP_DIR        = 104; ///< Temp
        directory, defined in config.dat, individual or depending on
        environment variable %TMP%
const int SYS_SYSTEM_FOLDER_DIRECTORY_KERNEL_IO_DIR  = 105; ///< Kernel IO,
        defined in config.dat
const int SYS_SYSTEM_FOLDER_DIRECTORY_EDIT_PRG       = 201; ///< Editor
        program, defined in config.dat
```

```
// System Routines
// API-UserDLL
const int USER_DLL_CALLBACK_AUXUPDATE_MAX = 12; ///< Max number of
        callback fct for AuxUpdate, see set_callback_AuxUpdate
const int USER_DLL_CALLBACK_PROGLINEUPDATE_MAX = 12; ///< Max number of
        callback fct for ProglineUpdate, see set_callback_ProglineUpdate
```

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

### SYS\_STATUS : System Dateien

char \* **ER\_CAPI\_SYS\_STATUS::get\_system\_folder\_file** ()

#### ◆ get\_system\_folder\_file()

static ER\_DLLExport char\*

ER\_CAPI\_SYS\_STATUS::get\_system\_folder\_file

( int request\_idx = SYS\_SYSTEM\_FOLDER\_DIRECTORY\_UNDEF )

static

Supplies requested system path or file. Parameter *request\_idx* is one of **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_UNDEF**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_LIST**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_CONFIG\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_LICENSE\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_MONITORING\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_ENVIRONMENT\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_WORKING\_PATHES\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_LOADFROMLIB\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_LOADFROMLIB\_PREFERRED\_FILE**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_APPLICATION\_DIR**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_WORKING\_DIR**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_USR\_DIR**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_TMP\_DIR**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_KERNEL\_IO\_DIR**, **SYS\_SYSTEM\_FOLDER\_DIRECTORY\_EDIT\_PRG**

#### Parameters

[in] **request\_idx**

#### Return values

**NULL** - in valid request

**else** - pointer to requested string of path or file name

## USER\_IO\_DIALOG: Checkbox Dialog und StringSingle Dialog

int **ER\_CAPI\_USER\_IO\_DIALOG::dialog\_checkbox** ()

### ◆ dialog\_checkbox()

```
static ER_DLLExport int ER_CAPI_USER_IO_DIALOG::dialog_checkbox ( char *  title,
                                                                int      n,
                                                                char ** ww,
                                                                int *   iv,
                                                                char ** ww_on = NULL,
                                                                char ** ww_off = NULL
                                                                )
```

static

Opens CheckBox dialog

This Dialog allows to check or uncheck the status for several items in the string array ww.

```
// Example:
char title_txt[MAXSTR] = {"Change Item status"};
char *ww[] = {"Item1", "Item2", "Item3"};
int n = sizeof(ww)/sizeof(ww[0]);           // get number of Items
int iv = 1;                                  // check
char *ww_on[] = {"On"};
char *ww_off[] = {"Off"};
int ret=er_user_io_dialog.dialog_checkbox(title_txt,n,ww,&iv,ww_on,ww_off);
if (ret==1)
{
    er_user_io_dialog._info_line_msg(0,"Item status is %s",iv?"ON:"OFF");
}
...
```

#### Parameters

- [in] **title** dialog title
- [in] **n** number of items
- [in] **ww** array of strings, size n
- [in] **iv** array of item status to check or uncheck, size n
- [in] **ww\_on** array of strings when item checked, size n
- [in] **ww\_off** array of strings when item unchecked, size n

#### Return values

- 1 - OK
- 1 - **ER\_CANCEL**, dialog cancelled

int ER\_CAPI\_USER\_IO\_DIALOG::dialog\_string\_single ()

#### ◆ dialog\_string\_single()

```
static ER_DllExport int ER_CAPI_USER_IO_DIALOG::dialog_string_single ( char * title,  
                                                                    char * header,  
                                                                    char * footer,  
                                                                    char * v,  
                                                                    int len,  
                                                                    int pw = 0  
                                                                    )
```

static

Opens dialog to enter a single string

This Dialog allows to enter a string v.

#### Parameters

[in]	<b>title</b>	dialog title
[in]	<b>header</b>	dialog header information
[in]	<b>footer</b>	dialog footer information
[in,out]	<b>v</b>	string value to modify
[in]	<b>len</b>	maximum length for each v
[in]	<b>pw</b>	password, hide text if true

#### Return values

- 1 - OK
- 1 - **ER\_CANCEL**, dialog cancelled

## SIM\_MONITORING: Collision Methods

Neue Methoden erlauben einen effizienten Kollisionstest zwischen Devices (=Robots).  
Es können z.B. explizit zwei Devices auf Kollision oder auch ein Device gegen alle anderen Devices in der Arbeitszelle auf Kollision getestet werden.

`int ER_CAPI_SIM_MONITORING::chk_collision_workcell ()`  
=> identisch mit dem Aufruf: `chk_limits (AUX_UPDATE_IDX_COLLISION)`

### ◆ `chk_collision_workcell()`

`static ER_DllExport int ER_CAPI_SIM_MONITORING::chk_collision_workcell ( )` static

Check Workcell Collision

Test all devices versus all other devices, including environment geometries

Current settings, such as device visibility, "reference collision" or geometry collision tolerance, etc. are considered

Use `get_collision_workcell_msg()` to retrieve information about colliding devices and objects, in case of collision.

#### Return values

- 0 - no collision
- 1 - collision detected

`int ER_CAPI_SIM_MONITORING::chk_collision_devices_idx ()`

### ◆ `chk_collision_devices_idx()`

`static ER_DllExport int ER_CAPI_SIM_MONITORING::chk_collision_devices_idx ( int dev_idx_1,  
int dev_idx_2  
)` static

Check for collision between two devices

The device index `dev_idx_1`, `dev_idx_2` must be in `[1..n_dev]` and not equal

If `dev_idx_1` or `dev_idx_2` are not valid, -1 is returned

In case `dev_idx_1` and/or `dev_idx_2` are 0, the specified device will be tested versus all other devices

See also `chk_collision_workcell()`, `chk_collision_devices_uid()`, `chk_collision_devices_name()`

#### Parameters

- [in] `dev_idx_1` idx 1st device `[1..n_devices]`
- [in] `dev_idx_2` idx 2nd device `[1..n_devices]`

#### Return values

- 1 - collision occurred
- 0 - no collision
- 1 - error

**int ER\_CAPI\_SIM\_MONITORING::chk\_collision\_devices\_uid ()**

◆ **chk\_collision\_devices\_uid()**

```
static ER_DllExport int ER_CAPI_SIM_MONITORING::chk_collision_devices_uid ( ER_UID dev_uid_1,
                                                                           ER_UID dev_uid_2
                                                                           )
```

Check for collision between two devices  
If dev\_uid\_1 or dev\_uid\_2 are not valid, -1 is returned  
In case dev\_uid\_1 and/or dev\_uid\_2 are 0, the specified device will be tested versus all other devices  
See also [chk\\_collision\\_workcell\(\)](#), [chk\\_collision\\_devices\\_idx\(\)](#), [chk\\_collision\\_devices\\_name\(\)](#)

**Parameters**

[in] **dev\_uid\_1** unique id 1st device  
[in] **dev\_uid\_2** unique id 2nd device

**Return values**

1 - collision occurred  
0 - no collision  
-1 - error

**int ER\_CAPI\_SIM\_MONITORING::chk\_collision\_devices\_name ()**

◆ **chk\_collision\_devices\_name()**

```
static ER_DllExport int ER_CAPI_SIM_MONITORING::chk_collision_devices_name ( char * dev_name_1,
                                                                           char * dev_name_2
                                                                           )
```

Check for collision between two devices  
If dev\_name\_1 or dev\_name\_2 are not valid, -1 is returned  
In case dev\_name\_1 and/or dev\_name\_2 are NULL, the specified device will be tested versus all other devices  
See also [chk\\_collision\\_workcell\(\)](#), [chk\\_collision\\_devices\\_uid\(\)](#), [chk\\_collision\\_devices\\_idx\(\)](#)

**Parameters**

[in] **dev\_name\_1** device name 1st device  
[in] **dev\_name\_2** device name 2nd device

**Return values**

1 - collision occurred  
0 - no collision  
-1 - error

**int ER\_CAPI\_SIM\_MONITORING::get\_collision\_workcell\_msg ()**

=> identisch mit dem Aufruf: [get\\_warning\\_msg \(AUX\\_UPDATE\\_IDX\\_COLLISION, msg\)](#)

◆ **get\_collision\_workcell\_msg()**

```
static ER_DllExport int ER_CAPI_SIM_MONITORING::get_collision_workcell_msg ( char * msg )
```

Collision message in case of detected collision  
Call [chk\\_collision\\_workcell\(\)](#) first to detect workcell collision.

**Parameters**

[out] **msg** collision message, size **DMASTR**

**Return values**

0 - valid message  
1 - invalid

## SYS\_USERDLL: Callback Funktionen für AuxUpdate(...) und ProgLineUpdate(...)

Ein Export von AuxUpdate(int idx, int sub\_idx) bzw. ProgLineUpdate(char \*cmd) aus der API-UserDLL heraus ist nun nicht bindend. Stattdessen können zur Laufzeit die Callback-Funktionen flexible definiert bzw. getrennt werden.

Bis zu USER\_DLL\_CALLBACK\_AUXUPDATE\_MAX = 12 und USER\_DLL\_CALLBACK\_PROGLINEUPDATE\_MAX = 12 sind jeweils definierbar.

int ER\_CAPI\_SYS\_USERDLL::set\_callback\_AuxUpdate ()

### ◆ set\_callback\_AuxUpdate()

```
static ER_DllExport int ER_CAPI_SYS_USERDLL::set_callback_AuxUpdate ( int          callback_fct_id,
                                                                    int(*) (int idx, int sub_idx) callback_AuxUpdate
                                                                    )
```

Defines callback fct pointer array for AuxUpdate

Parameter callback\_fct\_id is zero based in [0..USER\_DLL\_CALLBACK\_AUXUPDATE\_MAX-1]

Parameter callback\_AuxUpdate points to callback function.

Prototype: int MyAuxUpdate\_fct (int idx, int sub\_idx)

To unset the callback fct, set callback\_AuxUpdate = NULL.

#### Parameters

[in] **callback\_fct\_id** in [0..USER\_DLL\_CALLBACK\_AUXUPDATE\_MAX-1]

[in] **callback\_AuxUpdate** (int, int) callback function pointer

#### Return values

0 - Ok

1 - Error

int ER\_CAPI\_SYS\_USERDLL::set\_callback\_ProgLineUpdate ()

### ◆ set\_callback\_ProglineUpdate()

```
static ER_DllExport int ER_CAPI_SYS_USERDLL::set_callback_ProglineUpdate ( int          callback_fct_id,
                                                                    int(*) (char *propline) callback_ProglineUpdate
                                                                    )
```

Defines callback fct pointer array for ProglineUpdate

Parameter callback\_fct\_id is zero based in [0..USER\_DLL\_CALLBACK\_PROGLINEUPDATE\_MAX-1]

Parameter callback\_ProglineUpdate points to callback function.

Prototype: int MyProglineUpdate\_fct (char \*propline)

To unset the callback fct, set callback\_ProglineUpdate = NULL.

#### Parameters

[in] **callback\_fct\_id** in [0..USER\_DLL\_CALLBACK\_PROGLINEUPDATE\_MAX-1]

[in] **callback\_ProglineUpdate** (char \*) callback function pointer

#### Return values

0 - Ok

1 - Error



## Namensänderungen von API-Methods

Die folgenden API-Methods ersetzen obsoletere API-Methods, siehe Kapitel "Linkage ersetzt Synchronize". Wir empfehlen die neuen Methoden zu verwenden.

```
static ER_DllExport int ER_CAPI_DEVICES::Device_Link_by_idx (  
    int new_reference_type,  
    int new_reference_device_idx = 0,  
    int * new_reference_jnt_link_idx = NULL  
)
```

ersetzt **Device\_Sync\_by\_idx (.)**

```
static ER_DllExport int ER_CAPI_DEVICES::Device_Link_by_name (  
    int new_reference_type,  
    char * new_reference_device_name = NULL,  
    int * new_reference_jnt_link_idx = NULL  
)
```

ersetzt **Device\_Sync\_by\_name (.)**

```
static ER_DllExport int ER_CAPI_DEVICES::Device_Link_by_uid (  
    int new_reference_type,  
    ER_UID new_reference_device_uid = 0,  
    int * new_reference_jnt_link_idx = NULL  
)
```

ersetzt **Device\_Sync\_by\_uid (.)**

```
static ER_DllExport ER_UID* ER_CAPI_DEVICES::inq_device_link_ref_sys_grp_uid ( void )
```

ersetzt **inq\_device\_sync\_ref\_sys\_grp\_uid (.)**

```
static ER_DllExport int* ER_CAPI_DEVICES::inq_device_link_ref_sys_jnt_link_idx ( void )
```

ersetzt **inq\_device\_sync\_ref\_sys\_jnt\_link\_idx (.)**

```
static ER_DllExport int* ER_CAPI_DEVICES::inq_device_link_ref_sys_type ( void )
```

ersetzt **inq\_device\_sync\_ref\_sys\_type (.)**

```
static ER_DllExport char* ER_CAPI_DEVICES::inq_device_link_ref_sys_type_name ( void )
```

ersetzt **inq\_device\_sync\_ref\_sys\_type\_name (.)**

## 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:

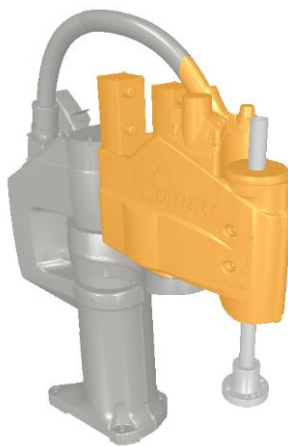
<http://www.easy-rob.com/produkt/erweiterungen/roboter-bibliotheken.html>

Hier eine kleine Auswahl an neuen Modellen, welche zu den zahlreichen, bestehenden Robotern neu dazugekommen sind:

- von Comau der Racer 3
- von Stäubli der Scara TS 40 STD FL
- von FANUC der CR 35 iA



Comau Racer 3



Stäubli TS 40 STD FL



FANUC CR 35 iA

### Wichtig:

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

## Dual-Arm Robots

Cobots sind in der Mensch-Roboter-Kollaboration (MRK) als fleißige Helfer nicht mehr wegzudenken. Dual Arm Roboter werden diesen Siegeszug fortführen und als nächste Stufe u.a. komplexe Montagetätigkeiten übernehmen. Deshalb widmet EASY-ROB diesem Thema besondere Aufmerksamkeit.

Simulieren Sie schon heute die Aufgaben von morgen- mit EASY-ROB™ zukunftsicher neuen Aufgaben begegnen!

EASY-ROB wird von nun an kontinuierlich Dual-Arms in einer optionalen Roboterbibliothek aufnehmen. Diese sind als RAS-Dateien (Robot Assembly) abgelegt.

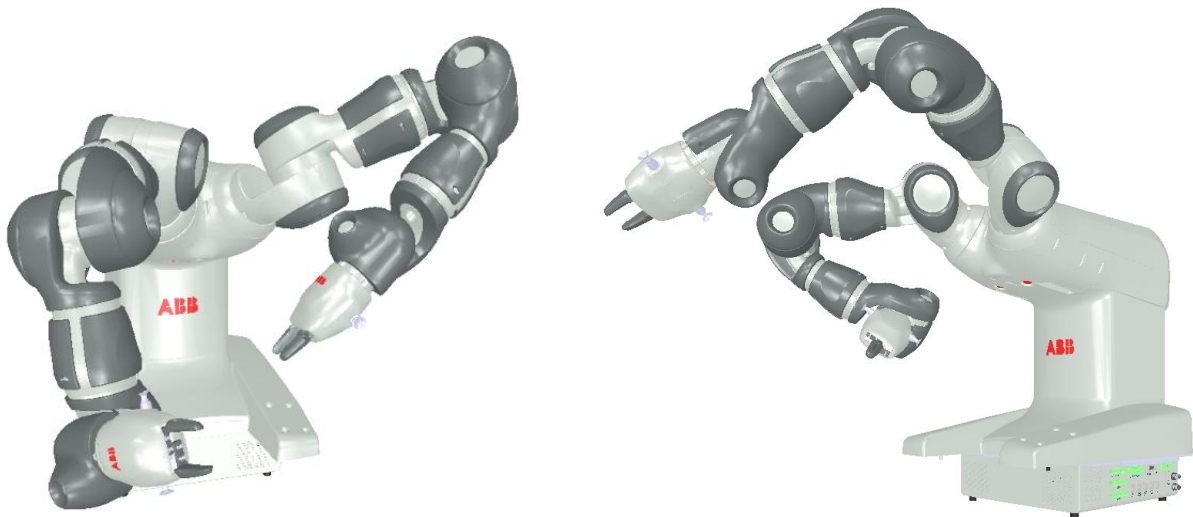
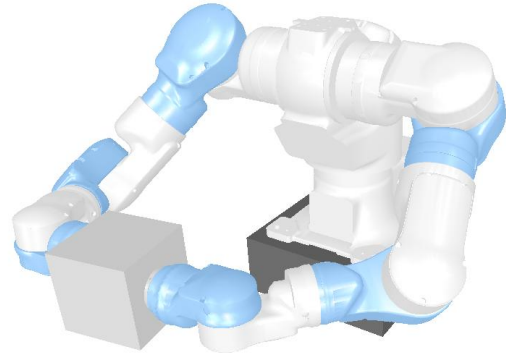
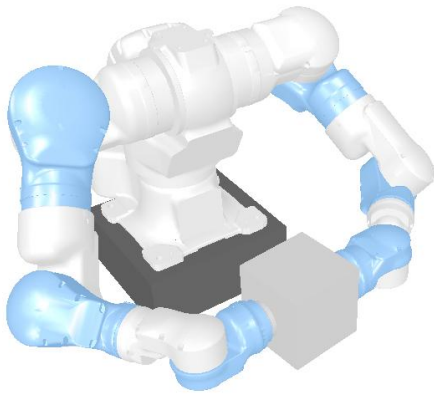
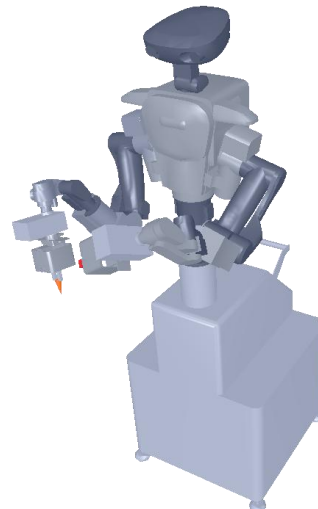
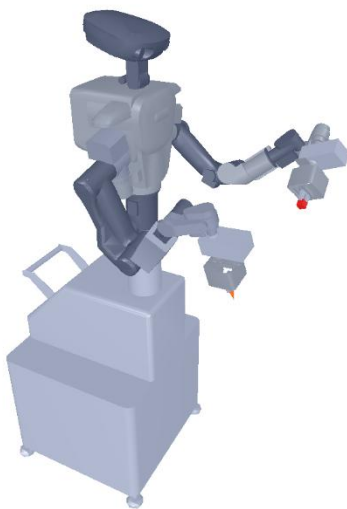


ABB Yumi



YASKAWA SDA20F

Auch Modelle aus der Service Robotik sind enthalten:



KAWADA NEXTAGE

## Neue ERPL- und ERCL-Befehle

Eine ausführliche Beschreibung sämtlicher ERPL- und ERCL-Befehle kann dem Dokument EASY-ROB-ERPL\_DE.pdf entnommen werden.

### ERCL - Linkage Device

**ERC LINKAGE DEVICE SET** 'DeviceName' AIdx(1) .. AIdx(n)

Koppelt im Simulationslauf das aktuelle Gerät mit einem anderen Gerät 'DeviceName', wobei die Parameter AIdx(1) .. AIdx(n) den jeweiligen AchsIndex für die Kopplung definieren.

**ERC LINKAGE DEVICE UNSET** ['DeviceName']

Löscht die bestehende Kopplung des aktuellen Geräts zu einem anderen Gerät 'DeviceName', sofern eine Kopplung existiert.

### ERCL - Display Device

**ERC DISPLAY DEVICE** 'DeviceName' ON/OFF

Ein-/Ausschalten der Visualisierung des Geräts 'DeviceName'

### Parser Funktionen

**collision\_devices\_idx** (dev\_idx1, dev\_idx2)

Prüft ob das 1. Gerät mit dem Geräte-Idx dev\_idx1 mit dem 2. Gerät dev\_idx2 kollidiert. Die Parameter dev\_idx1 und dev\_idx2 können flexibel gesetzt werden, so dass z.B. das 1. Gerät gegen alle anderen Geräte dev\_idx2 = 0 in der Arbeitszelle auf Kollision geprüft wird.

**collision\_devices\_name** ("DeviceName1", "DeviceName2")

Prüft ob das 1. Gerät mit dem Gerätenamen "DeviceName1" mit dem 2. Gerät "DeviceName2" kollidiert. Die Parameter "DeviceName1" und "DeviceName2" können flexibel gesetzt werden, so dass z.B. das 1. Gerät gegen alle anderen Geräte "DeviceName2" = "0" in der Arbeitszelle auf Kollision geprüft wird.

**get\_device\_idx** ("DeviceName")

Gibt den Geräte-Idx im Bereich [1 bis n=Geräteanzahl] des Geräts mit dem Namen "DeviceName" zurück. Der Parameter "DeviceName" kann flexibel gewählt werden, so dass z.B. "" oder () den Geräte-Idx des aktuellen Geräts zurückgibt. Bei Fehler wird 0 zurückgegeben.

## Kontakt

### EASY-ROB Software GmbH

Adresse: Hauptstr. 42  
65719 Hofheim am Taunus  
Germany

Kontakt: Stefan Anton, Patryk Lischka

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