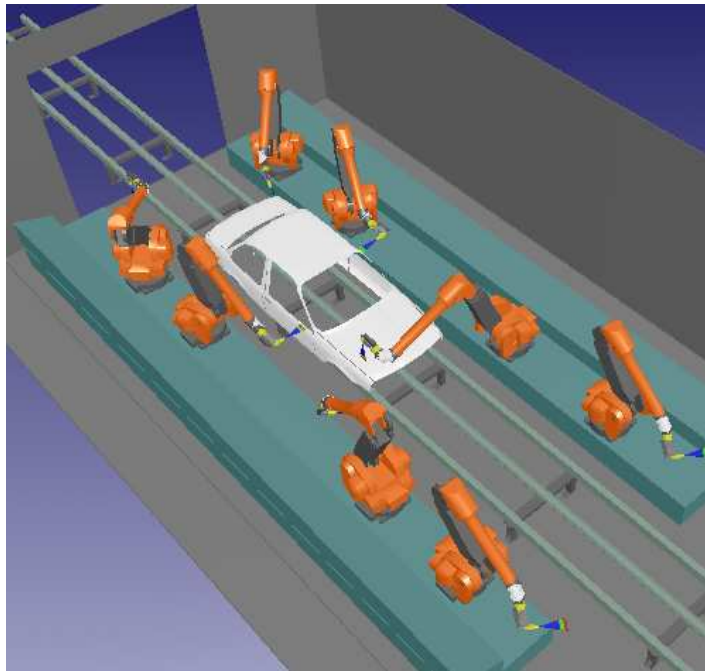


Update

EASY-ROB™ V4.606



März 2008

Version 1.0

EASY-ROB™

Inhaltsverzeichnis

Einleitung	5
Der Project Manager	7
Projektübersicht	8
Detaillierte Informationen	8
Screenshots anzeigen und Video abspielen	9
Laden einer Arbeitszelle ins EASY-ROB™	9
Neue Funktionalität im Visual File Interface	10
Pfadhistorie	10
Bevorzugte Pfade	11
Auswahl des anzuzeigenden Dateityps	11
Sortieren nach Datum	12
Screenshots als JPG und BMP	12
Projektdateien zum Project Manager senden	13
Screenshots mit Capture Image erstellen	14
Optimierung der Merge Funktion	15
Sonstige Erweiterungen	16
Toolbar „Select Robot“	16
Menüleiste in Dialogen	16
Erweiterung in Teach Window Dialogen	17
ERC Searcher	18
Neue Arbeitszellen-Beispiele	19
Erweitertes Tutorial zum Thema Signale	19
ERPL-Befehle	20
API-Funktionen	21
Bewegungsplanung	21
Roboterattribute	22
Speichern	23
Mouse Mode	24
Mouse Picking	24
AuxUpdate	26
Kontakt	27
Eigene Notizen	29

EASY-ROB™ V4.606

Einleitung

Mit der aktuellen EASY-ROB™ Version 4.606 stehen wieder neue Funktionalitäten und Verbesserungen zur Verfügung.

Neu ist der Project Manager, mit dem EASY-ROB™ Arbeitszellen projektbezogen Informationen verwaltet werden können. Der Bediener behält so eine gute Übersicht über seine Projekte.

Das Visual File Interface wurde erneut mit Funktionen erweitert. Zu der Pfadhistorie, den bevorzugten Pfaden, der Vorauswahl des anzuzeigenden Dateityps mit Sortierung nach Datum, können Projektinformationen direkt zum Project Manager gesendet und dort abgelegt werden.

Auch das EASY-ROB™ Capture Image wurde erweitert. Screenshots können zusätzlich im Grafikformat JPEG abgespeichert werden.

Um die Funktionsweise und die Programmierung mit digitalen Signalen zu verdeutlichen, wurden neue Beispielarbeitszellen erzeugt und das Tutorial mit der TrainLib um das Kapitel „Multi-Program“ erweitert.

Informationen zur Bedienung der neuen Funktionalitäten finden Sie in den überarbeiteten Bedienungshinweisen.

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

Vielen Dank



Stefan Anton

EASY-ROB
3D Robot Simulation Tool

Der Project Manager

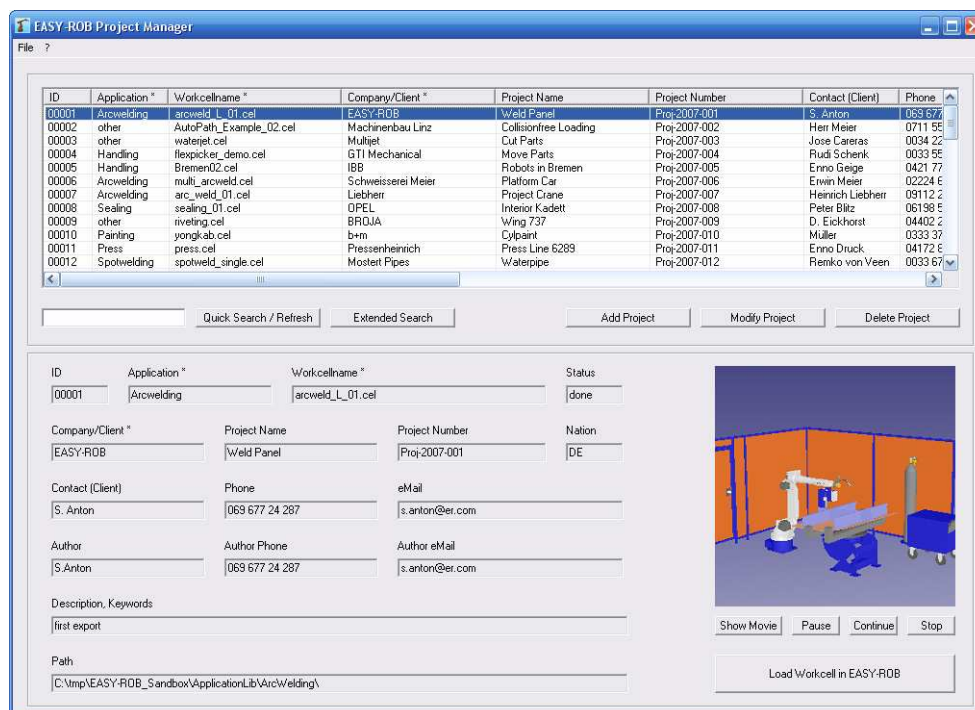
Mit dem **Project Manager** steht ab der Version EASY-ROB™ V4.606 eine neue Funktionalität zur Verfügung, die den Bediener unterstützt seine Projekte übersichtlich zu verwalten und einen schnellen Zugriff auf die Arbeitszellen zu gewährleisten.

EASY-ROB™ Arbeitszellen können zusammen mit wichtigen projektbezogenen Informationen im Project Manager verwaltet werden und lassen sich über verschiedene Suchfilter auf kürzestem Weg abrufen und laden.

Die Projektdaten werden entweder über eine Eingabemaske im Project Manager eingegeben, oder über das **Visual File Interface** gespeichert.

Zudem lassen sich Videodateien einer Arbeitszelle direkt im Project Manager abspielen.

Um den Project Manager zu starten, drücken Sie die Tastenkombination „Ctrl+Shift+P“ oder wählen Sie im Menu: File -> Load -> Start Project Manager. Die Daten des Project Managers werden in der Datei „./ProjectManagerDb/er_project_manager.dat“ gespeichert. Backup Kopien im Verzeichnis „./ProjectManagerDb/backup_db/“.



Projektübersicht

Die Übersicht zeigt alle gespeicherten Projekte in tabellarischer Form.

Neben den Funktionen zum Hinzufügen, Ändern und Löschen eines Projektes, gibt es auch Suchfunktionen mit denen mit verschiedenen Filtern nach bestimmten Einträgen gesucht werden kann.

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

Detaillierte Informationen

Bei Auswahl eines Projektes aus der Liste werden die detaillierten Informationen zum Projekt übersichtlich dargestellt.

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

Die folgenden Informationen werden dargestellt:

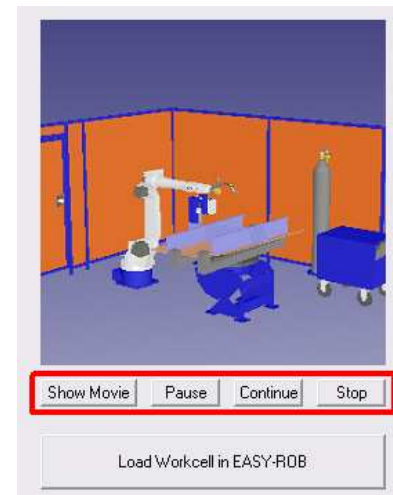
- Anwendung
- Arbeitszellenname
- Status des Projekts
- Kundenname
- Projektname
- Projektnummer
- Nation
- Kontaktperson mit Telefon und eMail
- Autor der Zelle mit Telefon und eMail
- Kurzkommentar
- absoluter Pfad der Arbeitszellendatei

Screenshots anzeigen und Video abspielen

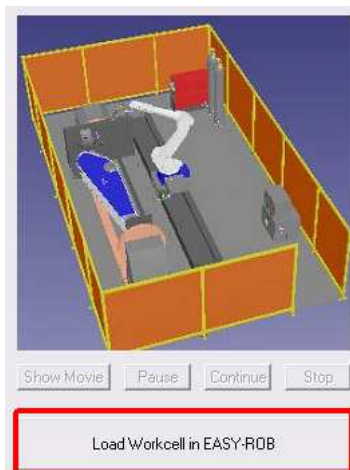
Wurde mit dem EASY-ROB™ Capture Image ein Screenshot für eine Arbeitszelle erzeugt, dann wird dieses Bild automatisch beim Auswählen des Projekts angezeigt.

Ist zusätzlich noch ein Video zu einer Arbeitszelle erzeugt worden, kann dieses Video direkt vom Project Manager aus gestartet und angesehen werden. Das Video muss im AVI-Format vorliegen und im selben Verzeichnis liegen sowie denselben Dateinamen wie die Arbeitszelle haben.

(Beispiel: MyCell.cel, MyCell.jpg, MyCell.avi)



Laden einer Arbeitszelle ins EASY-ROB™



Soll eine Arbeitszelle nochmals genauer untersucht oder die Simulation präsentiert werden, kann mit „Load Workcell in EASY-ROB“ die Arbeitszelle direkt aus dem Project Manager in eine neue EASY-ROB™ Session geladen werden.

Neue Funktionalität im Visual File Interface

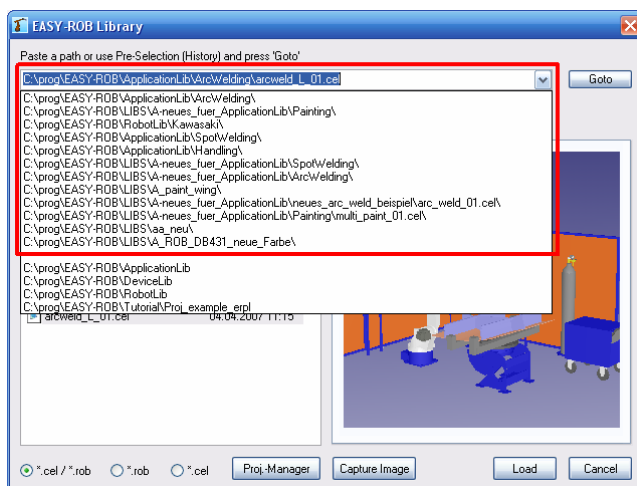
Das Visual File Interface ist nach dem kürzlich erschienenen Major-Release der Version 4.603 erneut mit neuen Funktionen erweitert worden.

Übersicht der neuen Funktionalitäten:

- Pfadhistorie
- Bevorzugte Pfade
- Auswahl des anzuzeigenden Dateityps
- Sortieren der Dateien nach Datum
- Anzeige der Screenshots als JPEG und BMP
- senden von Informationen der Arbeitszelle zum Project Manager
- neues Dateiformat für Capture Image (JPEG)

Pfadhistorie

Lädt der Benutzer eine Zelle oder einen Roboter, wird der Pfad aus dem die Datei geladen wurde in die Pfadhistorie an erster Stelle gespeichert. Maximal 12 Pfade werden abgelegt, wobei gleiche Pfadnamen nur einmal gelistet sind.

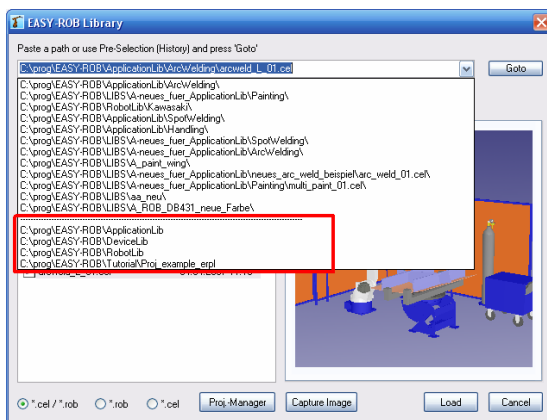


Die Pfadhistorie dient dem schnelleren und einfacheren Zugriff auf die zuletzt geladenen Zellen oder Roboter. Sie können über ein Pulldown-Menü erreicht und ausgewählt werden.

Die Datei „er_LoadFromLibPb.ini“ wird automatisch verwaltet und sollte nicht manuell editiert werden.

Bevorzugte Pfade

Zusätzlich zur automatisch erzeugten Pfadhistorie werden im selben Pulldown-Menü auch so genannte „bevorzugte Pfade“ zur Verfügung gestellt. Die „bevorzugten Pfade“ werden vom Bediener manuell angelegt, in dem er in der Datei „*er_LoadFromLibPb_prefered.ini*“ seine bevorzugten Pfade angibt. Diese Datei wird einmalig angelegt, falls sie nicht existiert. Im Unterschied zu der automatisch erzeugten Pfadhistorie, bleiben die „bevorzugten Pfade“ unverändert und ermöglichen dem Bediener so einen schnellen Zugriff auf seine 4 wichtigsten Pfade.



Die „bevorzugten Pfade“ sind vom Bediener anzulegen und werden in der Datei „*er_LoadFromLibPb_prefered.ini*“ gespeichert.

Auswahl des anzuzeigenden Dateityps

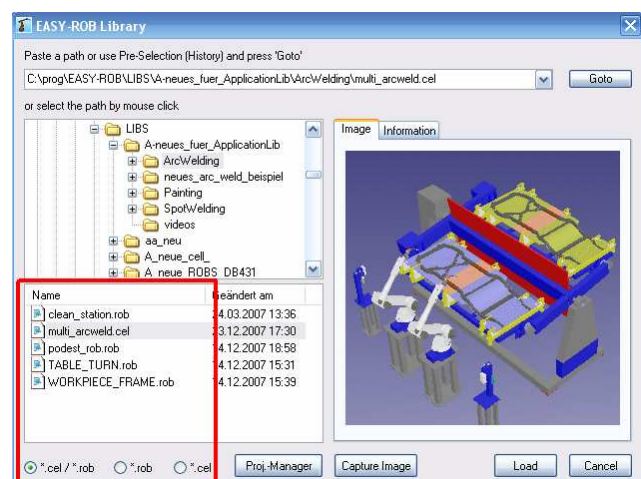
Um bei größeren Projekten, mit mehreren Zellen und Robotern, eine bessere Übersicht zu behalten und die Auswahl der gewünschten Datei zu vereinfachen, kann der Dateityp der angezeigten Dateien vor eingestellt werden.

Die drei Optionen:

- Arbeitszellen und Roboter (*.cel, *.rob)
- nur Roboter (*.rob)
- nur Arbeitszellen (*.cel)

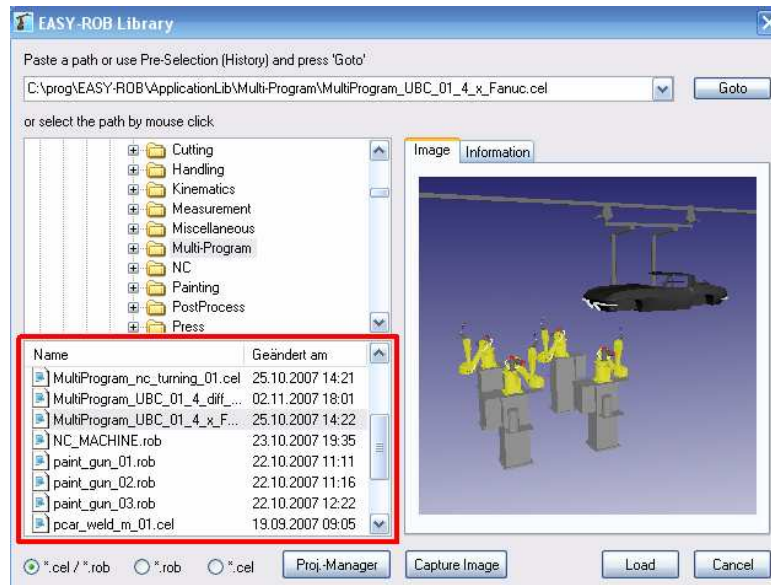
Die Defaulteinstellung beim Start von EASY-ROB™ ist „Arbeitszellen und Roboter“.

Zur Laufzeit wird der zuletzt gesetzte Filter gespeichert und beim nächsten Aufruf der Library gesetzt.



Sortieren nach Datum

Das Feld für die Anzeige der Dateien wurde um eine Datumsspalte erweitert, um Arbeitszellen und Roboter nach Namen oder nach Erstellungszeitpunkt sortieren zu können



Screenshots als JPG und BMP

Das Visual File Interface kann zusätzlich zum BMP Format auch Screenshots im JPEG Format lesen und anzeigen. Bei der Anzeige haben Screenshots im JPEG-Format Vorrang.

Sollte kein Screenshot existieren, wird ein Standardbild angezeigt.

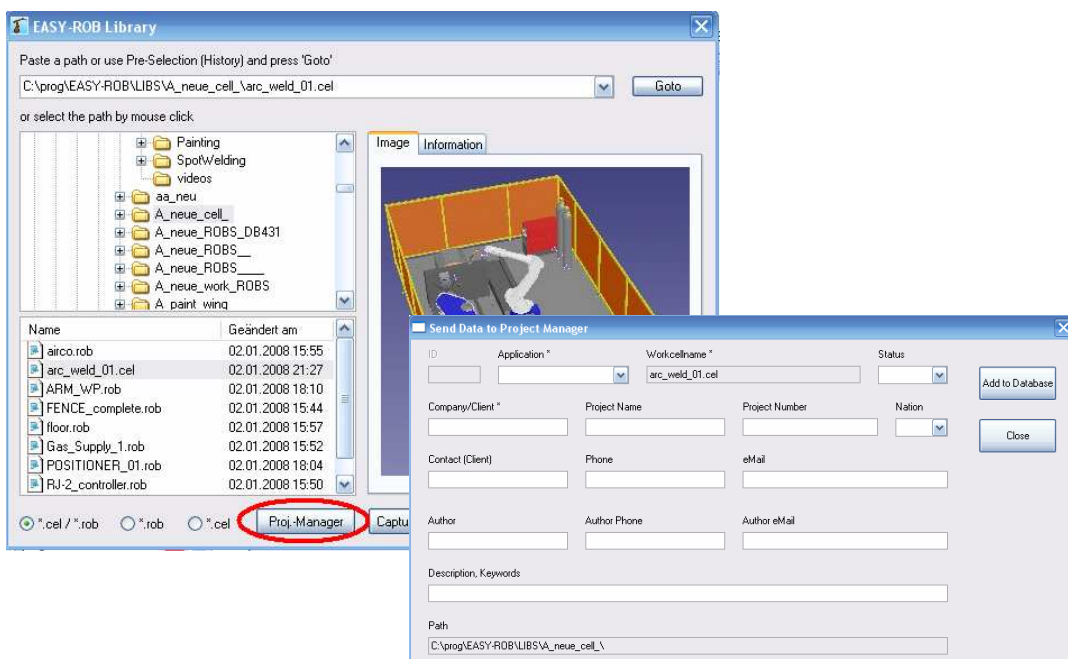
(zur Erstellung von Screenshots siehe auch „Screenshots mit Capture Image erstellen“ in diesem Dokument)

Projektdaten zum Project Manager senden

Mit dem Visual File Interface werden Projektdaten direkt dem Project Manager hinzugefügt.

Beim Auswählen einer Zelle im Visual File Interface können mit der Funktion „Proj.-Manager“ die gewünschten Projektdaten eingegeben und anschließend mit „Add to Database“ zum Project Manager gesendet werden.

Tipp: Markieren Sie zuerst die gewünschte Datei (im Bild „*arc_weld_01.cel*“) – nur bei ausgewählter Zelle kann die Funktion „Project Manager“ aktiviert werden.

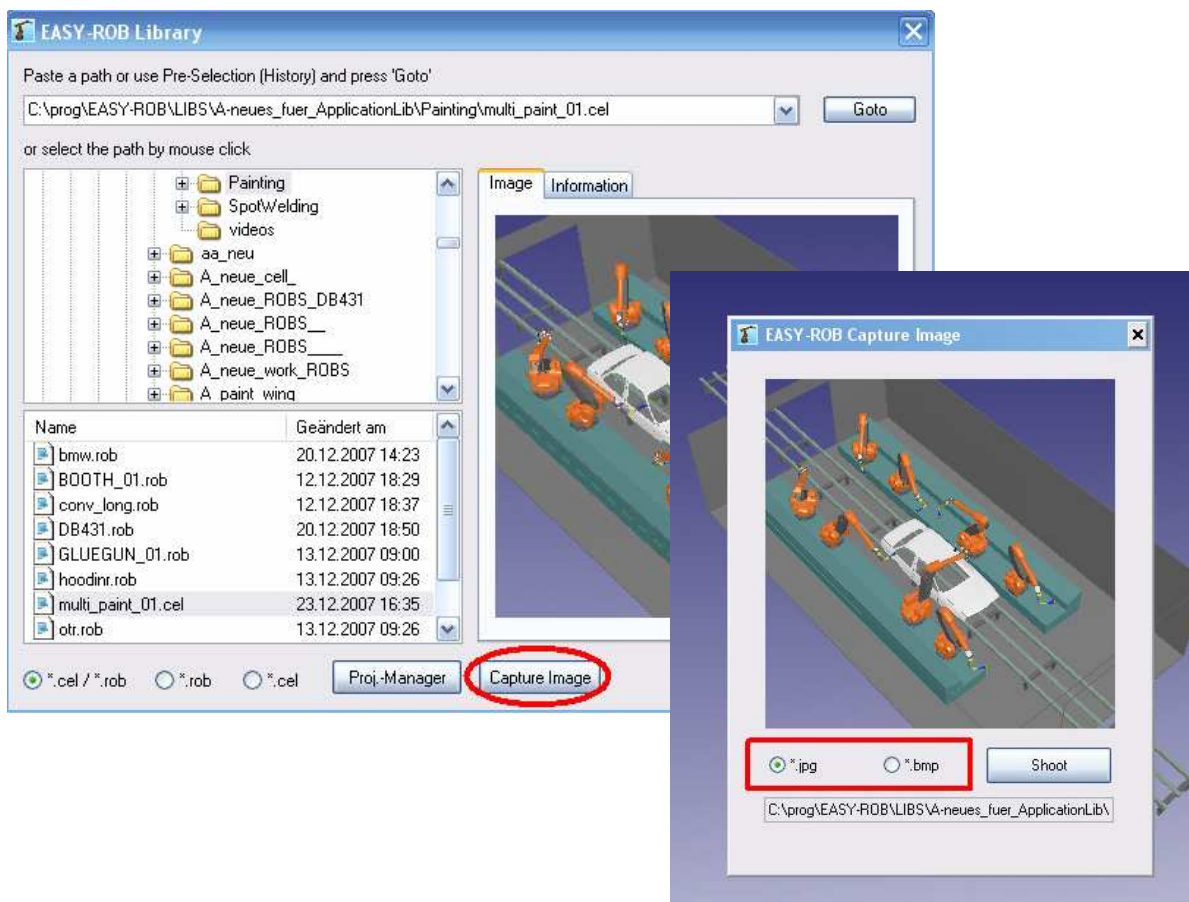


Screenshots mit Capture Image erstellen

Zusätzlich zum BMP Format lassen sich Screenshots auch im JPEG Format erzeugen.

Nach dem Laden einer Zelle/Roboter wird „Capture Image“ aufgerufen und das Motiv im „Capture Image Fenster“ nach Belieben positioniert. Mit „Shoot“ wird vom gewählten Ausschnitt eine BMP- oder JPEG-Datei erzeugt.

Tipp: Markieren Sie zuerst die gewünschte Datei (im Bild „*multi_paint_01.cel*“), bevor Sie „Capture Image“ aktivieren. Nur so wird der Dateiname für das Bild übernommen.



Optimierung der Merge Funktion

Die seit der Version 4.6 existierende Merge-Funktion ist verbessert worden, um beim Rendern von großen CAD Daten nochmals eine bessere Performance zu erzielen.

Beim „Mergen“ wird geprüft wie viele Polygone pro Objekt und wie viele Objekte gleicher Farbe vorliegen. Anhand dieser Information wird bestimmt, ob das Bauteil in einem guten oder schlechten Zustand ist.

Sämtliche Objekte gleicher Farbe werden in einem Objekt zusammengefasst, womit sich die Anzahl der Objekte verringert und die Anzahl der Polygone pro Objekt erheblich steigert. Daraus folgt bessere Performance, die auch beim Kollisions-Check vorteilhaft ist.

Als problematisch erwiesen sich jedoch Bauteile mit extrem vielen Objekten einer einzigen Farbe mit insgesamt extrem vielen Polygonen (>2.000.000). Nach dem Mergen erhielt man so ein einziges Objekt, welches sämtliche Polygone enthielt. Diese lange Polygonkette wurde dann in die Grafikkarte geladen, was zu Problemen und teilweise zur Instabilität führte.

Als Gegenmaßnahme wird nun die maximale Anzahl der Polygone pro Objekt auf 100.000 begrenzt. Somit entstehen im obigen Beispiel mehr als 20 Objekte gleicher Farbe.

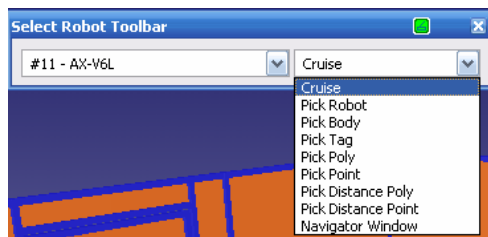
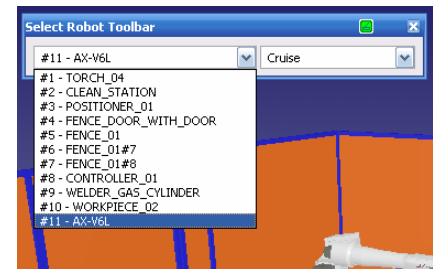
Sonstige Erweiterungen

Toolbar „Select Robot“

Ab der Version 4.606 steht eine weitere Toolbar zur Verfügung, mit der der Bediener bei seiner täglichen Arbeit noch schneller ans Ziel gelangt.

Die „Select Robot“ Toolbar enthält zwei Pull-down-Menüs.

Über das erste Pull-down-Menü kann der Roboter ausgewählt werden, der manipuliert werden soll.

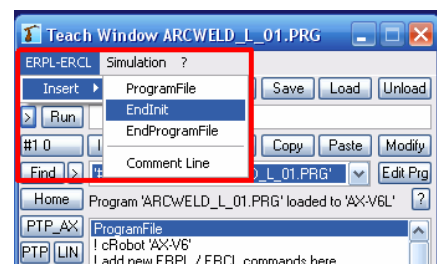
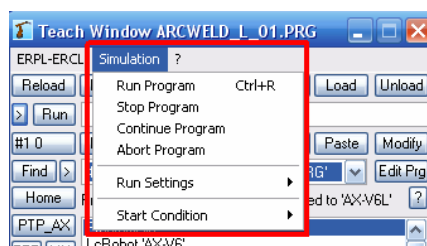


Über das zweite Pull-down-Menü eine Aktion, wie zum Beispiel „Cruise“ oder „Distanzmessung“.

Durch die komfortable Bedienung über die Pull-down-Menüs werden die Anzahl der Mausklicks reduziert und das Ergebnis schneller erreicht.

Menüleiste in Dialogen

Auch die Dialoge sind erweitert worden. Die neuen Menüleisten beinhalten dialogbezogene Funktion, um mit wenigen Mausklicks direkt aus dem Dialog die gewünschte Aktion zu starten.

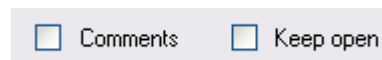


Erweiterung in Teach Window Dialogen

Die drei Teach Window Dialoge „Motion Commands“, „Control Commands“ und „ERCL Commands“ wurden übersichtlicher gestaltet und mit zwei neuen Check-Boxen versehen.

Beim Einfügen von Befehlen hat der Bediener nun die Wahl, ob Kommentare zu den Befehlen mit eingefügt werden oder nicht. Dazu muss die Checkbox „Comments“ aktiviert werden.

Sollen mehrere Befehle aus demselben Kommando Dialog nacheinander eingefügt werden, kann mit der Checkbox „Keep open“ der Dialog offen gehalten werden.



ERC Searcher

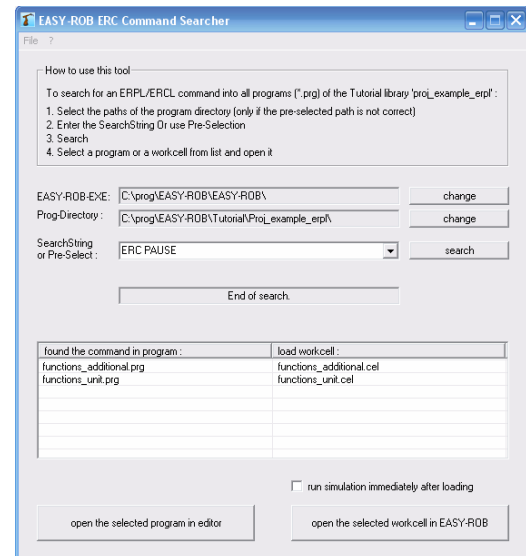
Für jeden ERPL-/ERCL-Befehl sind in der Beispielbibliothek Arbeitszellen mit Programmen vorbereitet.
Der ERC Command Searcher unterstützt den Bediener bei der Suche nach bestimmten ERC Kommandos in der Beispielbibliothek "Proj_example_erpl", die in der Regel im Verzeichnis:
„.\\EASY-ROB\\Tutorial\\Proj_example_erpl“ installiert ist.

Diese Bibliothek enthält viele kurze Beispielprogramme anhand derer der Bediener die Funktionsweise der unterschiedlichen ERC Kommandos leicht nachvollziehen kann.

Nach dem Start des Searchers kann der Bediener das zu suchende Kommando manuell eingeben oder per Pre-selection auswählen und danach suchen lassen:

Weiterhin kann jedes Programm im Editor bzw. jede Workcell im EASY-ROB geöffnet werden, um die Funktionsweise und richtige Anwendung des gesuchten Befehls einzusehen.

Lesen Sie hierzu bitte auch den Abschnitt ERC Command Searcher in den Bedienungshinweisen.

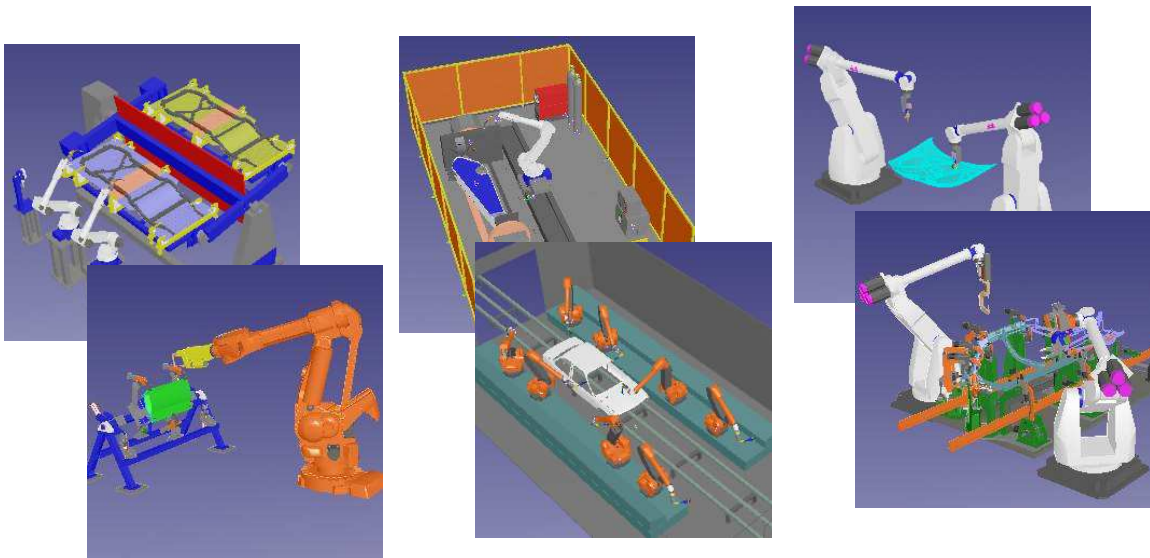


Neue Arbeitszellen-Beispiele

Die seit dem Major Release V4.6 neu verfügbare Funktionalität „**Multi-Program**“ und die damit verbundene Möglichkeit digitale Signale zu verwenden, hat uns veranlasst, einige neue Beispielarbeitszellen für verschiedene Anwendungen bereitzustellen.

Anhand dieser Beispiele und des erweiterten Tutorials soll der Bediener die Vorgehensweise beim Umgang mit digitalen Signalen nachvollziehen und üben können.

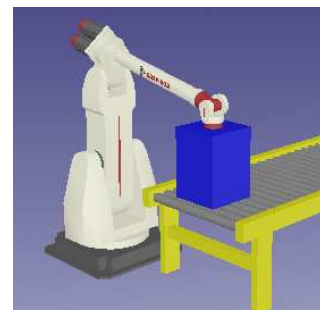
Die neuen Beispiele sind in der ApplicationLib abgelegt.



Erweitertes Tutorial zum Thema Signale

Auch das Tutorial ist entsprechend der neuen Funktion erweitert worden und führt den Bediener Schritt für Schritt durch den Aufbau einer Simulation mit Multi-Program und digitalen Signalen.

Das TrainLib-Tutorial befindet sich bei den Manuals.



ERPL-Befehle

- SPEED_PTP_AX v1 .. vn [m/s, deg/s]
- ACCEL_PTP_AX a1 .. an [m/s², deg/s²]
- SPEED_PTP_OV [%]
- ACCEL_PTP_OV [%]
- SPEED_CP_OV [%]
- ACCEL_CP_OV [%]
- SPEED_ORI_CP_OV [%]
- ACCEL_ORI_CP_OV [%]

Bei PTP Bewegungen wird für jede Achse eine Geschwindigkeit (SPEED_PTP_AX) bzw. Beschleunigung (ACCEL_PTP_AX) programmiert. Somit sind die „alten“ SPEED_PTP- und ACCEL_PTP-Befehle, der für alle Achsen die gleiche Geschwindigkeit/Beschleunigungen programmieren, obsolete. Werden diese alten Befehle dennoch verwendet, wird für jede rotatorische Achse der Wert nach speed_ptp_ax kopiert.

Beispiel: Die 3. Achse des Roboters ist translatorisch, die Achsen 1,2 und 4 rotatorisch.

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

ergibt intern oder entspricht

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

Gleiches gilt für die ACCEL-Befehle

Führende (dominante) Achse bei PTP-Bewegungen:

Fährt der Roboter von A nach B, so fährt die dominierende Achse mit ihrer vorgegebenen speed_ptp_ax Geschwindigkeit, sofern diese Geschwindigkeit erreicht werden kann. Das hängt von der Distanz AB und den Beschleunigungen ab.

SPEED_PTP_OV 20

ergibt für jede Achse 20% der maximalen Achsgeschwindigkeitswerte.

Beispiel: Die maximalen Achsgeschwindigkeiten des Roboters seien 200° 250° 1500mm 400°.

```
speed_ptp_ov 50 // %
```

ergibt intern oder entspricht

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

Mit den neuen OV Befehlen kann recht elegant eine passende Geschwindigkeit/Beschleunigung vorgegeben werden.

API-Funktionen

Siehe header files „er_dvlp.h“ und „er_dvlp_ext.h“

Bewegungsplanung

- **float *inq_ipo_path_vq_axis (void)**
Programmed joint speed
Return: pointer to programmed joint speed
- **float *inq_ipo_path_aq_axis (void)**
Programmed joint acceleration
Return: pointer to programmed joint acceleration
- **float *inq_ipo_path_aq_decel_axis (void)**
Programmed joint deceleration
Return: pointer to programmed joint deceleration
- **float *inq_ipo_path_vq_ov_set (float ov)**
Calculates programmed joint speed vq_axis as percentage override value of the maximum robot joint speed *inq_v_max()
In: ov Percentage override value [1 . . 200]
Return: pointer to programmed joint speed
- **float *inq_ipo_path_aq_ov_set (float ov)**
Calculates programmed joint acceleration aq_axis as percentage override value of the maximum robot joint acceleration *inq_a_max()
In: ov Percentage override value [1 . . 200]
Return: pointer to programmed joint acceleration
- **float *inq_ipo_path_vx_ov_set (float ov)**
Calculates programmed cartesian speed vx as percentage override value of the maximum robot cartesian speed *inq_vx_max()
In: ov Percentage override value [1 . . 200]
Return: pointer to programmed cartesian speed

Neue API-Funktionen

- float *inq_ipo_path_ax_ov_set (float ov)**
 Calculates programmed cartesian acceleration ax as percentage override value of the maximum robot cartesian acceleration *inq_ax_max()
 In: ov Percentage override value [1 . . 200]
 Return: pointer to programmed cartesian acceleration
- float *inq_ipo_path_vx_ori_ov_set (float ov)**
 Calculates programmed cartesian orientation speed vx_ori as percentage override value of the maximum robot cartesian orientation speed *inq_vx_ori_max()
 In: ov Percentage override value [1 . . 200]
 Return: pointer to programmed cartesian orientation speed
- float *inq_ipo_path_ax_ori_ov_set (float ov)**
 Calculates programmed cartesian orientation acceleration ax_ori as percentage override value of the maximum robot cartesian orientation acceleration *inq_ax_ori_max()
 In: ov Percentage override value [1 . . 200]
 Return: pointer to programmed cartesian orientation acceleration

Roboterattribute

- float * inq_vx_max (void)**
 Maximum cartesian TCP speed [m/s]
 Return: pointer to max. cartesian speed
- float * inq_ax_max (void)**
 Maximum cartesian TCP acceleration [m/s²]
 Return: pointer to max. cartesian acceleration
- float * inq_vx_ori_max (void)**
 Maximum cartesian TCP orientation speed [rad/s]
 Return: pointer to max. cartesian orientation speed
- float * inq_ax_ori_max (void)**
 Maximum cartesian TCP orientation acceleration [rad/s²]
 Return: pointer to max. cartesian orientation acceleration

Speichern

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

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

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

Return: 0 - OK, 1 - Error
- int `er_vad_DataSaveCell_ext` (char `*new_cel_flm`=NULL,int `confirm_overwrite`=1, int `reset_pos`=1,int `save_pos`=0)**

Save current loaded cell file as ...

In:	<code>new_cel_flm</code>	Cell file name
	<code>confirm_overwrite</code>	Prompts user to confirm if file exist
	<code>reset_pos</code>	1 - Reset to start condition, 0 - no reset
	<code>save_pos</code>	1 - Save as start condition, 0 - don't save

Return: 0 - OK, 1 - Error
- int `er_vad_DataSaveRobot_ext` (char `*new_rob_flm`=NULL,int `confirm_overwrite`=1)**

Save current selected robot as ...

In:	<code>new_rob_flm</code>	Robot file name
	<code>confirm_overwrite</code>	Prompts user to confirm if file exist

Return: 0 - OK, 1 - Error
- int `er_vad_DataSaveTool_ext` (char `*new_tol_flm`=NULL,int `confirm_overwrite`=1)**

Save current selected robot tool as ...

In:	<code>new_tol_flm</code>	Tool file name
	<code>confirm_overwrite</code>	Prompts user to confirm if file exist

Return: 0 - OK, 1 - Error

Neue API-Funktionen

Mouse Mode

Zwei Funktionen um den „Mouse Mode“ zu setzen, bzw. zu lesen

```
const int MM_CRUISE           = 0
const int MM_TCP_JOG_WZKS     = 1
const int MM_TCP_JOG_WORLD    = 2
const int MM_JNT_JOG          = 3
const int MM_ROBBASE          = 4
const int MM_BODY             = 5
const int MM_BODY_ALL         = 6
const int MM_TAG              = 7
const int MM_BCKGNDCOLOR      = 8
const int MM_FLOORCOLOR       = 9
const int MM_TAG_OFFSET       = 10
const int MM_TOOL             = 11
//const int MM_SELECTION      = 12; // use MousePicking functions
const int MM_LIGHTPOS         = 13
```

- **int er_vad_Set_MouseMode (int mouse_mode)**
Set new desired mouse mode
In: mouse_mode [MM_CRUISE .. MM_LIGHTPOS]
Return: old (previous) mouse mode
- **int er_vad_Get_MouseMode (void)**
Get current mouse mode
Return: current mouse mode [MM_CRUISE .. MM_LIGHTPOS]

Mouse Picking

Zwei weitere "MousePicking" Funktionen, um den Abstand zu messen, siehe Check Boxen des Navigator Windows.

- **int er_vad_PickDistancePolyCenter (int hold_first_point=0)**
Enables the user to measure the distance between two center of polygons.
In: hold_first_point 1st ChkBox in NavigatorWindow
Return: 0-OK, 1-Error

Neue API-Funktionen

- `int er_vad_PickDistancePoint (int hold_first_point=0)`

Enables the user to measure the distance between two vertices.

In: `hold_first_point` 1st ChkBox in NavigatorWindow

Return: 0-OK, 1-Error

Lesen des aktuellen MousePickung Zustandes.

Zu „`int er_vad_Get_cPickMode (void)`“ kommt hinzu.

- `int er_vad_Get_cPickMode_Chk_1 (void)`

Current check_1 box mode in Navigator Window

Return: check box mode

- `int er_vad_Get_cPickMode_Chk_2 (void)`

Current check_2 box mode in Navigator Window

Return: check box mode

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

- `float *er_vad_Get_cPickResults (int result_idx)`

XYZ position or vector component of picked item

In: `result_idx` [PICK_RESULT_DEFAULT ... PICK_RESULT_DISTANCE]

Return: pointer to result vector, xyz component

AuxUpdate

Neue AuxUpdate(int idx) Indices, wenn vom Bediener ein Objekt (Geometrie, Roboter, Tag, etc.) gepickt wurde.

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

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

- **int *er_vad_Get_cPickItem (void)**

Return picked item, case AUX_UPDATE_IDX_SELECTION_HIT in AuxUpdate(.)

Return: Item [MP_HIT_INVALID . . . MP_HIT_CADPREVIEW]

Example:

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

Kontakt

EASY-ROB

3D Robot Simulation Tool

Stefan Anton

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

Tel. +49(0)69 677 24 287

Fax. +49(0)69 677 24 320

Email: info@easy-rob.com, sales@easy-rob.com

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

EASY-ROB Kundenbereich

Online verfügbar: Programm-Updates und Roboterbibliotheken

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

Zugangsdaten:

Benutzer: customer

Passwort: *****

Eigene Notizen