

# KUKA



KUKA System Technology  
KUKA.UserTech 5.0  
适用于 KUKA System Software 8.7



发布日期: 04.05.2023  
KUKA.UserTech 5.0 V1  
KUKA Deutschland GmbH

©版权说明 2023

KUKA Deutschland GmbH  
Zugspitzstraße 140  
D-86165 Augsburg  
德国

此文献或节选只有在征得 KUKA Deutschland GmbH 明确同意的情况下才允许复制或对第三方开放。除了本文献中说明的功能外，控制系统还可能具有其他功能。但是在新供货或进行维修时，无权要求 KUKA Deutschland GmbH 提供这些功能。

我们已就印刷品的内容与描述的硬件和软件内容是否一致进行了校对。但是不排除有不一致的情况，我们对此不承担责任。但是我们定期校对印刷品的内容，并在之后的版本中作必要的更改。

我们保留在不影响功能的情况下进行技术更改的权利。

KIM-PS5-DOC

原版文件的译文

出版物: Pub KUKA.UserTech 5.0 (PDF) zh  
PB21346

书页构造: KUKA.UserTech 5.0 V1.1  
BS18354

版本: KUKA.UserTech 5.0 V1

# 目录

<b>1</b>	<b>引言.....</b>	<b>5</b>
1.1	目标群体.....	5
1.2	工业机器人文档.....	5
1.3	安全提示和注意事项的图示说明.....	5
1.4	所用术语.....	6
1.5	许可证.....	6
<b>2</b>	<b>产品说明.....</b>	<b>7</b>
2.1	产品说明.....	7
2.2	预期用途和违规使用.....	7
<b>3</b>	<b>安全.....</b>	<b>9</b>
<b>4</b>	<b>安装.....</b>	<b>11</b>
4.1	系统前提条件.....	11
4.2	安装 KUKA.UserTech.....	11
<b>5</b>	<b>配置.....</b>	<b>13</b>
5.1	配置：概览.....	13
5.2	使用 KUKA.UserTech 创建工艺.....	14
5.2.1	将 KUKA.UserTech 添加到 KSS 项目中.....	14
5.2.2	打开 User-Tech5 编辑器.....	15
5.2.3	将新应用添加到配置中.....	16
5.2.4	打开应用的编辑视图.....	17
5.2.5	给应用添加和配置新参数.....	18
5.2.5.1	参数：概览.....	18
5.2.5.2	参数： Static, Free.....	19
5.2.5.3	参数： Name.....	19
5.2.5.4	参数： Integer, Real.....	20
5.2.5.5	参数： List.....	20
5.2.5.6	参数： Boolean.....	21
5.2.5.7	参数： Parameter list.....	22
5.2.5.8	参数： Switch.....	25
5.2.6	给应用添加和配置新脚本.....	27
5.2.6.1	操作：概览.....	28
5.2.6.2	操作： Do.....	29
5.2.6.3	操作：对话框.....	29
5.2.6.4	操作： Message.....	30
5.2.6.5	操作： SetVar.....	30
5.2.6.6	操作： ShowVar.....	31
5.2.6.7	操作： Redekl.....	32
5.2.6.8	操作： Switch.....	33
5.2.6.9	操作： SetKey.....	34
5.2.6.10	操作： SetKeyBar.....	36
5.2.7	给应用添加和配置新的行指令.....	37
5.2.8	给应用添加和配置新的状态键.....	38
5.2.9	给应用添加和配置新的状态键栏.....	40

5.2.10	管理图标.....	41
5.2.10.1	创建图标图形的规则.....	42
5.2.11	生成菜单配置文件.....	44
5.3	KXR 语言文件.....	45
5.4	载入配置.....	48
5.5	从配置中导入应用.....	48
5.6	从配置中导入单个元素.....	49
5.7	导出 XML 工艺文件.....	50
5.8	通过 KUKA.WorkVisual 项目将工艺传输至控制器.....	51
5.9	使用 KUKA.OptionPackageEditor 创建备选软件包.....	51
5.9.1	关于 KUKA.OptionPackageEditor 的提示.....	51
5.9.2	创建备选软件包.....	51
5.9.3	定义对 KSS 的依赖性.....	52
5.9.4	定义对 KUKA.UserTech 的依赖性.....	53
5.9.5	添加 XML 工艺文件.....	54
5.9.6	添加图像文件.....	54
5.9.7	添加菜单配置文件.....	55
5.9.8	添加 KXR 语言文件.....	55
5.9.9	添加 KRL 文件.....	56
<b>6</b>	<b>KUKA 客户服务.....</b>	<b>59</b>
6.1	技术支持咨询.....	59
6.2	KUKA 客户支持系统.....	59
	<b>索引</b>	<b>61</b>

# 1 引言

## 1.1 目标群体

本文档面向具备下列知识的用户：

- KRL 编程方面的专业知识
- KUKA.WorkVisual 方面的专业知识
- 机器人控制器的系统知识
- XML 的基础知识



为确保最佳使用 KUKA 产品，我们建议您参加 KUKA 学院的培训课程。有关培训计划的信息请访问公司主页 [www.kuka.com](http://www.kuka.com) 或直接在公司分支机构处获取。

## 1.2 工业机器人文档

工业机器人文档由以下几部分组成：

- 机器人本体文档
- 机器人控制器文档
- smartPAD-2 或 smartPAD pro 的文档（如果使用）
- 系统软件操作和编程指南
- 选件及附件说明
- KUKA Xpert 中的备件概览

每份文档均为独立文件。

## 1.3 安全提示和注意事项的图示说明

### 安全

这些安全提示事关安全，**必须遵守**。



#### 危险

这类安全提示表示：如果未采取预防措施，则肯定或极有可能**会导致死亡或严重伤害**。



#### 警告

这类安全提示表示：如果未采取预防措施，**可能会导致死亡或严重伤害**。



#### 小心

这类安全提示表示：如果未采取预防措施，**可能会导致轻伤**。

#### 注意

这类安全提示表示：如果未采取预防措施，**可能会导致财产损失**。



这类安全提示提及安全相关信息或一般安全措施。  
这类安全提示并不涉及具体危险或具体安全措施。

本安全提示提醒注意用于预防或补救紧急情况或故障的操作步骤：

#### 安全须知

**必须严格遵守以下操作步骤！**

必须严格遵守带有此安全提示标识的操作步骤。

提示

这类提示可使您工作更加便利或提供进一步信息的说明。



有助于您轻松工作或者包含进一步信息的提示。

1.4 所用术语

术语	说明
KFDx	KUKA Form Descriptionx 使用 KUKA.UserTech 4.0 配置的工艺的 XML 文件格式
KRL	KUKA Robot Language KUKA 机器人编程语言
smarthMI	smart Human-Machine Interface (smart 人机界面) smartPAD 上的操作界面
smartPAD	机器人控制系统的手持编程器 smartPAD 具备机械手操作和编程所需的各种操作和显示功能。
String	固定或可变长度的字母数字字符串

1.5 许可证

KUKA 许可证条件和所用开源软件的许可证条件可在下列文件夹中找到：

- 在带 KUKA 软件安装文件的数据载体上的 .\LICENSE 下
- 在 KUKA smarthMI 主菜单的 **帮助** > **信息**，选项卡 **许可证** 下
- 在机器人控制器上的 D:\KUKA\_OPT\备选软件包名\LICENSE 下
- 在 WorkVisual 的**选项**中名称与选项包名称相同的许可证文件夹下



关于开源许可证的更多信息可通过下面的邮件地址获取：  
opensource@kuka.com

## 2 产品说明

### 2.1 产品说明

#### 说明

KUKA.UserTech 5.0 是用于为 KUKA System Software 配置工艺的备选软件包。可以完全使用 KUKA.WorkVisual 中的编辑器来创建配置。

#### 功能

- 行指令和参数列表的配置
- 状态键和状态键栏的配置
- 消息和按钮的配置
- 生成菜单配置文件，以调用 KUKA smarHMI 上的行指令和状态键栏

### 2.2 预期用途和违规使用

#### 应用

备选软件包 KUKA.UserTech 5.0 用于为 KUKA System Software 配置工艺。该软件仅允许用于此用途，且仅能在满足特定系统要求的情况下运行。

#### 违规使用

禁止以任何有别于预期用途的方式使用，否则即被视为违规使用。违规使用会导致丧失保修和责任索赔权。KUKA 对因此类违规使用造成的任何损失概不负责。





### 3 安全

有关工业机器人的安全信息请参阅系统集成商操作和编程指南或最终用户操作及编程指南中的“安全”一章。

**注意与安全相关的信息**

安全使用本产品需要掌握知识和遵守基本安全措施。否则有可能造成死亡、严重身体伤害或财产损失。

- 必须注意遵守 KUKA 系统软件 (KSS) 操作及编程指南中的“安全”一章。



## 4 安装

### 4.1 系统前提条件

#### 硬件

- KR C5 机器人控制系统
- 笔记本电脑 / PC

#### 软件

##### 机器人控制器:

- KUKA System Software 8.7

##### 笔记本电脑/台式电脑:

- KUKA.WorkVisual 6.0

### 4.2 安装 KUKA.UserTech

#### 操作步骤

- 在 KUKA.WorkVisual 中安装备选软件包 KUKA.UserTech。



关于 WorkVisual 中流程步骤的信息见 KUKA.WorkVisual 的文档。



## 5 配置

### 5.1 配置：概览

#### A: 使用 KUKA.UserTech 创建工艺

这些步骤在 KUKA.WorkVisual 中执行。

步骤	说明
1	在 WorkVisual 中安装 KUKA.UserTech (>>> 4.2 “安装 KUKA.UserTech” 页面 11)
2	将 KUKA.UserTech 添加到 KSS 项目中 (>>> 5.2.1 “将 KUKA.UserTech 添加到 KSS 项目中” 页面 14)
3	在 User Tech5 编辑器中打开配置 (>>> 5.2.2 “打开 User-Tech5 编辑器” 页面 15)
4	在 User-Tech5 编辑器中： <ul style="list-style-type: none"><li>创建应用 (&gt;&gt;&gt; 5.2.3 “将新应用添加到配置中” 页面 16)</li><li>打开应用进行编辑 (&gt;&gt;&gt; 5.2.4 “打开应用的编辑视图” 页面 17)</li><li>配置行指令参数 (&gt;&gt;&gt; 5.2.5 “给应用添加和配置新参数” 页面 18)</li><li>为行指令和状态键配置脚本 (&gt;&gt;&gt; 5.2.6 “给应用添加和配置新脚本” 页面 27)</li><li>配置行指令 (&gt;&gt;&gt; 5.2.7 “给应用添加和配置新的行指令” 页面 37)</li><li>配置状态键 (&gt;&gt;&gt; 5.2.8 “给应用添加和配置新的状态键” 页面 38)</li><li>配置状态键栏 (&gt;&gt;&gt; 5.2.9 “给应用添加和配置新的状态键栏” 页面 40)</li><li>导入状态键图标 (&gt;&gt;&gt; 5.2.10 “管理图标” 页面 41)</li><li>生成菜单配置文件 (&gt;&gt;&gt; 5.2.11 “生成菜单配置文件” 页面 44)</li></ul>

#### B: 创建 KXR 语言文件

UI 文本和消息文本可以翻译出来。翻译必须保存在一个（或多个）KXR 语言文件中。

如果将工艺转换成一个备选软件包，则可将 KXR 语言文件导入软件包。之后在安装备选软件包时，这些文件会被传输到机器人控制器。

#### C: 创建附加的 KRL 文件

行指令可以运行自己的 KRL 代码，也可以调用 KRL 函数。必须定义 KRL 函数，必要时还需要创建相关的 KRL 文件。KRL 文件可以通过 KUKA.WorkVisual 项目传输到机器人控制器。

如果将工艺转换成一个备选软件包，则可将 KRL 文件导入软件包。之后在安装备选软件包时，这些文件会被传输到机器人控制器。

**D: 将工艺传输到控制器**

或者：通过 KUKA.WorkVisual 项目将工艺传输到机器人控制器

该方式适用于在机器人控制器上测试新的行指令和/或状态键。

(>>> 5.8 “通过 KUKA.WorkVisual 项目将工艺传输至控制器” 页面 51)

或者：使用 KUKA.OptionPackageEditor 创建备选软件包并添加工艺

之后，可以将备选软件包安装在机器人控制器上。

这是所设想的方式。作为备选软件包，该工艺可以很容易地应用于多个控制器。如果需要，还可以将附加的 KRL 文件和 KXR 语言文件导入备选软件包。

步骤	说明
在 KUKA.WorkVisual 中：	
1	导出 XML 工艺文件 (>>> 5.7 “导出 XML 工艺文件” 页面 50)
在 KUKA.OptionPackageEditor 中：	
2	创建备选软件包 (>>> 5.9.2 “创建备选软件包” 页面 51) <b>注意：</b> 详细信息请见 KUKA.OptionPackageEditor 文档。
3	定义软件包对系统软件的依赖性 (>>> 5.9.3 “定义对 KSS 的依赖性” 页面 52)
4	定义软件包对 KUKA.UserTech 的依赖性 (>>> 5.9.4 “定义对 KUKA.UserTech 的依赖性” 页面 53)
5	添加 XML 工艺文件 (>>> 5.9.5 “添加 XML 工艺文件” 页面 54)
6	如果使用状态键图标：添加图像文件 (>>> 5.9.6 “添加图像文件” 页面 54)
7	添加菜单配置文件 (>>> 5.9.7 “添加菜单配置文件” 页面 55)
8	如果使用翻译：添加 KXR 语言文件 (>>> 5.9.8 “添加 KXR 语言文件” 页面 55)
9	如果使用 KRL 文件：添加文件 (>>> 5.9.9 “添加 KRL 文件” 页面 56)
10	必要时：定义文件更改 <b>注意：</b> 详细信息请见 KUKA.OptionPackageEditor 文档。

**5.2 使用 KUKA.UserTech 创建工艺****5.2.1 将 KUKA.UserTech 添加到 KSS 项目中****说明**

为了能够创建一项工艺，必须在 KUKA.WorkVisual 中将 KUKA.UserTech 添加到 KSS 项目中。

- 从功能的角度来看，无论是将 KUKA.UserTech 添加到来自实际控制器的项目中还是添加到专门在 WorkVisual 中创建的项目（与实际控制器无关）中，都没有区别。

- 从实际控制器加载项目的优势在于，项目可以传输回控制器并且可以在那里测试工艺。
- 如果项目是在 WorkVisual 中专门为创建工艺而创建的，则必须遵守某些设置。

前提条件

- 已在 WorkVisual 中安装 KUKA.UserTech 选项。

操作步骤

从控制器载入项目：

1. 从控制器载入项目。
2. 添加 “UserTech” 选项。  
按照以下步骤操作，但不要设置**固件版本**：这由控制器进行设置。

创建自己的项目：

1. 在按键栏中点击 **新建...**。  
**WorkVisual 项目浏览器** 打开。左侧选出选项卡**创建项目**。
2. 在 **可用的模板** 区域中选定以下模板：
  - **KR C5 项目**
3. 在 **文件名** 栏中输入项目名称，例如 “Technology configuration”。
4. 在 **存储位置** 栏中指定项目的默认目录。必要时选择另一个目录。
5. 点击**新建**按钮。新项目打开。

添加 “UserTech” 选项：

1. 在 **项目结构** 窗口中选择 **设备** 选项卡。
2. 用右键点击节点 **选项** 并选择**添加...**。
3. 一个窗口打开。选择编目**选项**。
4. 选定 **UserTech** 选项并点击**添加**。
5. 一个窗口打开。输入以下值：
  - **固件版本**：8.7.4点击 **Ok** 确认。
6. 一个新窗口打开。将显示被添加到项目中或更改的文件的概览。  
点击 **Ok**。窗口关闭，项目即被保存。

在 **设备** 选项卡中，**选项** 节点中此时显示 **UserTech** 备选软件包。

5.2.2 打开 User-Tech5 编辑器

说明

使用 User-Tech5 编辑器可以打开工艺配置进行编辑。



图 5-1: User-Tech5 编辑器

- 配置名称可以自由选择。
- 配置可以导出并保存为 XML 文件。  
(>>> [5.7 “导出 XML 工艺文件” 页面 50](#))
- 导出的配置可以重新载入编辑器中，包括“旧” KFDx 文件格式的配置。  
(>>> [5.4 “载入配置” 页面 48](#))
- 一个配置可以由多个应用组成。  
(>>> [5.2.3 “将新应用添加到配置中” 页面 16](#))
- 从导出的配置中也可以将单个应用或应用的单个元素导入到编辑器的配置中。或者也可以从“旧” KFDx 文件格式的配置中导入。  
(>>> [5.5 “从配置中导入应用” 页面 48](#))  
(>>> [5.6 “从配置中导入单个元素” 页面 49](#))

#### 前提条件

- KUKA.UserTech 选项已添加到项目中。

#### 操作步骤

- 在窗口 **项目结构** > 选项卡 **设备** > 下双击 **UserTech** 选项。
- 或者：双击一个依赖于 KUKA.UserTech 选项的任意选项。
- 或者：在按钮栏中点击  (**User-Tech5 配置编辑器**) 按钮。
- 或者：选择菜单序列 **编辑器** > **备选软件包** > **User-Tech5 编辑器**。

### 5.2.3 将新应用添加到配置中

#### 说明

一个配置可以由多个应用组成。每个应用都独立于其他应用。在配置中，每个应用都必须具有唯一的名称和密钥。

应用名称稍后会显示在 smarthMI 的以下菜单中：

- 菜单 **指令** > **应用名称**  
在此菜单下可调用应用的所有行指令。
- 主菜单 **配置** > **状态键** > **应用名称**  
在此菜单下可调用应用的所有状态键栏。

应用名称可以翻译出来。为此需要提供 KXR 语言文件。

(>>> [5.3 “KXR 语言文件” 页面 45](#))

可通过多种方式将新应用添加到配置中：


- 创建新应用
- 复制现有应用
- 从配置中导入应用  
(>>> [5.5 “从配置中导入应用” 页面 48](#))

#### 前提条件

- 已在 User-Tech5 编辑器中打开配置。  
(>>> [5.2.2 “打开 User-Tech5 编辑器” 页面 15](#))

#### 操作步骤


##### 创建新应用：

1. 点击  (**创建新应用**) 图标。  
在 **可用的应用** 区域中添加了一个新应用。该应用还是空的并收到了一个自动生成的名称和密钥。



2. 根据需要更改名称和密钥。

复制现有的应用：

- 1. 在 可用的应用 区域中，将光标放置在包含要复制的应用的行中。
- 2. 点击图标 （复制应用）。  
在 可用的应用 区域中添加了一个应用副本。该应用会收到一个自动生成的名称和密钥。
- 3. 根据需要更改名称和密钥。

参数

参数	说明
名称	应用的唯一名称 如要翻译名称，必须按照以下方式保存名称： <i>KXR 模块名称#应用名称</i>
密钥	用于识别/引用应用的唯一密钥

5.2.4 打开应用的编辑视图

说明

在编辑视图左侧显示了一个元素树，其中包含了能够为应用定义或已经定义的所有元素。右侧提供了所选元素的预览。

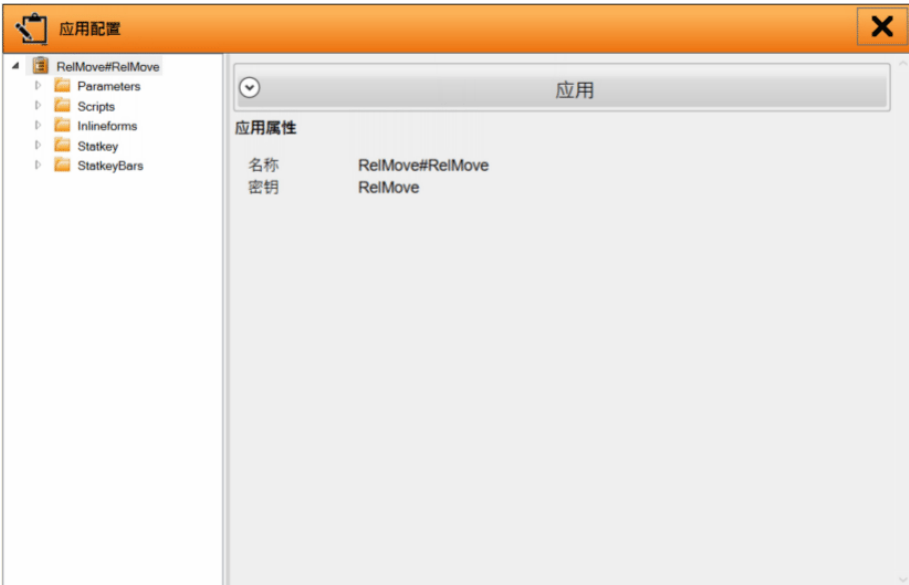


图 5-2：应用编辑视图


- **Parameters**  
包含在应用行指令中使用的所有参数、参数列表和 Switch
- **Scripts**  
包含在应用行指令和状态键中使用的所有脚本
- **Inlineforms**  
包含所有行指令
- **Statkey**  
包含各个状态键
- **StatkeyBars**  
包含完整定义的状态键栏

通过右键单击元素树中的元素可打开上下文菜单，其中包含了可用于该元素的编辑功能。

前提条件

- 已在 User-Tech5 编辑器中打开配置。  
(>>> 5.2.2 “打开 User-Tech5 编辑器” 页面 15)

操作步骤

- 在 可用的应用 区域中，将光标放置在包含要编辑的应用的行中。然后点击  (配置应用) 图标。
- 或者：在 可用的应用 区域中双击应用。

5.2.5 给应用添加和配置新参数

说明

应用可以包含任意数量的参数。在应用中，参数必须具有唯一的名称和密钥。参数名称可以翻译出来。为此需要提供 KXR 语言文件。  
(>>> 5.3 “KXR 语言文件” 页面 45)

前提条件

- 应用的编辑视图已打开。  
(>>> 5.2.4 “打开应用的编辑视图” 页面 17)

操作步骤

- 右键单击元素树中的 **Parameters** 文件夹。将显示一个可用参数列表。
- 选择所需的参数。一个新参数被添加到元素树中。
- 选择新参数。将显示行指令参数预览和参数的默认属性。
- 如要编辑参数属性，点击 **配置** 按钮。一个对话框打开。
- 根据需要配置属性并点击 **Ok** 应用配置。

5.2.5.1 参数：概览

可配置下列参数以在行指令中使用：

参数	在行指令中的作用
Static	显示任何不可编辑的文本 (>>> 5.2.5.2 “参数： Static, Free” 页面 19)
Free	输入任何可编辑的文本 (>>> 5.2.5.2 “参数： Static, Free” 页面 19)
Name	输入名称 (>>> 5.2.5.3 “参数： Name” 页面 19)
Integer	输入 Integer 值 (>>> 5.2.5.4 “参数： Integer, Real” 页面 20)
Real	输入 Real 值 (>>> 5.2.5.4 “参数： Integer, Real” 页面 20)
List	从列表中选择条目 (>>> 5.2.5.5 “参数： List” 页面 20)

参数	在行指令中的作用
Boolean	输入布尔值 (>>> 5.2.5.6 “参数: Boolean” 页面 21)
Parameter list	选择参数列表 (>>> 5.2.5.7 “参数: Parameter list” 页面 22)
Switch	在行指令的不同变型之间切换 (>>> 5.2.5.8 “参数: Switch” 页面 25)

### 5.2.5.2 参数: Static, Free

#### 说明

在行指令中的作用:

- **Static:** 显示任何不可编辑的文本
- **Free:** 输入任何可编辑的文本

#### 属性

属性	说明	是否需要?
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认	用于预定义行指令中参数的值	可选
单位	行指令中显示的参数的单位	可选

### 5.2.5.3 参数: Name

#### 说明

在行指令中的作用:

- 输入变量、函数或子程序的名称

适用 KRL 变量名称的规则。输入内容不允许违背这些规则。

- 允许: 字母 (A-Z、a-z)、数字 (0-9)、特殊字符 “\_” 和 “\$”
- 名称不允许以数字开头
- 最多 24 个字符

#### 属性

属性	说明	是否需要?
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认	用于预定义行指令中参数的值	可选
单位	行指令中显示的参数的单位	可选

#### 5.2.5.4 参数：Integer, Real

##### 说明

在行指令中的作用：

- **Integer**：输入 Integer 值
- **Real**：输入 Real 值

##### 属性

属性	说明	是否需要？
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认	用于预定义行指令中参数的值	可选
最大值	最大可能值	需要
最小值	最小可能值	需要
步幅	按下键盘上的箭头键时输入值的变化步幅 <ul style="list-style-type: none"> <li>• 向上箭头：数值按照[步幅]增加。</li> <li>• 向下箭头：数值按照[步幅]减小。</li> </ul>	可选
单位	行指令中显示的参数的单位	可选

#### 5.2.5.5 参数：List

##### 说明

在行指令中的作用：



- 从列表中选择条目

##### 属性

属性	说明	是否需要？
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认	用于预定义行指令中参数的值（从所有已配置的列表条目中选择）	需要

##### 配置

在该区域中可以配置列表条目。

- 如要创建新的列表条目，点击加号图标 .
- 如要删除列表条目，点击 x 图标 .

属性	说明	是否需要？
显示	行指令中列表条目的显示名称	可选
密钥	在选择行指令中列表条目时写入的 KRL 值 该值可以通过编程方式处理，例如通过将其传递给函数。	可选

##### 示例

选择行指令中的各种数据列表：



图 5-3：列表条目的配置

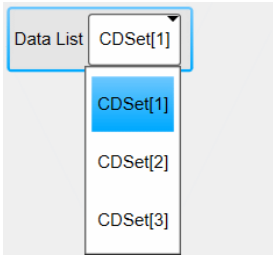


图 5-4：参数预览

5.2.5.6 参数：Boolean

说明

在行指令中的作用：

- 使用复选框输入布尔值

属性

属性	说明	是否需要？
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认	用于预定义行指令中参数的值 <ul style="list-style-type: none"><li>• 勾选 = TRUE</li><li>• 未勾选 = FALSE</li></ul>	可选
TRUE 的 KRL	TRUE 时程序中的 KRL 代码	可选
FALSE 的 KRL	FALSE 时程序中的 KRL 代码	可选

示例



图 5-5: Boolean 示例

5.2.5.7 参数: Parameter list

说明

在行指令中的作用:

- 选择参数列表



在参数列表中，几个参数组合成一组。一个参数列表最多可配置 3 组。这些组显示在 KUKA smartHMI 的不同选项卡中。

参数列表属性

属性	说明	是否需要?
密钥	用于识别/引用参数列表的唯一密钥	需要
默认值	首次打开行指令时用于预定义行指令中参数列表的默认值  当下次打开行指令时，该值会自动递增。示例：默认值“ParList”变为值“ParList1”、“ParList2”等。	需要

组属性

在该区域中可以配置组和组属性（带预览）:

- 如要创建新组，点击加号图标 .
- 如要删除组，点击 x 图标 .
- 所需参数可通过拖放功能插入到相应的组中并在组中重新定位。
- 插入的参数可通过拖放功能移动到回收站中并从组中删除。

属性	说明	是否需要?
名称	要在参数列表中显示的组名称	需要
KRL 类型	要创建的 KRL 结构的数据类型  设置的参数值保存在 KRL 结构中。 <b>注意:</b> 新的 KRL 数据类型必须在 DAT 文件（数据列表）中进行声明。如有必要，创建附加文件或在 KUKA.OptionPackageEditor 中更改文件。	需要

属性	说明	是否需要?
默认	KRL 结构的默认值 用于使用默认值对 KRL 结构进行初始化。	需要
Shortcut	所创建变量的前缀  <b>注意:</b> 在 KRL 程序中总是使用 <b>Shortcut + Value</b> (行指令中参数列表的当前值) 作为变量名。  <b>注意:</b> <b>Shortcut + Value</b> 不能超过 24 个字符。	可选

示例

为用于编程相对运动的参数列表配置了 3 个组:

- 组 1: X、Y、Z 方向运动参数
- 组 2: 围绕 A、B、C 转角的旋转参数
- 组 3: 用于选择参照坐标系的参数: TOOL, BASE

在组 1 中, FRAME 被定义为要创建的 KRL 结构的数据类型。



图 5-6: 参数: 参数列表示例: 组 1

在组 2 中, FRAME 也被定义为要创建的 KRL 结构的数据类型。为了将两组参数保存在一个结构 (变量) 中, 组 1 和组 2 必须使用相同的前缀 (此处为 **Shortcut \_DP**)。



图 5-7：参数：参数列表示例：组 2

在组 3 中，Coorsystem\_T 被定义为要创建的 KRL 结构的数据类型。该数据类型在 KRL 中还未知，必须进行声明：GLOBAL STRUC Coorsystem\_T INT Base, Tool



图 5-8：参数：参数列表示例：组 3

如果在包含参数列表的行指令中设置指令，则会创建一个新变量“DP\_RelPar7”。



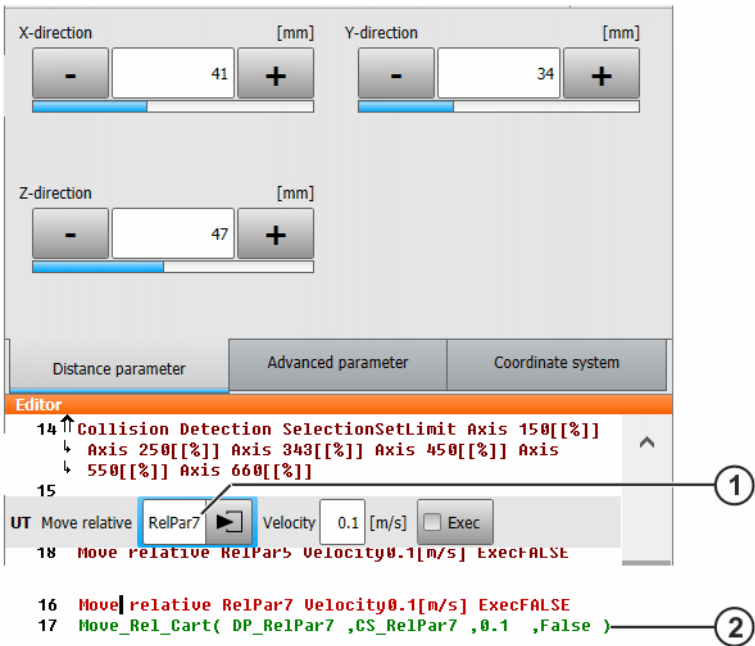


图 5-9：程序示例中的行指令及参数列表

- 1 参数列表的当前值
- 2 保存的变量名（展开折叠）

“DP\_RelPar7”变量的值保存在 KRL 程序的数据列表中。

```
DECL FRAME DP_RelPar7={X 41.0000,Y 34.0000,Z 47.0000,A 10.0000,B 12.0000,C 24.0000}  
DECL CoordSystem_T CS_RelPar7={Base 1,Tool 1}
```

图 5-10：参数列表示例：数据列表中保存的值

5.2.5.8 参数：Switch

说明

在行指令中的作用：

- 在行指令的不同变型之间切换

为每个变型创建一个 Case。之后可以从行指令的列表中选择这些 Case。

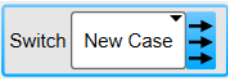



图 5-11：Switch 参数


属性

属性	说明	是否需要？
名称	要在行指令中显示的参数名称	可选
密钥	用于识别/引用参数的唯一密钥	需要
默认值	用于预定义行指令中参数的值（从所有已配置的 Case 中选择）	需要

案例配置；

在该区域中可以创建和配置不同的 Case（变型）（带预览）：

- 如要创建新 Case，点击加号图标 .

- 如要删除 Case，点击 x 图标 。
- 使用拖放功能可以将选择 Case 时要显示的行指令变型参数插入 Case 中并重新定位。
- 插入的参数可通过拖放功能移动到回收站中并从 Case 中删除。
- 插入的参数可通过拖放功能插入 KRL 代码中。作为 KRL 代码，例如可以传输一个函数用来处理程序中行指令变型的参数。
- 在行指令的选择列表中，Case 的显示顺序与此处创建的顺序相同。

使用按钮   可以更改顺序。

属性	说明	是否需要？
名称	要在行指令选择列表中显示的 Case 名称	可选
密钥	用于识别/引用 Case 的唯一密钥	需要
KRL 代码	在使用程序中的行指令变型时要执行的 KRL 代码	可选



如果在 KRL 代码中使用变量、函数等，程序员必须确保它们已被声明。如有必要，创建附加文件或在 KUKA.OptionPackageEditor 中更改文件。

示例

为用于碰撞检测的行指令配置一个包含以下 Case 的 Switch：

- SetLimit: SET\_TQ\_VALUES(int Axis1, int Axis2, int Axis3, int Axis4, int Axis5, int Axis6)
- UseDataSet: USE\_CD\_PARAMS(int DataList)

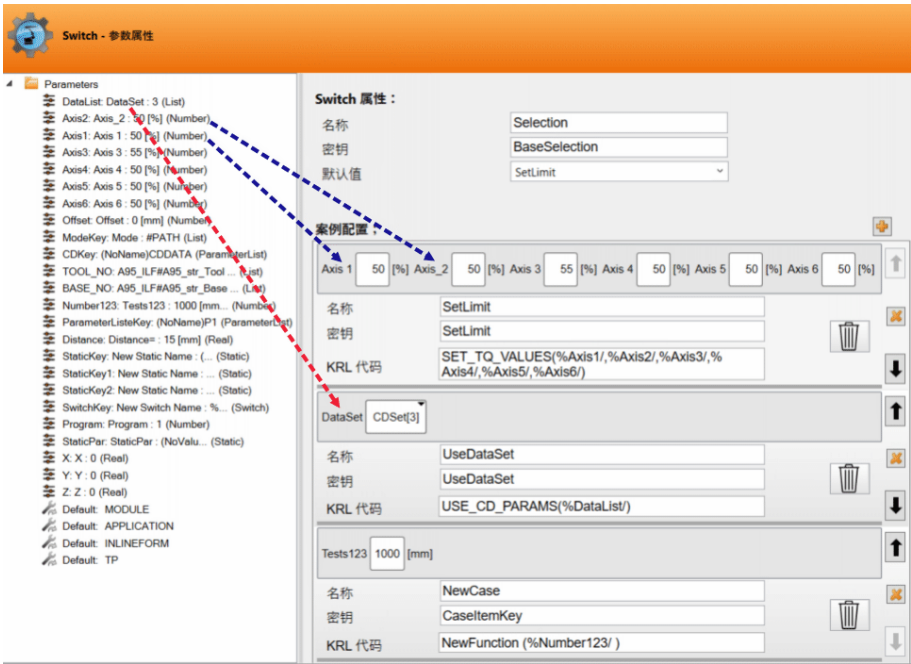


图 5-12: 参数: Switch 示例

行指令“Collision Detection”中的 Switch 参数：

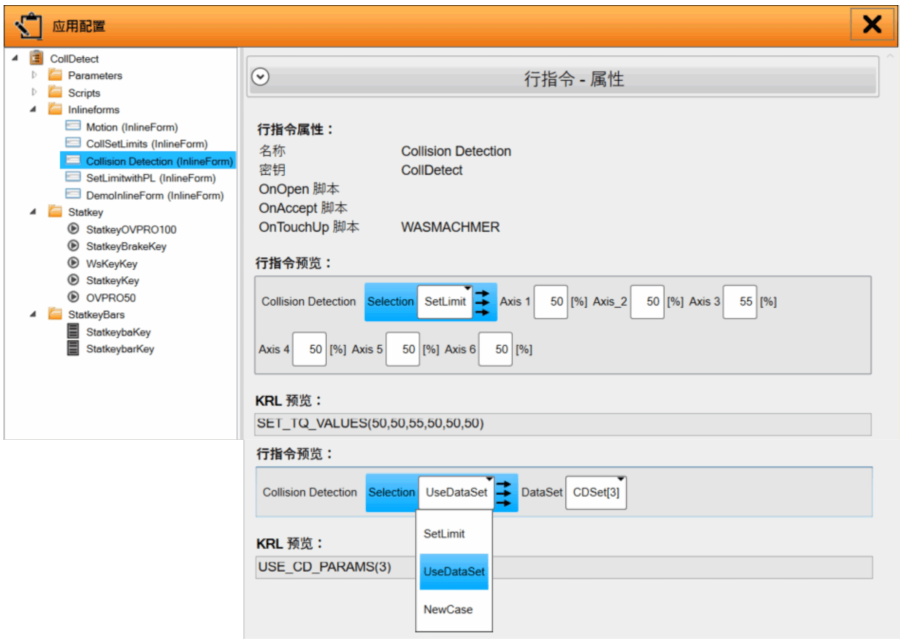


图 5-13: 参数: 行指令中的 Switch 示例

5.2.6 给应用添加和配置新脚本

说明

脚本由一系列依次执行的操作（任意数量）组成。一个应用可以包含任意数量的脚本。在应用中，脚本必须具有唯一密钥。

应用

行指令和状态键可以使用脚本。脚本运行总是与特定的事件相关联。脚本可用于以下行指令事件：

- **OnOpen 脚本**  
当使用 **更改** 按钮或通过 **指令** 菜单打开行指令时，将运行脚本。
- **OnAccept 脚本**  
当使用 **指令 OK** 按钮关闭打开的行指令时，将运行脚本。
- **OnTouchUp 脚本**  
当使用 **Touch-Up** 按钮示教位置并应用于行指令时，将运行脚本。

(>>> [5.2.7 “给应用添加和配置新的行指令” 页面 37](#))

脚本可用于以下状态键事件：

- **OnKeyShow**  
当从主菜单调用状态键时，将运行脚本。
- **OnKeyDown**  
当按下具有 PLUS 功能的状态键时，将运行脚本。
- **OnKeyDownMinus**  
当按下具有 MINUS 功能的状态键时，将运行脚本。
- **OnKeyUp**  
当松开具有 PLUS 功能的状态键时，将运行脚本。
- **OnKeyUpMinus**  
当松开具有 MINUS 功能的状态键时，将运行脚本。

- **OnKeyRepeat**  
当长按具有 PLUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。
- **OnKeyRepeatMinus**  
当长按具有 MINUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。

(>>> [5.2.8 “给应用添加和配置新的状态键” 页面 38](#))

前提条件

- 应用的编辑视图已打开。  
(>>> [5.2.4 “打开应用的编辑视图” 页面 17](#))

操作步骤

1. 右键单击元素树中的 **Scripts** 文件夹，然后选择 **Add Script**。一个新脚本被添加到元素树中。
2. 选择新脚本。将显示带有关联操作列表的预览（仍为空）。
3. 为脚本指定唯一密钥：为此，点击 **配置** 按钮，输入密钥并点击 **Ok** 应用配置。
4. 右键单击 **Scripts** 文件夹中的新脚本。将显示一个可用操作列表。
5. 选择所需的操作。一个新操作被添加到脚本中。
6. 选择新操作。将显示操作预览。
7. 如要配置操作，点击 **配置** 按钮。一个对话框打开。
8. 根据需要配置操作并点击 **Ok** 应用配置。
9. 如要添加更多操作，请返回至步骤 4。

5.2.6.1 操作：概览

可以为脚本配置以下操作：

操作	说明
Do	运行另一个脚本 (>>> <a href="#">5.2.6.2 “操作：Do” 页面 29</a> )
对话框	设置对话框（在 smarthMI 上显示） (>>> <a href="#">5.2.6.3 “操作：对话框” 页面 29</a> )
Message	设置消息（在 smarthMI 的消息窗口中显示） (>>> <a href="#">5.2.6.4 “操作：Message” 页面 30</a> )
SetVar	将 KRL 变量设置为一个值 (>>> <a href="#">5.2.6.5 “操作：SetVar” 页面 30</a> )
ShowVar	读取 KRL 变量的值并将该值设置为参数 (>>> <a href="#">5.2.6.6 “操作：ShowVar” 页面 31</a> )
Redecl	声明新的 KRL 变量或变量结构或覆盖现有的 KRL 变量或变量结构 (>>> <a href="#">5.2.6.7 “操作：Redecl” 页面 32</a> )
Switch	根据变量、参数或表达式的值运行不同的脚本（Switch-Case 指令） (>>> <a href="#">5.2.6.8 “操作：Switch” 页面 33</a> )
SetKey	设置状态键 (>>> <a href="#">5.2.6.9 “操作：SetKey” 页面 34</a> )

操作	说明
SetKeyBar	设置状态键栏 (>>> 5.2.6.10 “操作: SetKeyBar” 页面 36)

### 5.2.6.2 操作: Do

#### 说明

通过此操作可以运行另一个脚本。所有已为应用定义的脚本以及预定义的脚本均可供选择。

#### 预定义的脚本

预定义了以下脚本:

- **AcceptInlineForm**  
只能用作行指令中的 OnAccept 脚本: 当使用 **指令 OK** 按钮关闭行指令时, 将运行 OnAccept 脚本。  
(>>> 5.2.7 “给应用添加和配置新的行指令” 页面 37)  
使用 AcceptInlineForm 可以在关闭行指令时应用更改的参数。
- **CancelInlineForm**  
只能用作行指令中的 OnAccept 脚本: 当使用 **指令 OK** 按钮关闭行指令时, 将运行 OnAccept 脚本。  
(>>> 5.2.7 “给应用添加和配置新的行指令” 页面 37)  
使用 CancelInlineForm 可以在关闭行指令时不应用更改的参数。
- **End**  
正在运行的脚本将会中断。
- **Nothing**  
脚本的占位符

### 5.2.6.3 操作: 对话框

#### 说明

通过此操作可以在 smartHMI 上显示带有消息文本的对话框。为响应对话框中消息文本, 最多可以配置 7 个调用定义脚本的按钮。

消息文本和按钮标注可以翻译出来。为此需要提供 KXR 语言文件。

(>>> 5.3 “KXR 语言文件” 页面 45)



#### 文本

在此处输入对话框的消息文本。

- 消息文本中的参数引用将替换为当前值。
- 消息参数必须用竖线符号 “|” 分隔。

#### 选项 (按钮)

在此处配置按钮 (带有按钮标注预览):

- 如要创建新按钮, 点击加号图标 .
- 如要删除按钮, 点击 x 图标 .
- 可以为每个按钮输入标注文本并选择要运行的脚本。
- 标注文本中的参数引用将替换为当前值。

5.2.6.4 操作：Message

说明

通过此操作可以在 smartHMI 的消息窗口中显示消息。  
消息文本可以翻译出来。为此需要提供 KXR 语言文件。  
(>>> 5.3 “KXR 语言文件” 页面 45)

信息

- 在此处输入消息文本。
- 消息文本中的参数引用将替换为当前值。
  - 消息参数必须用竖线符号 “|” 分隔。

5.2.6.5 操作：SetVar

说明

通过此操作可以为 KRL 变量赋予新值。所赋的值可以是常量、KRL 变量、参数或算术表达式。

- 在表达式中不允许将 “/” 用作运算符。

参数

参数	说明
带路径的变量	要设定的变量名称 字符串中的参数引用将替换为当前值。该字符串将作为变量名称写入路径说明中。
数值	变量要被设置的值 字符串中的参数引用将替换为当前值。该字符串将作为目标值解读。

示例

对 Char 数组的赋值必须用引号 “...” 设置：



图 5-14：操作：带有 Char 数组的 SetVar 示例

参数值的分配：

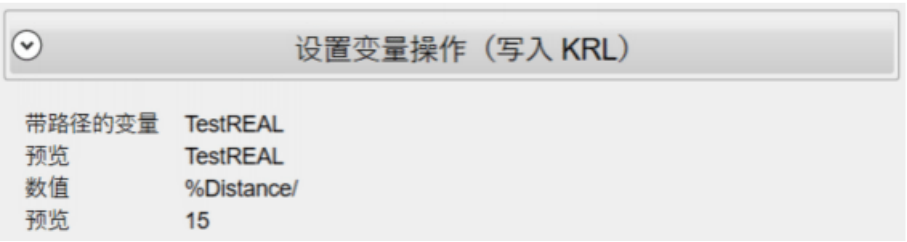


图 5-15：操作：带有参数值分配的 SetVar 示例

参数值在不同表达式中的使用：



图 5-16：操作：表达式中参数值的 SetVar 示例

5.2.6.6 操作： ShowVar

说明

通过此操作可以读取 KRL 变量的值并将应用参数设置为该值。变量也可以是常量或算术表达式。

参数

参数	说明
带路径的变量	要读取的变量名称 字符串中的参数引用将替换为当前值。该字符串将作为变量名称写入路径说明中。
参数	要写入读取值的参数 可选择已为应用创建的参数密钥。

字符串常量的使用

通过添加前缀 “\$Str:” 将路径声明为常量字符串：



图 5-17：操作：带有字符串常量的 ShowVar 示例

在这种情况下解析对参数的引用并将参数 “StaticPar” 设置为值 “a1.DAT”。

表达式中除法的使用

通过添加前缀 “=” 将路径声明为算术表达式：



图 5-18：操作：带有表达式中除法的 ShowVar 示例

在这种情况下，“/”不是变量路径的分隔符，而是具有算术运算符的含义。如果后面跟着控制符，则还需要在“/”后面加一个空格。在这种情况下，“%”用作参数引用的标记。

5.2.6.7 操作：Redecl

说明

通过此操作可以声明新的 KRL 变量或变量结构，或者覆盖现有的 KRL 变量或变量结构。

参数

参数	说明
带路径的变量	变量的完整路径（无变量名） 字符串中的参数引用将替换为当前值。该字符串将作为变量名称写入路径说明中。
声明	变量的声明 字符串中的参数引用将替换为当前值。该字符串将作为目标值解读。
KRL 预览：	显示变量的当前值（声明部分） 字符串中的参数引用将替换为当前值。
全局	变量的适用范围 <ul style="list-style-type: none"><li>勾选 = TRUE 变量全局有效。将全局变量写入文件 Global_Points.Dat 中。</li><li>未勾选 = FALSE 变量局部有效，即仅在相关应用中有效。</li></ul>

示例

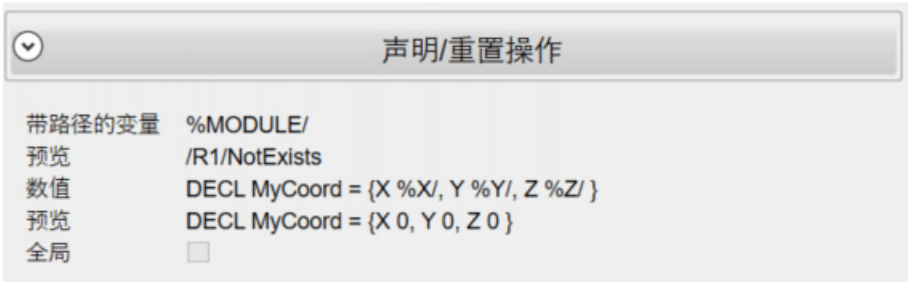


图 5-19：操作：Redecl 示例



5.2.6.8 操作：Switch

说明

通过此操作可以根据条件运行不同的脚本（Switch-Case 指令）。所有已为应用定义的脚本以及预定义的脚本均可供选择。

预定义的脚本

预定义了以下脚本：

- **AcceptInlineForm**  
只能用作行指令中的 OnAccept 脚本：当使用 **指令 OK** 按钮关闭行指令时，将运行 OnAccept 脚本。  
(>>> [5.2.7 “给应用添加和配置新的行指令” 页面 37](#))  
使用 AcceptInlineForm 可以在关闭行指令时应用更改的参数。
- **CancelInlineForm**  
只能用作行指令中的 OnAccept 脚本：当使用 **指令 OK** 按钮关闭行指令时，将运行 OnAccept 脚本。  
(>>> [5.2.7 “给应用添加和配置新的行指令” 页面 37](#))  
使用 CancelInlineForm 可以在关闭行指令时不应用更改的参数。
- **End**  
正在运行的脚本将会中断。
- **Nothing**  
脚本的占位符

条件



在执行操作时评估参数 **条件**。可以指定 KRL 变量、参数引用或表达式。对于局部 KRL 变量，必须指定路径。

此外，还可以输入以下预定义的状态作为条件：

- **IsChange**
  - = TRUE，如果使用 **更改** 按钮打开了行指令。
  - = FALSE，如果通过 **指令** 菜单打开了行指令。
- **DoAlways**  
DoAlways 仅对 OnOpen 脚本有意义，因为此脚本仅在打开或更改行指令时运行。

案例

在该区域中可以创建和配置各种 Case。

- 如要创建新 Case，点击加号图标 。
- 如要删除 Case，点击 x 图标 。

参数	说明
案例条件	Case 出现条件必须具备的值
待执行的脚本	Case 出现时要运行的脚本

Else

在该区域中可以选择在未出现所配置的 Case 时要运行的脚本。

示例

如果参数 “Pic\_Key” 的值为 “2”，则运行脚本 “ScriptSet1”，否则运行脚本 “ScriptSet2”。

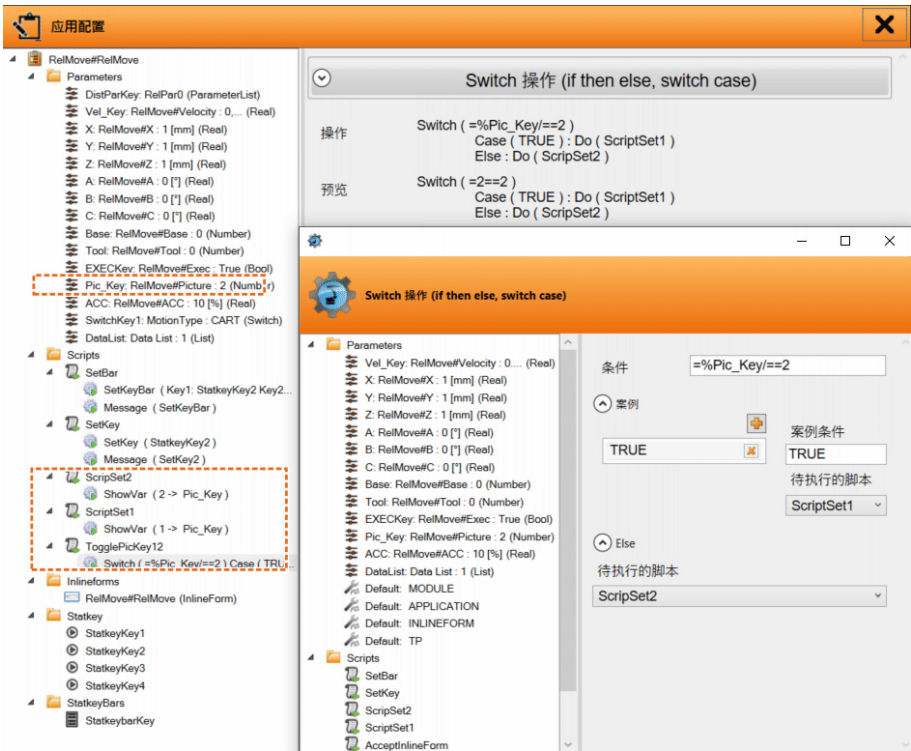


图 5-20： 操作： Switch 示例

5.2.6.9 操作： SetKey

说明

通过此操作可以更改单个状态键的当前属性。未在操作中设置的属性值将保留其当前值。

基础参数

参数	说明
密钥	要更改的状态键的密钥

显示文本参数

在该区域中可以配置状态键格式和状态键的显示文本。  
显示文本可以翻译出来。为此需要提供 KXR 语言文件。

(>>> 5.3 “KXR 语言文件” 页面 45)

参数	说明
上部	smarthMI 侧框上状态键旁边显示的文本（上部）
中间	smarthMI 侧框上状态键旁边显示的文本（中间）
下部	smarthMI 侧框上状态键旁边显示的文本（下部）

参数	说明
Style	<p>状态键格式</p> <ul style="list-style-type: none"> <li>SWITCH 单功能状态键 单功能状态键对应一个键。该键始终具有 PLUS 功能。</li> <li>TOGGLE 双功能状态键 双功能状态键对应 2 个键。上部按键具有 PLUS 功能。下部按键具有 MINUS 功能。</li> </ul>

### 前提条件参数

在该区域中可以定义状态键的激活条件。

参数	说明
安全开关	<p>状态键是否激活取决于使能开关的状态。</p> <ul style="list-style-type: none"> <li><b>True:</b> 按下使能开关后，状态键激活。</li> <li><b>False:</b> 无需使能来激活</li> </ul>
驱动装置 OK	<p>状态键是否激活取决于驱动装置的状态。</p> <ul style="list-style-type: none"> <li><b>True:</b> 在驱动装置接通时，状态键激活。</li> <li><b>False:</b> 无需接通驱动装置来激活</li> </ul>
提交状态	<p>状态键是否激活取决于提交解释器的状态和/或机器人解释器的状态。</p>
机器人解释器状态	<ul style="list-style-type: none"> <li><b>Free</b> 在解释器中未选择任何程序。</li> <li><b>Stop</b> 在解释器中已选择一个程序。程序已停止运行。</li> <li><b>重置</b> 在解释器中已选择一个程序。程序已停止运行并重置。</li> <li><b>End</b> 在解释器中已选择一个程序，并且程序已经结束。</li> <li><b>Active</b> 在解释器中已选择一个程序，并且程序正在运行。</li> </ul> <p>可以选择多个状态。如果未选择任何解释器状态，则状态键默认在以下状态下激活：</p> <ul style="list-style-type: none"> <li><b>提交状态 = Active</b></li> <li><b>机器人解释器状态 = Free Stop 重置 End</b></li> </ul>
ModeOp	<p>状态键是否处于激活状态下取决于运行模式。</p> <ul style="list-style-type: none"> <li><b>T1、T2、Aut、Ext</b></li> </ul> <p>可以选择多种运行模式。如果未选择运行模式，则状态键默认在运行模式 T1、T2 或 AUT 下激活。</p>

事件编辑



在该区域中可以将脚本分配给不同的状态键事件。从下拉列表中可以选为应用定义的所有脚本。

如要取消分配，请选择下拉列表中的第一个空行。

参数	说明
OnKeyShow	当从主菜单调用状态键时，将运行脚本。
OnKeyDown	当按下具有 PLUS 功能的状态键时，将运行脚本。
OnKeyDownMinus	当按下具有 MINUS 功能的状态键时，将运行脚本。
OnKeyUp	当松开具有 PLUS 功能的状态键时，将运行脚本。
OnKeyUpMinus	当松开具有 MINUS 功能的状态键时，将运行脚本。
OnKeyRepeat	当长按具有 PLUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。
OnKeyRepeatMinus	当长按具有 MINUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。

可用的图标

在该区域中可以为状态键的各种状态分配图标。所有可用图标均列出：

- 如要分配图标，选定所需图标并点击  按钮。
- 如要取消分配，点击  按钮。

参数	说明
默认	默认显示在 smartHMI 侧框上状态键旁边的图标
DownKey	按下具有 PLUS 功能的按键时 smartHMI 侧框上状态键旁边显示的图标
DownKeyMinus	按下具有 MINUS 功能的按键时 smartHMI 侧框上状态键旁边显示的图标

5.2.6.10 操作： SetKeyBar

说明



通过此操作可以更改现有的状态键栏，即重新分配状态键。

KeyBar 属性：

参数	说明
密钥	要更改的状态键栏的密钥

可用的状态键

在该区域中可以为状态键栏分配各个状态键及其在状态键中的位置。所有可用的状态键均列出：

- 如要分配状态键，选定所需的图标并点击  按钮。
- 如要取消分配，点击  按钮。

双功能状态键（TOGGLE） 占用 2 个位置。

5.2.7 给应用添加和配置新的行指令

说明

一个应用可以由多个行指令组成。使用行指令可以设置各种参数、编写 KRL 代码行和运行各种脚本。

前提条件

- 应用的编辑视图已打开。  
(>>> 5.2.4 “打开应用的编辑视图” 页面 17)

操作步骤

1. 右键单击元素树中的 **Inlineforms** 文件夹，然后选择 **Add InlineForm**。一个新的行指令被添加到元素树中。
2. 选择新的行指令。将显示行指令预览（仍为空）、相关的 KRL 代码和行指令的默认属性。
3. 如要配置行指令，点击 **配置** 按钮。一个对话框打开。将显示已为应用创建的所有参数和脚本。
4. 根据需要配置行指令并点击 **Ok** 应用配置。

行指令属性：

属性	说明	是否需要？
名称	要在行指令中显示的行指令名称	可选
密钥	用于识别/引用行指令的唯一密钥	需要
OnOpen 脚本	当使用 <b>更改</b> 按钮或通过 <b>指令</b> 菜单打开行指令时，将运行脚本。	可选
OnAccept 脚本	当使用 <b>指令 OK</b> 按钮关闭打开的行指令时，将运行脚本。	可选
OnTouchUp 脚本	当使用 <b>Touch-Up</b> 按钮示教位置并应用于行指令时，将运行脚本。	可选



行指令配置：

在该区域中可以配置行指令参数（带预览）：

- 所需参数可通过拖放功能插入行指令中并在行指令中重新定位。
- 插入的参数可通过拖放功能移动到回收站中并从行指令中删除。

KRL 配置：

在该区域中可以配置要插入程序的 KRL 代码。代码可以分布在多行中。

- 如要创建新行，点击加号图标 .
- 如要删除行，点击 x 图标 .
- 使用拖放功能可将行指令中的参数拖动到所需行中。



如果在 KRL 行中使用变量、函数等，程序员必须确保它们已被声明/定义。如有必要，创建附加文件或在 KUKA.OptionPackageEditor 中更改文件。

## 5.2.8 给应用添加和配置新的状态键

### 说明

KUKA smartPAD 左侧的 4 个状态键可以配置不同功能。这些功能是借助在特定状态键事件下运行的脚本进行配置的。

在一个应用中可以配置任意数量的状态键。在分配给状态键栏之前，不会应用这些状态键。

(>>> [5.2.9 “给应用添加和配置新的状态键栏” 页面 40](#))

除功能外，还可以配置状态键的以下属性：

- 格式和外观，例如可以存储状态键的显示文本或图标
- 状态键的激活条件，例如仅在特定运行模式下或按下使能开关时激活

使用脚本和 SetStatkey 操作可以更改各个状态键的属性和功能。

(>>> [5.2.6.9 “操作：SetKey” 页面 34](#))

### 前提条件

- 应用的编辑视图已打开。  
(>>> [5.2.4 “打开应用的编辑视图” 页面 17](#))
- 已配置所需脚本。  
(>>> [5.2.6 “给应用添加和配置新脚本” 页面 27](#))
- 已导入所需图标。  
(>>> [5.2.10 “管理图标” 页面 41](#))

### 操作步骤

1. 右键单击元素树中的 Statkey 文件夹，然后选择 Add Statkey。一个新的状态键被添加到元素树中。
2. 选择新的状态键。将显示状态键预览（仍为空）及其属性和参数。
3. 如要配置状态键，点击 **配置** 按钮。一个对话框打开。
4. 根据需要配置状态键并点击 **Ok** 应用配置。

### 基础参数

参数	说明
密钥	用于识别/引用状态键的唯一密钥

### 显示文本参数

在该区域中可以配置状态键格式和状态键的显示文本。

显示文本可以翻译出来。为此需要提供 KXR 语言文件。

(>>> [5.3 “KXR 语言文件” 页面 45](#))

参数	说明
上部	smarHMI 侧框上状态键旁边显示的文本（上部）
中间	smarHMI 侧框上状态键旁边显示的文本（中间）
下部	smarHMI 侧框上状态键旁边显示的文本（下部）

参数	说明
Style	<p>状态键格式</p> <ul style="list-style-type: none"> <li>SWITCH 单功能状态键 单功能状态键对应一个键。该键始终具有 PLUS 功能。</li> <li>TOGGLE 双功能状态键 双功能状态键对应 2 个键。上部按键具有 PLUS 功能。下部按键具有 MINUS 功能。</li> </ul>

### 前提条件参数

在该区域中可以定义状态键的激活条件。

参数	说明
安全开关	<p>状态键是否激活取决于使能开关的状态。</p> <ul style="list-style-type: none"> <li><b>True:</b> 按下使能开关后，状态键激活。</li> <li><b>False:</b> 无需使能来激活</li> </ul>
驱动装置 OK	<p>状态键是否激活取决于驱动装置的状态。</p> <ul style="list-style-type: none"> <li><b>True:</b> 在驱动装置接通时，状态键激活。</li> <li><b>False:</b> 无需接通驱动装置来激活</li> </ul>
提交状态	<p>状态键是否激活取决于提交解释器的状态和/或机器人解释器的状态。</p> <ul style="list-style-type: none"> <li><b>Free</b> 在解释器中未选择任何程序。</li> <li><b>Stop</b> 在解释器中已选择一个程序。程序已停止运行。</li> <li><b>重置</b> 在解释器中已选择一个程序。程序已停止运行并重置。</li> <li><b>End</b> 在解释器中已选择一个程序，并且程序已经结束。</li> <li><b>Active</b> 在解释器中已选择一个程序，并且程序正在运行。</li> </ul> <p>可以选择多个状态。如果未选择任何解释器状态，则状态键默认在以下状态下激活：</p> <ul style="list-style-type: none"> <li><b>提交状态 = Active</b></li> <li><b>机器人解释器状态 = Free Stop 重置 End</b></li> </ul>
机器人解释器状态	
ModeOp	<p>状态键是否处于激活状态下取决于运行模式。</p> <ul style="list-style-type: none"> <li><b>T1、T2、Aut、Ext</b></li> </ul> <p>可以选择多种运行模式。如果未选择运行模式，则状态键默认在运行模式 T1、T2 或 AUT 下激活。</p>





事件编辑

在该区域中可以将脚本分配给不同的状态键事件。从下拉列表中选择已为应用定义的所有脚本。  
如要取消分配，请选择下拉列表中的第一个空行。

参数	说明
OnKeyShow	当从主菜单调用状态键时，将运行脚本。
OnKeyDown	当按下具有 PLUS 功能的状态键时，将运行脚本。
OnKeyDownMinus	当按下具有 MINUS 功能的状态键时，将运行脚本。
OnKeyUp	当松开具有 PLUS 功能的状态键时，将运行脚本。
OnKeyUpMinus	当松开具有 MINUS 功能的状态键时，将运行脚本。
OnKeyRepeat	当长按具有 PLUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。
OnKeyRepeatMinus	当长按具有 MINUS 功能的状态键时，将运行脚本。脚本会被重复触发，直到松开状态键为止。时间间隔将会不断缩短。

可用的图标

在该区域中可以为状态键的各种状态分配图标。所有可用图标均列出：

- 如要分配图标，选定所需图标并点击  按钮。
- 如要取消分配，点击  按钮。

参数	说明
默认	默认显示在 smartHMI 侧框上状态键旁边的图标
DownKey	按下具有 PLUS 功能的按键时 smartHMI 侧框上状态键旁边显示的图标
DownKeyMinus	按下具有 MINUS 功能的按键时 smartHMI 侧框上状态键旁边显示的图标

5.2.9 给应用添加和配置新的状态键栏

说明

状态键栏是具有至少一到四个状态键的分组状态键。  
在应用中可以配置任意数量的状态键栏。以这种方式，例如可以通过一栏状态键控制抓手，通过另一栏状态键来切换状态信号。  
状态键栏可以从 KUKA smartHMI 的主菜单中调用以及借助脚本和 SetKeyBar 操作进行更改。  
(>>> [5.2.6.10 “操作： SetKeyBar” 页面 36](#))  
状态键栏的名称可以翻译出来。为此需要提供 KXR 语言文件。  
(>>> [5.3 “KXR 语言文件” 页面 45](#))

前提条件

- 应用的编辑视图已打开。  
(>>> [5.2.4 “打开应用的编辑视图” 页面 17](#))



操作步骤



- 1. 右键单击元素树中的 **StatkeyBars** 文件夹，然后选择 **Add Statkeybar**。一个新的状态键栏被添加到元素树中。
- 2. 选择新的状态键栏。将显示状态键栏（仍为空）的预览及其属性。
- 3. 如要配置状态键栏，点击 **配置** 按钮。一个对话框打开。
- 4. 根据需要配置状态键栏并点击 **Ok** 应用配置。

KeyBar 属性:

参数	说明
名称	状态键栏的唯一名称 如要翻译名称，必须按照以下方式保存名称： <i>KXR 模块名称#状态键栏名称</i>
密钥	用于识别/引用状态键栏的唯一密钥

可用的状态键

在该区域中可以为状态键栏分配各个状态键及其在状态键中的位置。所有可用的状态键均列出：

- 如要分配状态键，选定所需的状态键并点击  按钮。
- 如要取消分配，点击  按钮。

双功能状态键（TOGGLE）占用 2 个位置。

5.2.10 管理图标

说明

可以为工艺状态键分配图标。这些图标必须导入到工艺配置中。然后，导入的图标在所有工艺应用中均可使用。

通过 **组织图标** 对话框可以导入图标或将导入的图标导出。不需要的图标可以再次从配置中删除。

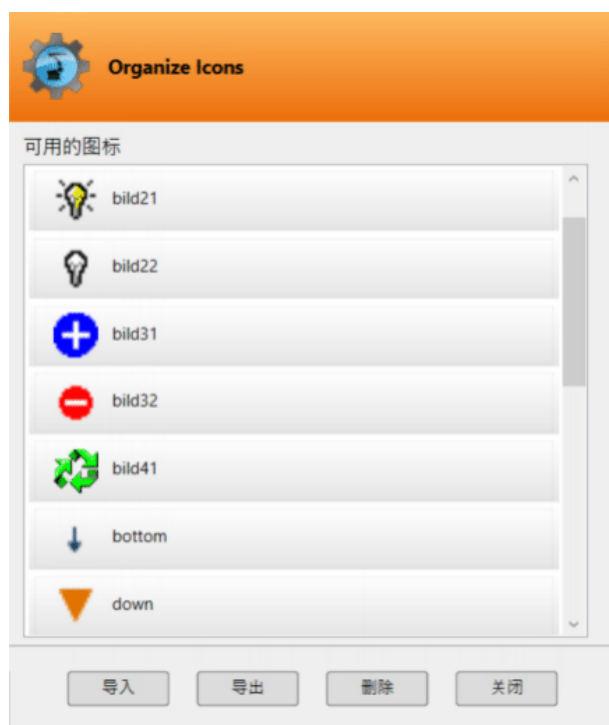



图 5-21: 组织图标 对话框

### 前提条件

- 已在 User-Tech5 编辑器中打开配置。  
(>>> [5.2.2 “打开 User-Tech5 编辑器” 页面 15](#))

### 操作步骤

- 点击  (管理图标) 图标。组织图标 对话框打开。

## 5.2.10.1 创建图标图形的规则

### 说明

为了确保图标图形在 smartHMI 上以最佳方式显示在状态键旁边，在创建图标图形时必须遵守以下规则。

### 图形格式

允许以下图形格式：

- 图标 \*.ICO (推荐)
- Bitmaps \*.BMP
- 便携式网络图形 \*.PNG

### 图像尺寸

根据状态键的格式，需要以下图像尺寸：

- 单功能状态键 (SWITCH)：32 x 32 像素
- 双功能状态键 (TOGGLE)：32 x 96 像素

### 布局图

根据状态键的格式，必须遵守以下布局规则：

- 单功能状态键 (SWITCH)：图形必须居中位于 32 x 32 像素大小的框中。

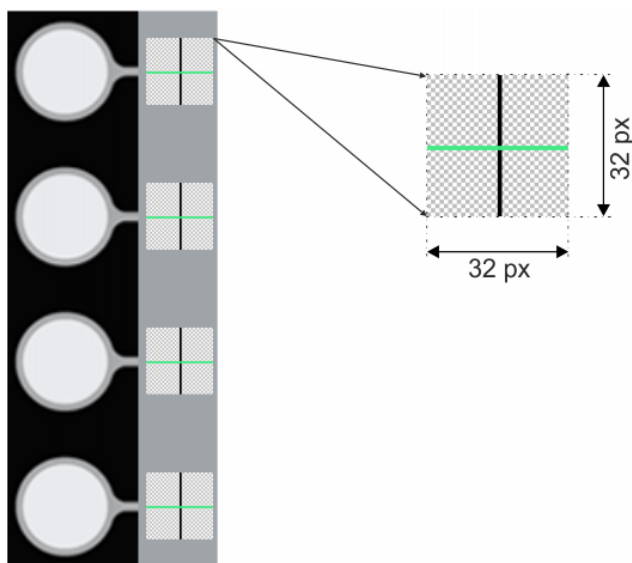


图 5-22：图示居中（单功能状态键）



图 5-23：布局示例（单功能状态键）

- 1 图示具有白色背景
- 2 图示具有透明或灰色背景

- 双功能状态键 (TOGGLE)：32 x 96 像素的图形被分割为 3 个相等高度的 32 x 32 像素大小的组块。

在 smartHMI 上将独立显示这些组块部分。要显示在一个组块中的图形必须相应居中。

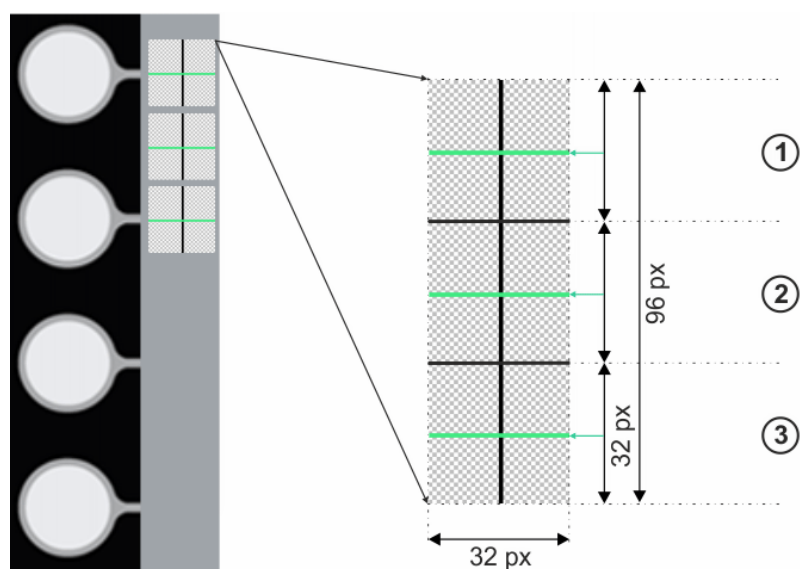


图 5-24：图示居中（双功能状态键）

- 1 图示位于上方
- 2 图示居中（可选）
- 3 图示位于下方

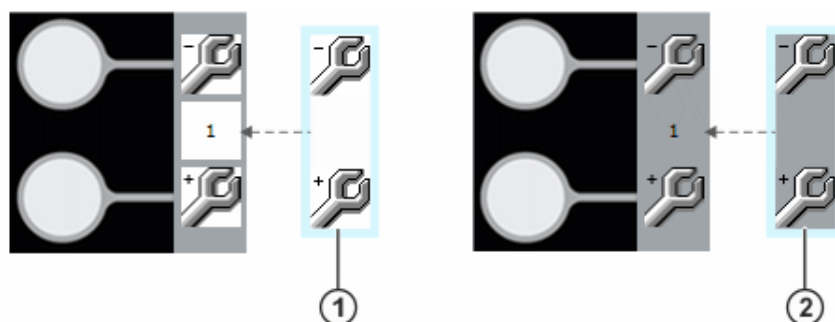


图 5-25：布局示例（双功能状态键）

- 1 图示具有白色背景
- 2 图示具有透明或灰色背景

## 背景颜色

如果所用图形的背景颜色与 smartHMI 颜色有差异，则可在 smartHMI 上看到。

如果在 smartHMI 上只需看到所使用的图标而非周边区域，则根据图形格式可提供下列选项：

- 图形格式 \*.ICO 或 \*.PNG：图标周围的区域定义为透明区域。
- 图形格式 \*.BMP：图标周围的区域按 smartHMI 的颜色调成灰色。
  - 灰度值 RGB：R:160, G:165, B:170

## 5.2.11 生成菜单配置文件

### 说明

为了能够在 smartHMI 上调用一项工艺的行指令和状态键栏，机器人控制器上需要有 2 个菜单配置文件：

- 在编辑程序时，行指令从 **指令** 菜单中调用。

- 状态键栏从主菜单中调用：**配置 > 状态键**

菜单配置文件可以在 User Tech5 编辑器中生成：

- 在菜单配置对话框中可以选择一项工艺的全部或部分可用的行指令和状态键栏。

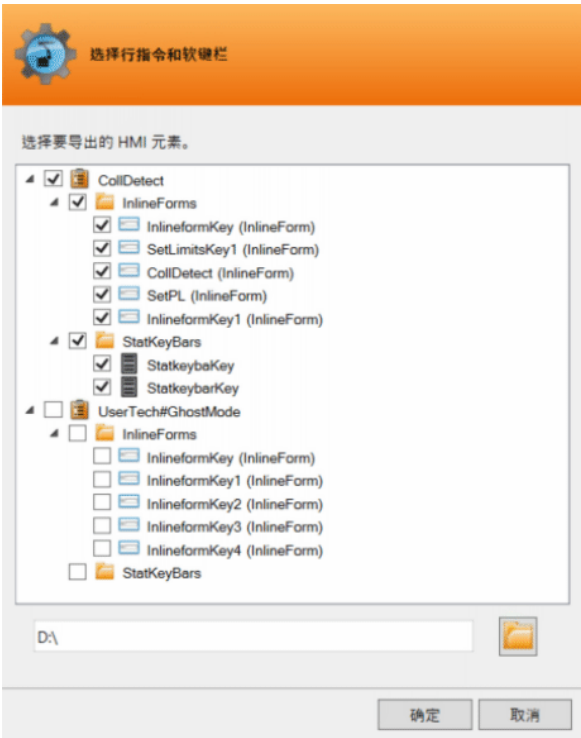



图 5-26：菜单配置的选择对话框

- 即使只选择了行指令或只选择了状态键栏，也始终会生成这两个文件。
- 工艺的配置名称是相应文件名的一部分，例如：
  - SmarthMI.exe.my\_Technology.InlineForms.config
  - SmarthMI.exe.my\_Technology.Statkeys.config
- 所设置的文件存储位置的网络路径被保存并在下次调用对话框时再次显示出来。

前提条件

- 已在 User-Tech5 编辑器中打开配置。  
(>>> 5.2.2 “打开 User-Tech5 编辑器” 页面 15)

操作步骤

1. 点击  (生成菜单配置) 图标。一个对话框打开。
2. 选择所需的行指令和/或状态键栏。
3. 如果尚未显示所需存储位置的网络路径：直接输入路径或通过文件夹搜索来选择路径。
4. 点击 **Ok** 确认选择。

5.3 KXR 语言文件

说明

UI 文本和/或消息文本的翻译保存在 KXR 语言文件中。  
必须为翻译定义以下元素：

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns="http://www.kuka.com/schemas/kxr/2009">
  <module name=" ">
    <uiText key=" ">
      <text xml:lang="de"> </text>
      <text xml:lang="en"> </text>
      <text xml:lang="zh"> </text>
      ...
    </uiText>
    ...
    <message key=" ">
      <text xml:lang="de"> </text>
      <text xml:lang="en"> </text>
      <text xml:lang="zh"> </text>
      ...
    </message>
    ...
  </module>
</resources>
```

元素	说明
<module name	语言文件的名称（可自由选择）
<uiText key	要翻译的 UI 文本的密钥
<message key	要翻译的消息文本的密钥
<text xml:lang="de"	在 smartHMI 上选择德语时显示的文本 通过语种缩写访问文本，此处为“de”。 必须为每个翻译创建一个带有相应语种缩写的条目。 (>>> <a href="#">“语种缩写” 页面 47</a> )

KXR 语言文件中定义的文本密钥可在应用配置中使用，以生成相应语言版本的显示。必须始终指定存储密钥的 KXR 语言文件的名称：

- module name#uiText key

如果访问 KXR 语言文件失败，将显示存储在 XML 工艺文件中的密钥而不是文本。

## 示例

用于编程相对运动的行指令主要需要速度和 X、Y 和 Z 方向的 Real 参数。相应语言版本的参数文本保存在 KXR 语言文件中。

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns="http://www.kuka.com/schemas/kxr/2009">
  <module name="RelMove">
    <uiText key="Velocity">
      <text xml:lang="de">Geschwindigkeit</text>
      <text xml:lang="en">Velocity</text>
    </uiText>
    <uiText key="X">
      <text xml:lang="de">X-Richtung</text>
      <text xml:lang="en">X direction</text>
    </uiText>
    <uiText key="Y">
      <text xml:lang="de">Y-Richtung</text>
      <text xml:lang="en">Y direction</text>
    </uiText>
    <uiText key="Z">
```

```
<text xml:lang="de">Z-Richtung</text>
<text xml:lang="en">Z direction</text>
</uiText>
...
</module>
</resources>
```

参数名称稍后可以按照文件中存储的方式显示在 smartHMI 上。为此，在配置中按如下方式指定例如速度和 X 方向的参数名称：

RelMove#Velocity 0.1 [m/s]

名称	RelMove#Velocity
密钥	Vel_Key
默认	0.1
最大值	1.5
最小值	0.1
步幅	0.1
单位	m/s

RelMove#X 1 [mm]

名称	RelMove#X
密钥	X
默认	1
最大值	100
最小值	0
步幅	1
单位	mm

图 5-27：KXR 密钥作为参数名称

语种缩写

语言	语种缩写
中文	zh
丹麦语	da
德语	de
英语	en
芬兰语	fi
法语	fr
希腊语	el
意大利语	it
日语	ja
韩语	ko
荷兰语	nl
波兰语	pl
葡萄牙语	pt
罗马尼亚语	ro
俄语	ru
瑞典语	sv
斯洛伐克语	sk
斯洛文尼亚语	sl
西班牙语	es
捷克语	cs
土耳其语	tr
匈牙利语	hu
越南语	vi

### 5.4 载入配置

#### 说明

工艺配置（即导出的 XML 文件）可以重新载入到编辑器中。

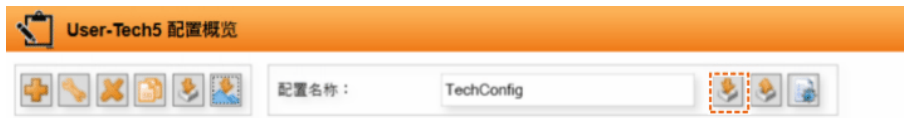


图 5-28：载入配置

另外也可以载入“旧”XML 文件格式 KFDx 的配置。配置会自动转换为当前的 XML 格式。


载入 KFDx 文件时须注意下面几点：

- 在 KUKA.UserTech 5.0 中，应用不使用全局数据。不导入全局数据。全局数据可能需要单独导入到应用中。  
(>>> 5.6 “从配置中导入单个元素” 页面 49)
- 用于在 KFDx 中设置状态键或状态键栏的“Set”指令会自动转换为“SetKey”或“SetKeyBar”操作。

#### 前提条件

- User-Tech5 编辑器已打开。  
(>>> 5.2.2 “打开 User-Tech5 编辑器” 页面 15)
- XML 工艺文件位于具有网络访问权限的目录中。

#### 操作步骤

1. 点击  （载入配置）图标。打开 窗口打开。
2. 如有必要，选择文件格式：针对当前 UserTech 版本的配置选择 XML，或针对“旧”配置选择 KFDx。
3. 导航至 XML 工艺文件，将其选中并点击**打开**确认选择。工艺被载入并显示在 User-Tech5 编辑器中。



User-Tech5 编辑器中的工艺名称不一定与 XML 工艺文件的名称相同。

### 5.5 从配置中导入应用

#### 说明

从工艺配置中可以将一个（或多个）应用导入编辑器的配置中。



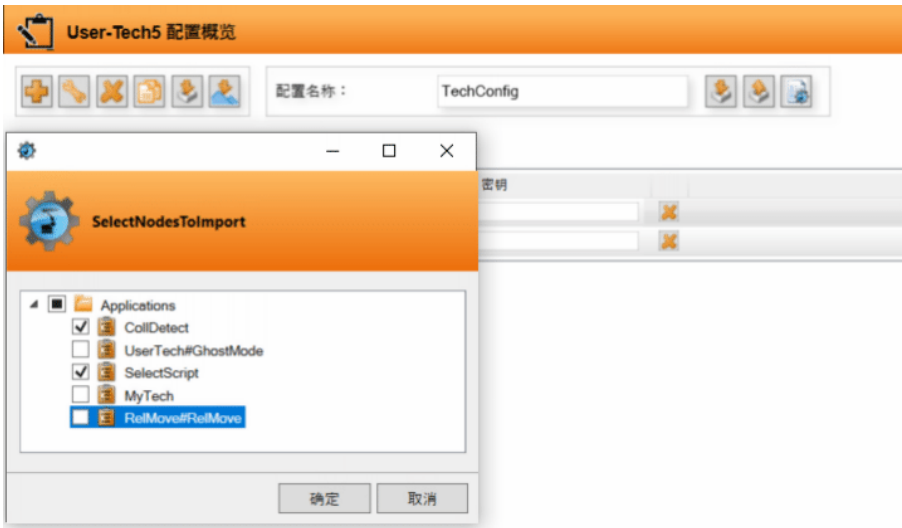


图 5-29：导入应用

另外也可以导入“旧”XML 文件格式 KFDx 的应用。应用会自动转换为当前的 XML 格式。


从 KFDx 格式导入时必须注意以下几点：

- 在 KUKA.UserTech 5.0 中，应用不使用全局数据。不导入全局数据。全局数据可能需要单独导入到应用中。  
(>>> 5.6 “从配置中导入单个元素” 页面 49)
- 用于在 KFDx 中设置状态键或状态键栏的“Set”指令会自动转换为“SetKey”或“SetKeyBar”操作。

前提条件

- User-Tech5 编辑器已打开。  
(>>> 5.2.2 “打开 User-Tech5 编辑器” 页面 15)
- XML 工艺文件位于具有网络访问权限的目录中。

操作步骤

1. 点击  (导入应用) 图标。打开 窗口打开。
2. 如有必要，选择文件格式：针对当前 UserTech 版本的配置选择 XML，或针对“旧”配置选择 KFDx。
3. 导航至 XML 工艺文件，将其选中并点击**打开**确认。将打开一个用于选择应用的对话框。
4. 选择所需应用并点击 **Ok** 确认选择。所选应用在 可用的应用 区域中被添加到编辑器的配置中。
5. 如有必要，更改所添加应用的名称和密钥。

5.6 从配置中导入单个元素

说明

从工艺配置中可以将应用的单个元素（例如参数或脚本）导入编辑器中的应用中。以“旧”XML 文件格式 KFDx 配置的应用的全局元素也可以事后通过此处描述的操作步骤导入。

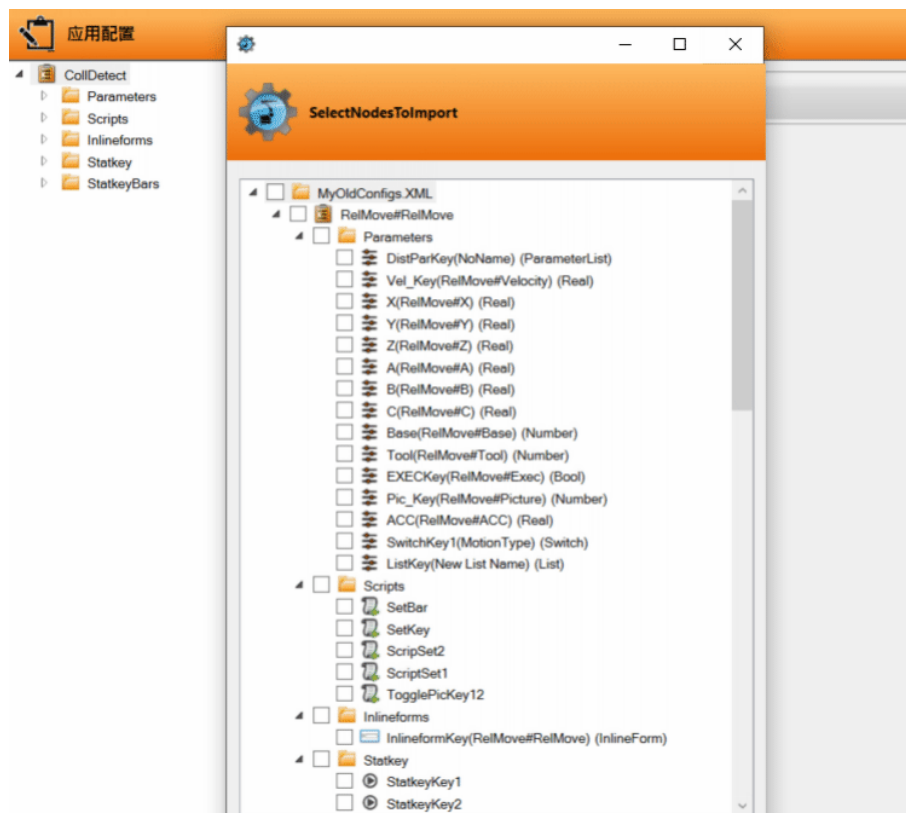


图 5-30：将单个元素导入应用

### 前提条件

- 应用的编辑视图已打开。  
(>>> [5.2.4 “打开应用的编辑视图” 页面 17](#))
- XML 工艺文件位于具有网络访问权限的目录中。

### 操作步骤

1. 右键单击元素树中的应用并选择**导入**。**打开** 窗口打开。
2. 如有必要，选择文件格式：针对当前 UserTech 版本的配置选择 XML，或针对“旧”配置选择 KFDx。
3. 导航至 XML 工艺文件，将其选中并点击**打开**确认。将打开一个用于选择元素的对话框。
4. 选择所需元素并点击 **Ok** 确认选择。所选元素被添加到应用中。

## 5.7 导出 XML 工艺文件


### 说明

工艺配置可以导出并保存为 XML 文件。

### 前提条件

- 已在 User-Tech5 编辑器中打开配置。  
(>>> [5.2.2 “打开 User-Tech5 编辑器” 页面 15](#))

### 操作步骤

1. 点击  (**导出配置**) 图标。**另存为** 窗口打开。
2. 指定文件名并导航至所需的存储位置。点击 **保存** 确认。

5.8 通过 KUKA.WorkVisual 项目将工艺传输至控制器

说明

为了测试行指令和/或状态键，可以通过 WorkVisual 项目将工艺传输至机器人控制器。生成代码时，所需文件会自动存储在 C:\KRC\TP\UserTech\ 下的指定目录下：

- SmarthMI：菜单配置文件
- UserTechResources：XML 工艺文件和所有图标（图像文件）

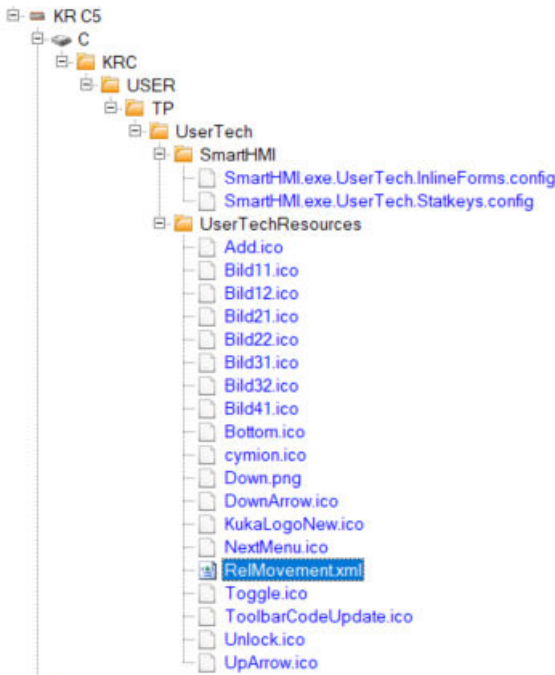


图 5-31：代码生成后的目录结构

操作步骤

- 将 WorkVisual 项目传输至机器人控制器。



关于 WorkVisual 中流程步骤的信息见 KUKA.WorkVisual 的文档。

5.9 使用 KUKA.OptionPackageEditor 创建备选软件包

5.9.1 关于 KUKA.OptionPackageEditor 的提示

本文档仅说明与 KUKA.UserTech 相关的 KUKA.OptionPackageEditor 中的流程和参数。

如果能够以不同方式实现目的，则仅介绍其中一种方式。



关于 KUKA.OptionPackageEditor 的详细信息请参见 KUKA.OptionPackageEditor 文档。

5.9.2 创建备选软件包

操作步骤

1. 选择菜单序列文件 > 新建...。  
创建新的备选软件包 窗口打开。

2. 为软件包分配一个名称并且选择保存位置。点击 **保存** 确认。  
此时在窗口左侧已选中 **概况** 区域。

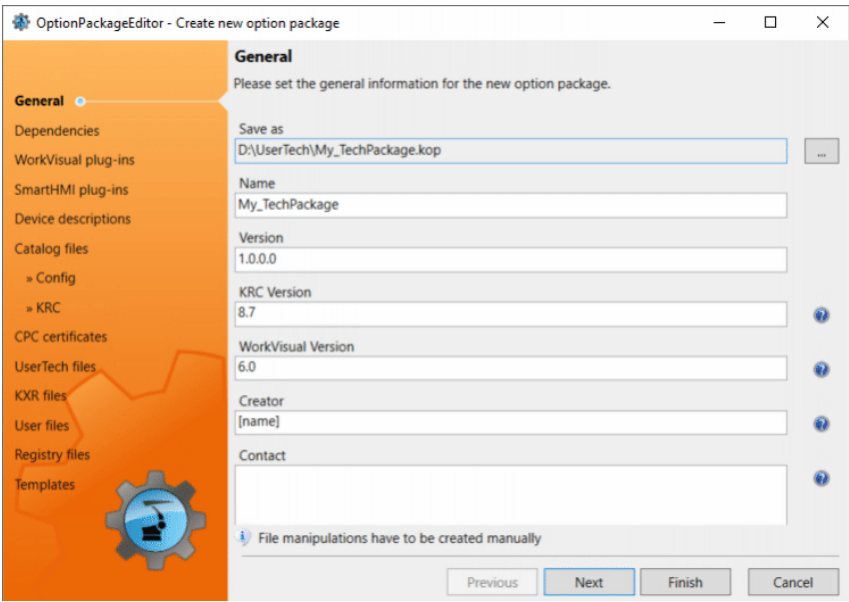


图 5-32：创建备选软件包

3. 填写 **创建者** 栏。



图中的 *[name]* 只是一个占位符。应填写一个有效的姓名。

4. 按如下方式填写这些栏位：

栏位	更多信息
KRC 版本	指定“8.7”或暂时留空。 (>>> <a href="#">5.9.3 “定义对 KSS 的依赖性” 页面 52</a> )
WorkVisual 版本	指定“6.0”。

5. 点击**完成**。创建新的软件包并在窗口**备选软件包结构**中显示。

5.9.3 定义对 KSS 的依赖性

操作步骤

- 在 **备选软件包结构** 窗口中的**常规**元素下方：  
展开 **依赖性** 元素。
- 双击 **KSS/VSS**。  
**KRC 依赖性**窗口打开。
- 在选择栏顶部设置 **KSS**。
- 对于 **支持的版本**，请指定 8.7。

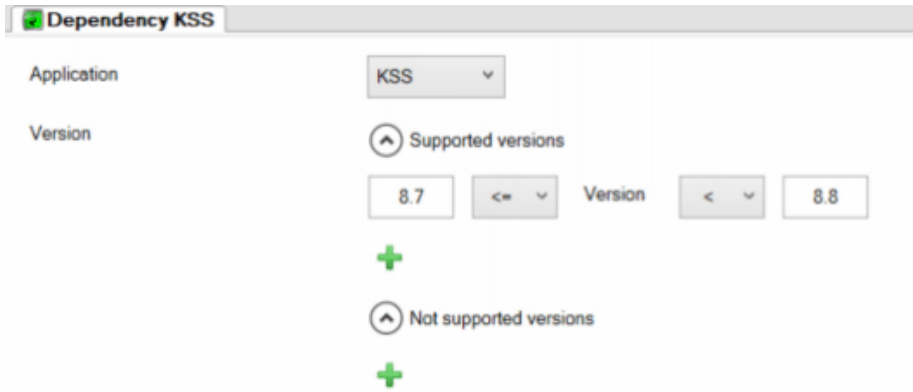


图 5-33：定义对 KSS 的依赖性

5.9.4 定义对 KUKA.UserTech 的依赖性

说明

使用 KUKA.UserTech 创建的工艺始终依赖于 KUKA.UserTech，因为它需要访问 KUKA.UserTech 的某些元素。  
下面为备选软件包定义的依赖性可确保在安装过程中提醒用户还需要安装 KUKA.UserTech。

前提条件

- KUKA.UserTech 的 KOP 文件位于具有网络访问权限的目录中

操作步骤

1. 在 备选软件包结构 窗口中的常规元素下方：  
右键单击 依赖性 元素并在上下文菜单中选择 添加 KOP 的依赖性...。  
一个窗口打开。
2. 导航至 KUKA.UserTech 的 KOP 文件，将其选中并在窗口点击打开。
3. 窗口关闭，依赖性... 窗口打开。
4. 对于 支持的版本，点击从 KOP 中应用。  
将自动读取并显示 KUKA.UserTech 的版本。

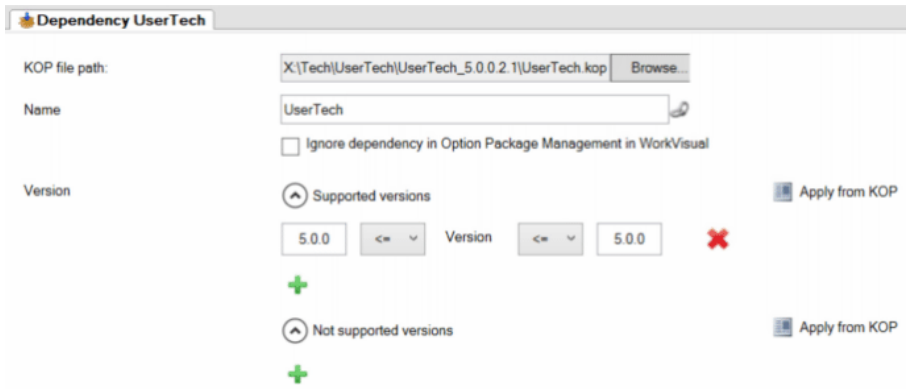


图 5-34：定义对 UserTech 的依赖性（路径 = 示例）

### 5.9.5 添加 XML 工艺文件

#### 前提条件

- 文件位于具有网络访问权限的目录中。

#### 操作步骤

1. 在**备选软件包结构**窗口：  
在**常规**下右键单击 **UserTech** 目录并选择 **添加文件...**。
2. 导航至所需文件，将其选中并在窗口中点击**打开**。

#### 示例

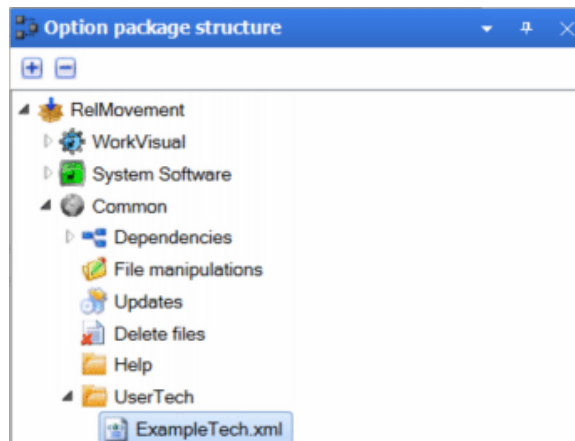


图 5-35: 已添加 XML 工艺文件

### 5.9.6 添加图像文件

#### 前提条件

- 文件位于具有网络访问权限的目录中。

#### 操作步骤

1. 在**备选软件包结构**窗口：  
在**常规**下右键单击 **UserTech** 目录并选择 **添加文件...**。
2. 导航至所需文件，将其选中并在窗口中点击**打开**。

示例

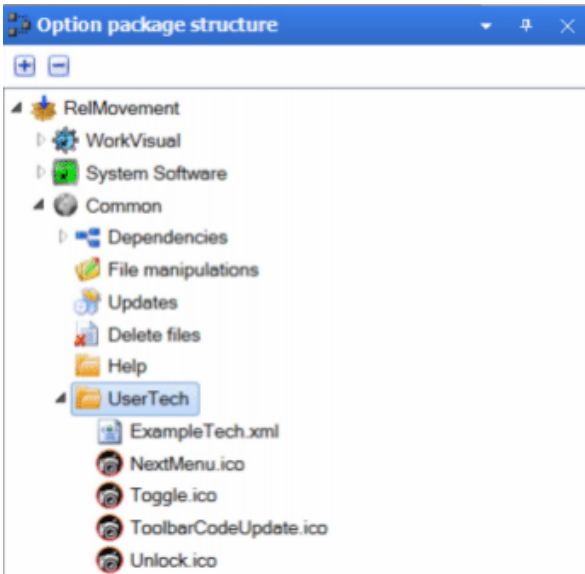


图 5-36：已添加图像文件

5.9.7 添加菜单配置文件

前提条件

- 文件位于具有网络访问权限的目录中。

操作步骤

1. 在**备选软件包结构**窗口：  
在**系统软件**下右键单击 **Plugins** 目录并选择 **添加文件...**。
2. 导航至所需文件，将其选中并在窗口中点击**打开**。

示例



图 5-37：已添加菜单配置文件

5.9.8 添加 KXR 语言文件

前提条件

- 文件位于具有网络访问权限的目录中。

操作步骤

1. 在**备选软件包结构**窗口：  
在**系统软件**下右键单击 **Kxr** 目录并选择 **添加文件...**。

2. 导航至所需文件，将其选中并在窗口中点击**打开**。

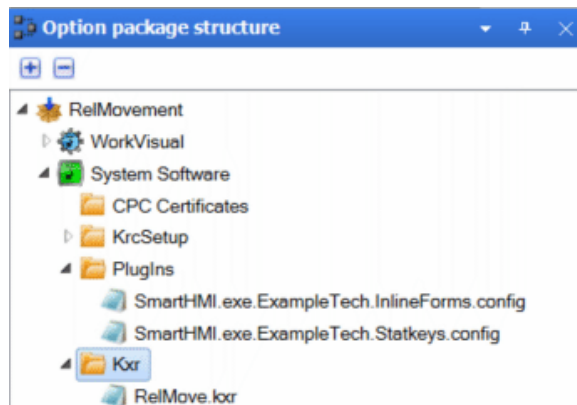


图 5-38：已添加 KXR 语言文件

### 5.9.9 添加 KRL 文件

#### 说明

如果用户为其配置的行指令创建了附加的 KRL 文件，则必须将这些文件传输到机器人控制器。

这通过将 KRL 文件导入备选软件包来完成。通过这种方式，这些文件与选项一起传输到机器人控制器。

#### 前提条件

- 文件位于具有网络访问权限的目录中。

#### 操作步骤

1. 在**备选软件包结构**窗口：  
在 **WorkVisual** > **Catalog** > **Option** > **KRC** > **R1** > **TP** 下创建一个目录。  
目录的名称（在示例中为 **RelMove**）是任意的。  
如要添加目录，右键单击并选择 **新文件夹...**。
2. 必要时创建其他子目录。KRL 文件的存储位置不由系统定义，可以根据需要进行设置。
3. 添加 KRL 文件：右键单击目录并选择 **添加文件...**。



通过右键单击目录和 **添加文件夹...** 可以导入完整的文件夹结构。



示例

配置

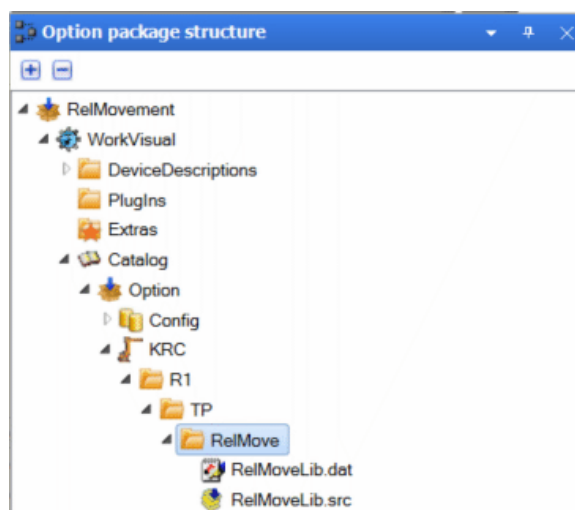


图 5-39: 已添加 KRL 文件



## 6 KUKA 客户服务

### 6.1 技术支持咨询

#### 引言

本文档提供有关机器运行及操作的信息，并可帮助您排除故障。当地各分支机构将乐于为您提供详细咨询。

#### 信息

##### 提供技术支持时需要以下信息：

- 问题描述，包括故障持续时间和频率的信息
- 有关整个系统的硬件和软件组件的尽可能详尽的信息

以下列表给出了在许多情况下相关的信息：

- 机器人运动系统（例如机械臂）的型号及序列号
- 控制器类型及序列号
- 外部管线包类型及序列号
- 系统软件名称及版本
- 更多/其他软件组件的名称及版本或修正版
- 系统软件的诊断包

针对 KUKA Sunrise 另外还需要：现有项目，包括应用程序

针对早于 V8 的 KUKA System Software 版本：软件存档（诊断包在此尚不可用。）

- 现有的应用程序
- 现有的附加轴

### 6.2 KUKA 客户支持系统

当地分支机构的联系方式请参见网页：

[www.kuka.com/customer-service-contacts](http://www.kuka.com/customer-service-contacts)



## 索引

### K

KFDx.....	6
KRL.....	6
KRL 文件, 附加.....	56
KRL 文件, 添加.....	56
KUKA 客户服务.....	59
KUKA 客户支持系统.....	59
KXR 语言文件.....	45

### S

smartHMI.....	6
smartPAD.....	6
String.....	6

### 安

安全.....	9
安全提示.....	5
安装.....	11

### 产

产品说明.....	7
-----------	---

### 技

技术支持咨询.....	59
-------------	----

### 开

开源.....	6
---------	---

### 目

目标群体.....	5
-----------	---

### 培

培训.....	5
---------	---

### 配

配置.....	13
---------	----

### 术

术语, 所用的.....	6
--------------	---

### 所

所用术语.....	6
-----------	---

## 提

提示.....	5
---------	---

## 违

违规使用.....	7
-----------	---

## 文

文档, 工业机器人.....	5
----------------	---

## 系

系统前提条件.....	11
系统要求	
软件.....	11
硬件.....	11

## 许

许可证.....	6
----------	---

## 引

引言.....	5
---------	---

## 预

预期用途.....	7
-----------	---

## 诊

诊断包.....	59
----------	----

## 知

知识, 所需的.....	5
--------------	---