

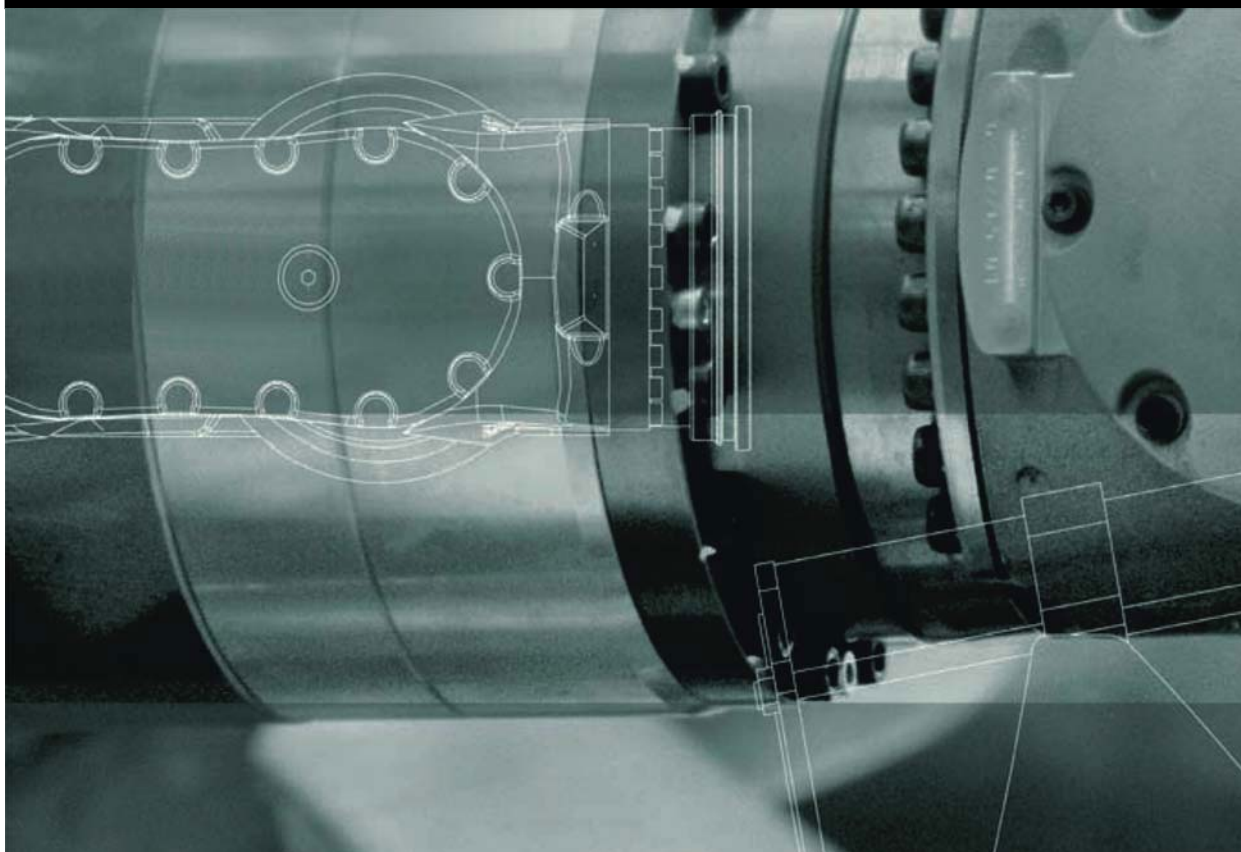


KUKA System Technology

KUKA Roboter GmbH

KUKA.UserTech 3.3

KUKA 系统软件 8.2、8.3 和 8.4



发布日期：06.03.2016

版本：KST UserTech 3.3 V4



© 版权 2016

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
德国

此文献或节选只有在征得库卡机器人集团公司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，无权要求库卡公司提供这些功能。

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。

本文件为原版文件的翻译。

KIM-PS5-DOC

出版物： Pub KST UserTech 3.3 (PDF) zh

书页构造： KST UserTech 3.3 V1.1

版本： KST UserTech 3.3 V4

目录

1	引言	7
1.1	目标群	7
1.2	工业机器人文献	7
1.3	提示的图示	7
1.4	所用概念	8
1.5	商标	8
2	产品说明	9
2.1	UserTech 概览	9
3	安全	11
4	安装	13
4.1	系统前提条件	13
4.2	初始化或更新 UserTech	13
4.3	卸载 UserTech	13
5	操作	15
5.1	更新 UserTech 界面	15
6	编程	17
6.1	KFD 句法概览	17
6.2	字符和字体	17
6.3	重要的 KFD 概念	18
6.3.1	特殊字符	18
6.3.2	字符串	19
6.3.3	参数引用	19
6.3.4	命名约定和关键词	19
6.3.5	简单数据类型	20
6.3.6	适用范围	20
6.4	工艺编程	21
6.4.1	工艺设计	21
6.4.2	创建 KFD 文件	22
6.4.3	DEFTP ... ENDTP	22
6.4.4	DECL PARAM	23
6.4.5	字段类型	24
6.4.5.1	静态字段类型	24
6.4.5.2	自由字段类型	24
6.4.5.3	字段类型为名称	24
6.4.5.4	字段类型为数字	25
6.4.5.5	字段类型 REAL	26
6.4.5.6	字段类型为列表	27
6.4.5.7	字段类型 BOOL	28
6.4.6	DECL FOLD	30
6.4.7	生成 KRL 代码	30
6.4.8	DECL INLINEFORM	30
6.4.9	格式类型	32
6.4.9.1	格式类型 WYSIWYG	32
6.4.9.2	格式类型 SUB	33

6.4.9.3	格式类型 DSUB	33
6.4.9.4	格式类型 FCT	34
6.4.9.5	格式类型 DFCT	34
6.4.9.6	格式类型 ASS	35
6.4.9.7	格式类型 ASSAGG	35
6.5	编程参数列表	36
6.5.1	DECL PARAM PL_	36
6.5.2	定义结构数据类型	37
6.5.3	DECL PLIST	37
6.6	工艺键编程	38
6.6.1	设计工艺键	38
6.6.2	DECL STATKEY	38
6.6.2.1	图像大小	41
6.6.2.2	布局 (KSS 8.3 以上)	41
6.6.2.3	得出激活选择的十进制数值	43
6.6.3	DECL STATKEYBAR	44
6.6.4	SET	45
6.7	脚本编程	45
6.7.1	DEFSCRIPT ... ENDSCRIPT	45
6.7.2	SETVAR	45
6.7.3	SHOWVAR	46
6.7.4	REDECL	46
6.7.5	DO	47
6.7.6	MESSAGE	47
6.7.7	SWITCH ... CASE (ELSE) ... END SWITCH	47
6.7.8	SWITCH DIALOG ... CASE ... END SWITCH	48
6.7.9	预定义脚本	49
6.7.10	信息和按键的语言系统密钥	49
6.8	工艺集成于 KUKA smarthMI	49
6.8.1	配置文件 SmartHMI.User.Config	50
6.8.2	定义菜单元素	51
7	存档	53
7.1	文件存档	53
7.2	恢复文件	53
8	示例	55
8.1	LASER 工艺	55
8.2	参数列表的示例程序	56
8.3	通过操作调用脚本	59
8.3.1	打开和关闭联机表单之后的脚本调用	60
8.3.2	更改和创建联机表单之后的脚本调用	60
8.3.3	更换联机表单之后的脚本调用	62
8.4	集成工艺的示例	63
8.4.1	TWINKLE 工艺	63
8.4.1.1	将 TWINKLE 工艺集成到菜单中	63
8.4.1.2	将 TWINKLE 工艺集成到子菜单中	64
8.4.2	GLUE 工艺	65
8.4.2.1	将 GLUE 工艺集成到菜单中	66
8.4.3	DISPSET 工艺	67

8.4.3.1	将 DISPSET 工艺集成到菜单中	68
9	KUKA Service	71
9.1	技术支持咨询	71
9.2	库卡客户支持系统	71
	索引	79

1 引言

1.1 目标群

本文档针对具备下列知识的用户：

- 机器人控制器的高级系统知识
- 高级 KRL 编程知识



我们推荐客户在库卡学院进行培训，以便能以最佳方式使用我们的产品。有关培训项目信息请访问公司主页 www.kuka.com 或直接在公司分支机构处获得。

1.2 工业机器人文献

工业机器人文献由以下部分组成：

- 机器人机械装置文献
- 机器人控制系统文献
- 系统软件操作及编程指南
- 选项及附件指南
- 保存在数据载体上的部件目录

每份指南均独立成篇。

1.3 提示的图示

安全性

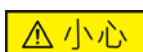
这些说明是安全提示，**必须** 遵守。



该提示表示，如果不采取预防措施，则很可能**将**导致死亡或严重的身体伤害。



该提示表示，如果不采取预防措施，则**可能**导致死亡或严重的身体伤害。



该提示表示，如果不采取预防措施，则**可能**导致轻微的身体伤害。



该提示表示，如果不采取预防措施，则**可能**导致财产损失。



该提示包含安全相关信息的说明或通用安全措施。
该提示不针对个别的危险或个别的预防措施。

此提示提醒您注意用于预防或消除紧急情况或故障的操作步骤：

安全须知

必须严格遵守用此提示所标记的操作步骤。

提示

这些提示可使工作便利或提供进一步信息的说明。



用来使工作便利或提供进一步说明信息的提示。

1.4 所用概念

概念	说明
KFD	KUKA 表单说明
KFD 文件	工艺中所描述的文件格式
KRL	库卡机器人编程语言（KUKA Robot Language）
KUKA smartHMI	KUKA 系统软件操作界面（KUKA smart 人机对话接口）
KUKA smartPAD	用于工业机器人的手持式编程器

1.5 商标

Windows 是微软公司（Microsoft Corporation）的商标。

2 产品说明

2.1 UserTech 概览

功能

UserTech 是一个可后续载入的应用程序包，具有下列功能：

- 定义用户特定的联机表单
- 定义用户特定的信息
- 定义用户特定的工艺键和按键
- 定义用户特定的联机表单和工艺键脚本
- 定义用户特定的联机表单折叠夹（Fold）
- 将用户特定的联机表单和工艺键集成到 KUKA smartHMI 操作界面中

3 安全

该文件包括针对此处所述软件的安全提示。

有关工业机器人的基本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。



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。否则可能会造成人员死亡、严重身体伤害或巨大的财产损失。

4 安装

4.1 系统前提条件

软件 ■ 库卡系统软件 8.2、8.3 或 8.4

4.2 初始化或更新 UserTech



我们建议在软件更新前将所有相关数据存档。

前提条件

- 专家用户组
- 运行方式 T1 或 T2
- 没有选定任何程序。
- 含待安装软件的 U 盘
 - ZIP 文件必须已经解压。
 - 在具有单个文件的目录中不允许具有其他的文件。

注意

建议使用库卡 U 盘。如果使用其它制造商的 U 盘，则数据可能会丢失。

操作步骤

1. 将 U 盘插在机器人控制系统或 smartPAD 上。
2. 在主菜单中选择**投入运行** > **辅助软件**。
3. 按**新软件**：在**名称**列中必须显示选项 **UserTech**，而在**路径**列中必须显示驱动器 **E:** 或 **K:**。
否则按下**刷新**。
4. 如果此时显示上述条目，则继续进行步骤 5。
否则必须先配置待安装程序的路径：
 - a. 按下按键**配置**。
 - b. 在区域**选项的安装路径**中选一个行。
提示：如果该行已经包含一个路径，则该路径被覆盖。
 - c. 按下**路径选择**。即显示现有的驱动器。
 - d. 如果 U 盘插接到机器人控制系统上：在 **E:** 上选中软件所在的目录。
如果 U 盘插接到 smartPAD 上：则 **K:** 替代 **E:**
 - e. 按下**保存**。再次显示区域**选项的安装路径**。它此时含有新的路径。
 - f. 选中带有新路径的行并再次按下**保存**。
5. 选中选项 **UserTech**，然后点击**安装**。点击**是**确认安全询问。
6. 用 **OK** 确认重启请求。
7. 拔出 U 盘。
8. 重启机器人控制系统。

LOG 文件 C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。

4.3 卸载 UserTech



我们建议在软件卸载前将所有相关数据存档。

前提

- 专家用户组

■ 运行方式 T 1 或 T 2

操作步骤

1. 在主菜单中选择**投入运行** > **辅助软件**。
2. 选中选项 **UserTech**，然后点击**卸载**。用**是**回答安全询问。卸载准备就绪。
3. 重新启动机器人控制系统。卸载继续进行并结束。

LOG 文件

C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。

5 操作

5.1 更新 UserTech 界面

说明	<p>通过此功能可以在机器人控制系统运行过程中更新 UserTech 界面，比如显示用户新定义的联机表单。</p> <p>UserTech 界面将被重新初始化，而无需重新启动系统。重新初始化进程会显示在信息窗口中。</p>
前提条件	<ul style="list-style-type: none">■ 专家用户组
操作步骤	<ul style="list-style-type: none">■ 在主菜单中选择配置 > 其它（或工具） > UserTech 重新初始化。

6 编程

6.1 KFD 句法概览

工艺	
DEFTP ... ENDTP	(>>> 6.4.3 "DEFTP ... ENDTP" 页码 22)

联机表格	
DECL FOLD	(>>> 6.4.6 "DECL FOLD" 页码 30)
DECL INLINEFORM	(>>> 6.4.8 "DECL INLINEFORM" 页码 30)
DECL PARAM	(>>> 6.4.4 "DECL PARAM" 页码 23)

参数表	
DECL PARAM PL_	(>>> 6.5.1 "DECL PARAM PL_" 页码 36)
DECL PLIST	(>>> 6.5.3 "DECL PLIST" 页码 37)

工艺键	
DECL STATKEY	(>>> 6.6.2 "DECL STATKEY" 页码 38)
DECL STATKEYBAR	(>>> 6.6.3 "DECL STATKEYBAR" 页码 44)
SET	(>>> 6.6.4 "SET" 页码 45)

脚本	
DEFSCRIPT ... ENDSCRIPT	(>>> 6.7.1 "DEFSCRIPT ... ENDSCRIPT" 页码 45)
DO	(>>> 6.7.5 "DO" 页码 47)
MESSAGE	(>>> 6.7.6 "MESSAGE" 页码 47)
REDECL	(>>> 6.7.4 "REDECL" 页码 46)
SET	(>>> 6.6.4 "SET" 页码 45)
SETVAR	(>>> 6.7.2 "SETVAR" 页码 45)
SHOWVAR	(>>> 6.7.3 "SHOWVAR" 页码 46)
SWITCH ... CASE (ELSE) ... ENDSWITCH	(>>> 6.7.7 "SWITCH ... CASE (ELSE) ... ENDSWITCH" 页码 47)
SWITCH DIALOG ... CASE ... ENDSWITCH	(>>> 6.7.8 "SWITCH DIALOG ... CASE ... ENDSWITCH" 页码 48)
ACCEPTINLINEFORM	(>>> 6.7.9 "预定义脚本" 页码 49)
CANCELINLINEFORM	
END	
NOTHING	

6.2 字符和字体

在句法说明中将使用下列字符和字体：

句法元素	显示
KRL 代码	<div><div>■ 字体为 Courier</div><div>■ 大写</div></div> <div>示例： GLOBAL; ANIN ON; OFFSET</div>
必须由程序相关说明数据替换的元素	<div><div>■ 斜体</div><div>■ 大写 / 小写</div></div> <div>示例： <i>距离</i>; <i>时间</i>; <i>格式</i></div>
可选元素	<div><div>■ 在尖括号中</div></div> <div>示例： <STEP <i>步幅</i>></div>
相互排斥的元素	<div><div>■ 通过符号 “ ” 分隔</div></div> <div>示例： IN OUT</div>

6.3 重要的 KFD 概念

6.3.1 特殊字符

在 KFD 代码中可使用下列特殊字符：

字符	说明
;	<div>表示一条注释</div> <div>示例：</div> <div><div>■ ; 这是一条注释</div></div>
_	<div>在 KFD 中的指令必须位于同一程序行中。</div> <div>使用 “_” 可换行继续程序行。在前一程序行和字符 “_” 之间必须有一个空格。</div> <div>示例：</div> <div><div>■ decl _</div><div>int _</div><div>zahl</div><div>(= 程序行： decl int 整数)</div></div>
/	<div>取消下一字符的特殊功能。字符 “/” 将不会在字符序列中显出。</div> <div>示例：</div> <div><div>■ /%: 取消字符 “%” 用来显示参数引用的功能。</div><div>■ /;: 取消字符 “;” 用来显示注释的功能。</div></div>

字符	说明
%	<p>表示参数引用</p> <p>(>>> 6.3.3 “参数引用” 页码 19)</p> <p>参数引用之后必须添加字符 “/” 或空格。若参数引用后还有其他字符或另外的语句，则在空格之后必须添加第二个空格。</p> <p>结束参数引用的空格将不会在字符序列中显示出来。</p> <p>示例：</p> <ul style="list-style-type: none"> ■ %INLINEFORM/ ■ %INLINEFORM
#	<p>表示赋值</p> <p>示例：</p> <ul style="list-style-type: none"> ■ #%INLINEFORM ■ #TOGGLE

6.3.2 字符串

字符串是长度固定或可变的字母数字的字符序列。

在 KFD 中适用下列规则：

- 字符串不可含有逗号。
- 字符串不可含有换行，就是说必须在一行中。
- 字符串始终用双引号括住，例如 “这是一个字符串序列”。
- 如果字符串中含有引号，则必须通过字符 “/” 表示其作为特殊字符的含义，例如 “字符 /”。



针对字段类型 BOOL 编程时还应遵守附加规则。(>>> 6.4.5.7 “字段类型 BOOL” 页码 28)

6.3.3 参数引用

参数引用由百分比符号和参数名称组成。百分比符号使参数名称不会文本而是作为数值输出。

下面的参数引用是预定义的：

名称	说明
%TP	<p>工艺名称</p> <p>每个联机表单中的第一个参数</p>
%INLINEFORM	<p>联机表单的名称</p> <p>每个联机表单中的第二个参数</p>
%MODULE	导航器中的程序名称


6.3.4 命名约定和关键词

名称

KFD 中的名称示例：变量名称、功能名称、子程序名称

- KFD 中的名称长度最多允许 24 个字符。
- KFD 中的名称允许含有字母 (A - Z)、数字 (0 - 9) 以及特殊字符 “_” 和 “\$”。


- KFD 中的名称不允许以数字开头。
- KFD 中的名称不允许为关键词。

 所有系统变量的名称均以 \$ 符号开头。为避免混淆，针对用户定义的变量来说，其名称将不以此符号开头。

关键词

关键词是含有指定含义的字母序列。不可将其在程序中按其他含义使用。大小写无关紧要。关键词在任何书写方式中均作为关键词。

示例 字母序列 CASE 是 KRL 句法 SWITCH...CASE...ENDSWITCH 的组成部分。因此 CASE 不可作为其他用途使用，例如变量名称等。

 不可使用针对 KRL 备用的关键词。其他关于关键词的信息请见系统集成商操作及编程指南。

6.3.5 简单数据类型


下列简单的数据类型已预定义：

数据类型	关键词	说明
Integer	INT	整数 <div>■ $-2^{31}-1 \dots 2^{31}-1$</div> <div>示例： 1; 32; 345</div>
Real	REAL	浮点数 <div>■ $+1.1\text{E}-38 \dots +3.4\text{E}+38$</div> <div>示例： 1.43; 38.50; 300.25</div>
Boolean	BOOL	逻辑状态 <div>■ TRUE</div> <div>■ FALSE</div>
Character	CHAR	1 个字符 <div>■ ASCII 字符</div> <div>示例： "A"; "1"; "q"</div>

6.3.6 适用范围

如果变量和数据对象在指令 DEFTP 和 ENDTP 之间进行了说明，则其适用于局部范围，就是说适用于一项定义的工艺。

如果变量和数据对象在指令 DEFTP 和 ENDTP 之外进行了说明，则其适用于全局范围，就是说适用于所有的工艺。

 建议将所有全局变量和数据对象汇总到一个 KFD 文件中。

6.4 工艺编程

概览

步骤	说明
1	工艺设计。 (>>> 6.4.1 “工艺设计” 页码 21)
2	创建 KFD 文件。 (>>> 6.4.2 “创建 KFD 文件” 页码 22)
3	定义工艺。 (>>> 6.4.3 “DEFTP ... ENFTP” 页码 22)
4	定义参数。 (>>> 6.4.4 “DECL PARAM” 页码 23)
5	定义参数表。 (>>> 6.5 “编程参数列表” 页码 36)
6	准备脚本。 (>>> 6.7.1 “DEFSCRIPT ... ENSCRIPT” 页码 45)
7	创建折叠夹 (Fold)。 (>>> 6.4.6 “DECL FOLD” 页码 30)
8	生成 KRL 代码。 (>>> 6.4.7 “生成 KRL 代码” 页码 30)
9	定义联机表格。 (>>> 6.4.8 “DECL INLINEFORM” 页码 30)
10	工艺键编程。 (>>> 6.6 “工艺键编程” 页码 38)
11	脚本编程。 (>>> 6.7 “脚本编程” 页码 45)
12	将工艺整合入操作界面中。 (>>> 6.8 “工艺集成于 KUKA smarthMI” 页码 49)

6.4.1 工艺设计

- 规定联机表单的结构，例如数量、形式和输入栏布局。
- 为每个联机表单规定所需的参数和相应的数值范围，必要时，规定一个要用联机表调用参数列表。



每个联机表单只允许一个参数列表。

- 设定添加的程序行 (= 折叠夹 Fold)。
- 设定工艺键，例如功能、激活操作和布局。
- 脚本设定调用时的用户操作。

例如下列操作：

- 打开或关闭联机表单。
- 按下 **Touch Up**。
- 按下、松开或按住工艺键。
- 显示工艺键。

6.4.2 创建 KFD 文件

说明

通过 UserTech 创建的工艺将在 KFD 文件中进行描述，并默认保存在目录 C:\KRC\TP\USERTECH\TEMPLATE 中。

- 可使用任意编辑器来编辑此 KFD 文件。
- KFD 文件必须以 ASCII 格式保存。
- KFD 文件名称可按照 Windows 操作系统的习惯自由选择。
- KFD 文件可包含一项或多项工艺。在系统中工艺名称仅可出现一次。

对于复杂工艺来说，建议将属于一项工艺的 KFD 文件按结构保存在目录 C:\KRC\TP\TPName\TEMPLATE 中。

针对文件夹 TPName 来说，必须在 Windows 操作系统注册表数据库中创建密匙。

- 操作步骤
1. 打开注册表编辑器。
 2. 选择路径
 \HKEY_Local_Machine\Software\KUKARoboterGmbH\Options\KFD。
 3. 新建字符序列：
 点击鼠标右键并选择菜单序列 **新建 > 字符序列**。
 4. 作为字符序列名称指定文件夹 TPName 的名称。
 5. 将光标置于字符序列元素上，点击鼠标右键并选择菜单项 **更改**：将会打开**编辑字符序列**窗口。
 6. 对于输入新密匙的数值：
 %InstallationDir%\TP\TPName\Template
 7. 点击 **OK** 确认密匙。

6.4.3 DEFTP ... ENDTP

说明

定义工艺

句法

```
DEFTP Name <= {<SOC Bool,> <SOT Bool>}>
ENDTP
```

句法说明

元素	说明
DEFTP	工艺名称
SOC	设定一项工艺的联机表单是否可在联机表单中通过列表栏选择。 <ul style="list-style-type: none">■ TRUE：可选择。■ FALSE：不可选择。 若未定义 SOC，则默认为： TRUE
SOT	设定工艺是否可在联机表单中通过列表栏选择。 <ul style="list-style-type: none">■ TRUE：可选择。■ FALSE：不可选择。 若未定义 SOT，则默认为： TRUE

示例 1

```
DEFTP MyTech
ENDTP
DEFTP OtherTech
ENDTP
```

工艺 **MyTech** 和 **OtherTech** 可通过列表栏在联机表单中进行选择。

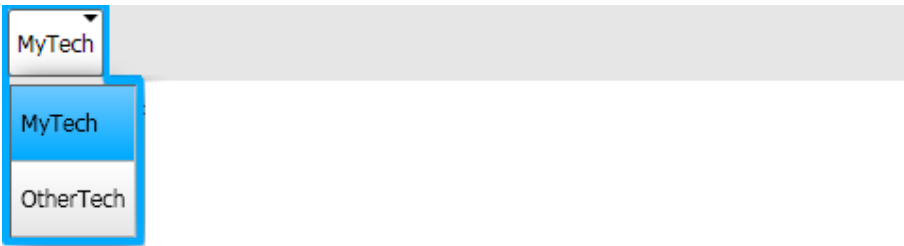


图 6-1：选择工艺的列表栏

示例 2

```
DEFTP MyTech = {SOT FALSE}
ENDTP
DEFTP OtherTech
ENDTP
```

工艺 **MyTech** 和 **OtherTech** 不可通过列表栏在联机表单中进行选择。

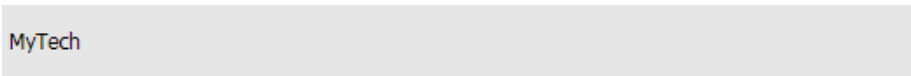


图 6-2：MyTech 工艺

6.4.4 DECL PARAM

说明

联机表单中的字段说明

句法

```
DECL PARAM Name =
{<SHORTNAME[] "String",>
<UNIT[] "String",>
<ENABLE[] Bool,>
VALUE { 字段类型 }}
```

句法说明

元素	说明
DECL PARAM	字段的名称
SHORTNAME[]	字段之前显示的名称 名称最长允许 16 个字符。
UNIT[]	字段之后显示的单位
ENABLE[]	激活输入栏 TRUE : 输入栏激活。 FALSE : 输入栏取消激活。 若未定义 ENABLE[]，则默认为: TRUE
VALUE	字段类型 (>>> 6.4.5 “字段类型” 页码 24)

示例

```
DECL PARAM FieldNumber = {SHORTNAME[] "Distance: ", UNIT[] "mm", _
VALUE {NUMBER: MIN 0, MAX 100, STEP 2, DEFAULT 50}}
```

可输入从 0 到 100 的整数值。该字段的类型为数值。



图 6-3：数字字段类型示例

通过键盘上的光标键可加减输入栏中的数值，步距为 2。

6.4.5 字段类型

6.4.5.1 静态字段类型

说明 联机表单上将显示静态文本，该文本无法编辑。

句法 VALUE {STATIC: DEFAULT[] "String"}

句法说明

元素	说明
DEFAULT[]	联机表单上显示的文本

示例

```
DECL PARAM FieldStatic = _
{VALUE {STATIC: DEFAULT[] "Field cannot be edited"}}
```

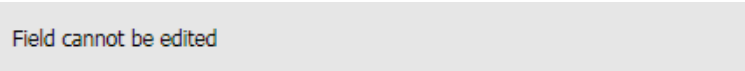


图 6-4：静态字段类型示例

6.4.5.2 自由字段类型

说明 输入栏中将显示可编辑的文本。

句法 VALUE {FREE: DEFAULT[] "String"}

句法说明

元素	说明
DEFAULT[]	首次打开联机表单时在输入栏中默认显示的文字 如果没有定义字符串，则该输入栏为空。


示例

```
DECL PARAM FieldFree = {SHORTNAME[] "Programmer: ", _
VALUE {FREE: DEFAULT[] " Name "}}
```




图 6-5：任意字段类型示例

6.4.5.3 字段类型为名称

 参数列表仅可与字段类型为名称的输入栏相关联。

说明 输入栏中将显示可编辑的变量名称、功能名称或子程序名称。将对名称的句法进行检查。

 不可使用针对 KRL 备用的关键词。其他关于关键词的信息请见系统集成商操作及编程指南。

句法 VALUE {Name: DEFAULT[] "Name"}

句法说明

元素	说明
DEFAULT[]	<p>变量、功能或子程序、首次打开联机表单时输入栏中默认显示的名称</p> <p>如果名称的最后一个字符是数字（0…9），则联机表单时将默认数字加 1。如果联机表单打开之后中断，则计数器不会重置。</p> <p>如果不希望数字增加，则必须将密匙 IncOnOpen 在注册表数据库中类型设为 DWORD。</p> <p>注册表数据库中的路径： HKEY_LOCAL_MACHINE\SOFTWARE\KUKA Roboter GmbH\Options\KFD</p> <p>IncOnOpen:</p> <ul style="list-style-type: none"> ■ DWORD=00000000: 数字不会增加。 ■ DWORD=00000001: 数字将会增加。

示例

```
DECL PARAM FieldName = {SHORTNAME[] "Welding-Point-Nr.: ", _
VALUE {NAME: DEFAULT[] "WPT1"}}
```

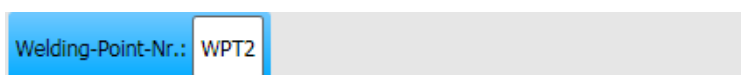


图 6-6: 名称字段类型示例

在注册表数据库中不要新建密匙 **IncOnOpen**。首次打开联机表单时默认显示“WPT1”值。如果下一次创建联机表单，数字将会增加一位，显示“WPT2”值。

6.4.5.4 字段类型为数字

说明

在输入栏中将显示一个整数值，可通过键盘上的数字键区进行编辑。

句法

```
VALUE {NUMBER: MIN 最小值, MAX 最大值,
<STEP 步幅,>
<DEFAULT 默认值,>
<AUTOLIMIT Bool>}
```

句法说明

元素	说明
MIN	<p>类型: INT</p> <p>最小输入值</p> <p>提示: 必须始终给出该值。</p>
MAX	<p>类型: INT</p> <p>最大输入值</p> <p>提示: 必须始终给出该值。</p>
STEP	<p>类型: INT</p> <p>键盘上的光标键步幅，可升高并降低输入栏中的数值</p> <p>若未定义 STEP，则默认为: 1</p>

元素	说明
DEFAULT	类型： INT 首次打开联机表单时默认显示的数值 若未定义 DEFAULT，则默认为： 0
AUTOLIMIT	TRUE ： 过小或过大的数值将会自动转换为最小输入值或最大输入值。（= 若未定义 AUTOLIMIT 时的默认值） <ul style="list-style-type: none">■ 所输入的数值 < 最小值： 数值将转换为最小值。■ 所输入的数值 > 最大值： 数值将转换为最大值。

示例

```
DECL PARAM FieldNumber = {SHORTNAME[] "Distance: ", UNIT[] "mm", _  
VALUE {NUMBER: MIN 0, MAX 100, STEP 2, DEFAULT 50}}
```

可输入从 0 到 100 的整数值。

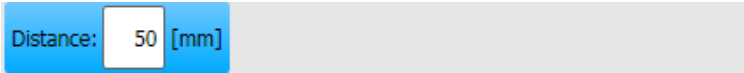


图 6-7：数字字段类型示例

通过键盘上的光标键可加减输入栏中的数值，步距为 2。

6.4.5.5 字段类型 REAL

说明 在输入栏中将显示一个浮点数，可通过键盘上的数字键区进行编辑。

句法
VALUE {REAL: MIN 最小值 , MAX 最大值 ,
<STEP 步幅 ,>
<DEFAULT 默认值 ,>
<AUTOLIMIT Bool>}

句法说明

元素	说明
MIN	类型： REAL 最小输入值 提示： 必须始终给出该值。
MAX	类型： REAL 最大输入值 提示： 必须始终给出该值。
STEP	类型： REAL 键盘上的光标键步幅，可升高并降低输入栏中的数值 若未定义 STEP，则默认为： 0.1

元素	说明
DEFAULT	类型: REAL 首次打开联机表单时默认显示的数值 若未定义 DEFAULT, 则默认为: 0.0
AUTOLIMIT	TRUE : 过小或过大的数值将会自动转换为最小输入值或最大输入值。 (= 若未定义 AUTOLIMIT 时的默认值) <ul style="list-style-type: none"> 所输入的数值 < 最小值: 数值将转换为最小值。 所输入的数值 > 最大值: 数值将转换为最大值。

示例

```
DECL PARAM FieldReal = {SHORTNAME[] "Delay: ", UNIT[] "sec", _
VALUE {REAL: MIN 0.5, MAX 5, STEP 0.5, DEFAULT 2}}
```

可输入从 0.5 到 5 的值。

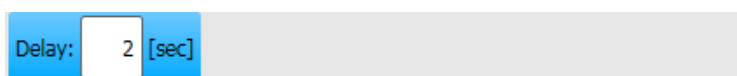


图 6-8: 字段类型 REAL 示例

通过键盘上的光标键可加减输入栏中的数值, 步距为 0.5。

6.4.5.6 字段类型为列表

说明 在联机表单中将显示列表栏。 可通过该列表选择参数。 列表中的参数无法编辑。

句法

```
VALUE {LIST: <DEFAULT[] "String",>|<POS 编号,>
ITEM[1] 列表参数 1,
<ITEM[2] 列表参数 2,>
< ... ,>
<ITEM[n] 列表参数 N>}
```

句法说明

元素	说明
DEFAULT[]	首次打开联机表单时默认显示的列表参数值 该参数必须相应于定义的列表参数之一。(VALUE[] 或 DISP[])
POS	类型: INT 首次打开联机表单时默认显示的列表参数编号 [n]
ITEM[1] ... ITEM[n]	列表参数值 [1] 至 [n] 列表参数的全部句法: {ITEM: VALUE[] "StringX" <, DISP[] "StringN">} <ul style="list-style-type: none"> VALUE[]: 该值将被改变。 DISP[]: 该值将被显示。 当未定义 DISP[] 时的默认值: 所显示数值相应于更改的数值。

示例 1

```
DECL PARAM FieldList = {SHORTNAME[] "Weave pattern:", _
VALUE {LIST: POS 1, ITEM[1] {ITEM: VALUE[] "Triangle"}, _
```

```
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
```

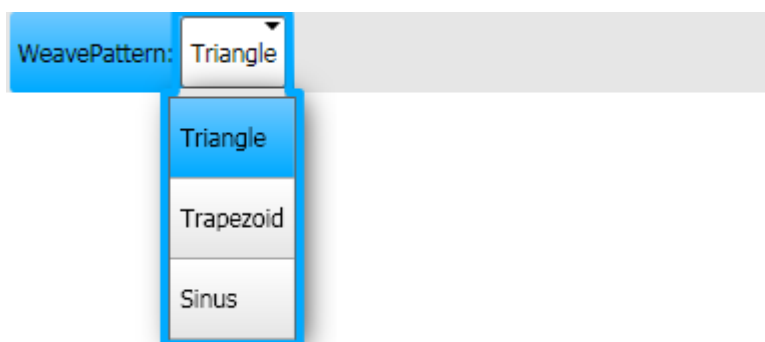


图 6-9：列表字段类型示例

示例 2

```
DEFTP MyTech = {SOC TRUE}
DECL PARAM FieldList = {SHORTNAME[] "Weave pattern:", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "3", DISP[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL FOLD MyFold[1]
MyFold[1] = "Model = %FieldList"
DECL INLINEFORM MyCmd = {FOLD[1] MyFold, PARAM[1] FieldList}
ENDTP
```

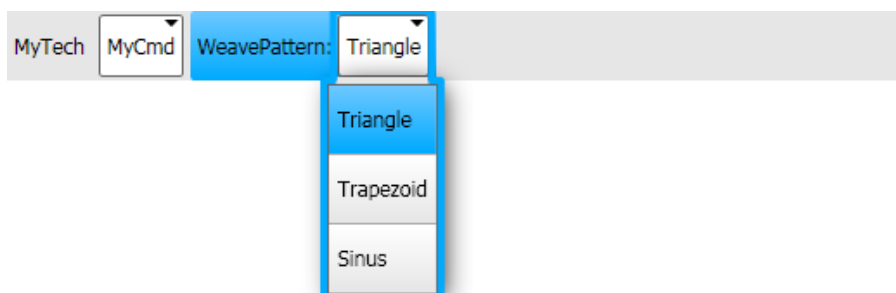


图 6-10：列表字段类型示例—联机表单 MyTech.MyCmd

在列表栏 **WeavePattern** 中所选数值 **Triangle** 与所处理数值不相符。在折叠夹（Fold）中将全局变量 **Model** 赋值为 3。

程序中的 KRL 代码：

```
5 MyTech.MyCmd WeavePattern: Triangle
```

```
6 Model = 3
```

6.4.5.7 字段类型 BOOL

说明

在联机表单中将显示一个复选框，可将其激活也可取消激活。

句法

```
VALUE {BOOL: DEFAULT Bool,
<TRUE[] "String",>
<FALSE[] "String">}
```

句法说明

元素	说明
DEFAULT	首次打开联机表单时默认显示的复选框值 <ul style="list-style-type: none"> ■ TRUE: 复选框激活。 ■ FALSE: 复选框未激活。
TRUE[]	当复选框激活时, 作为 KRL 代码添加到程序中的文本。 如果未定义 TRUE[] 且复选框激活, 则默认将 “TRUE” 作为 KRL 代码添加到程序中。 提示: 类型为 BOOL 的字段中适用构成字符串的附加规则。(>>>> “字符串规则” 页码 29)
FALSE[]	当复选框未激活时, 作为 KRL 代码添加到程序中的文本。 如果未定义 FALSE[] 且复选框激活, 则默认将 “FALSE” 作为 KRL 代码添加到程序中。 提示: 类型为 BOOL 的字段中适用构成字符串的附加规则。(>>>> “字符串规则” 页码 29)

字符串规则

字符串最长允许 16 个字符。

字符串中不允许下列书写方式或字符:

- "" (空字符串)
- " " (仅空格字符)
- "Las,on" (逗号)
- "Las"on" (引号)

示例

```

DEFTP MyTech
DECL PARAM ParamList = {SHORTNAME[] "Val: ", _
    VALUE {Name: DEFAULT[] "VAL1"}}
...
DECL PARAM FieldBool = {SHORTNAME[] "Laser", _
    VALUE {BOOL: DEFAULT TRUE, TRUE[] "EIN", FALSE[] "AUS"}}
...
DECL FOLD PFold[1]
    PFold[1] = "ParamList"
...
DECL INLINEFORM ILF_With_ParamList={FOLD[1] PFold, _
    PARAM[1] ParamList, PARAM[2] FieldBool
ENDTP
    
```



图 6-11: 字段类型 BOOL 示例 (复选框激活)

程序中的 KRL 代码 (复选框激活):

5 [MyTech.ILF_With_ParamList Val: val2, Laser EIN



图 6-12: 字段类型 BOOL 示例 (复选框未激活)

程序中的 KRL 代码 (复选框未激活):

5 [MyTech.ILF_With_ParamList Val: val2, Laser AUS

6.4.6 DECL FOLD

说明

折叠夹（Fold）的说明

折叠夹（Fold）使程序更加一目了然。在折叠夹（Fold）中将隐藏程序段。隐藏的程序段在程序运行时将会象可见程序段一样处理进程。

句法

DECL FOLD *NAME* [*n*]

NAME [*1*] = "*String1*",

...
NAME [*n*] = "*StringN*"

句法说明

元素	说明
<i>NAME</i>	折叠夹（Fold）的名称
[<i>n</i>]	折叠夹（Fold）中的行数
<i>String1</i> ... <i>StringN</i>	Fold 内容 字符串的所有参数引用均在程序中由联机表单中所选数值进行替换。

示例

DECL FOLD MyFold[2]
MyFold[1] = "Laser(%INLINEFORM/,%DataSet/,%Pattern/)"
MyFold[2] = "TRIGGER WHEN PATH=%DistanceWay/ DELAY=0 _
 DO LASER_ON=TRUE"


在联机表单中将选择下列数值：

- DataSet = DataSet6
- DistanceWay = 210
- Pattern = STEP
- INLINEFORM = ON


程序中的 KRL 代码：

Laser(#ON,DataSet6,STEP)
TRIGGER WHEN PATH=210 DELAY=0 DO LASER_ON=TRUE

6.4.7 生成 KRL 代码



关闭联机表单时也应添加在选定程序中的子程序和功能必须进行全局定义。



控制结构（即分支和循环）可以不作为 KRL 代码添加到选定程序中。

- 操作步骤
1. 在目录 KRC:\R1\TP 中创建程序。

2. 在程序中通过传递参数定义所需子程序和功能。

6.4.8 DECL INLINEFORM

说明

联机表单的说明

句法

DECL INLINEFORM *Name* =

{<FOCUS *Position*,>

<FOLD[1] *NameFold1*,> <...,> <FOLD[*n*] *NameFoldN*,>

```

<PARAM[1] NameParam1,> <...,> <PARAM[n] NameParamN,>

<SPLINE Bool,>

<ILF_TYPE IN_SPLINE|OFF_SPLINE|ALL,>

<STYLE WYSIWYG|SUB|DSUB|FCT|DFCT|ASS|ASSAGG,>

<ONACCEPT 脚本,>

<ONTOUCHUP 脚本,>

<ONOPEN 脚本>}

```

句法说明

元素	说明
DECL INLINEFORM	联机表单的名称
FOCUS	<p>类型： INT</p> <p>输入字段的编号，首次打开联机表单时显示在焦点中的默认数字</p> <p>此功能的前提条件：</p> <ul style="list-style-type: none"> 通过指令菜单调用了联机表单。 该联机表单中所声明的工艺定义如下： DEFTP 名称 = {SOC TRUE, SOT TRUE} ENDTP
FOLD[1] ... FOLD[n]	<p>联机表单所属文件夹的名称</p> <p>(>>> 6.4.6 “DECL FOLD” 页码 30)</p> <p>如果分配任何文件夹给该联机表格未，则将根据规定在程序的 STYLE 中添加其内容。</p>
PARAM[1] ... PARAM[n]	<p>联机表单中显示的字段名称</p> <p>(>>> 6.4.4 “DECL PARAM” 页码 23)</p>
SPLINE	<p>规定工艺程序包的联机表单是否作为含有 SPLINE 控制结构的联机表单解读。创建文件夹时将定义该联机表单是否真的含有 SPLINE 控制结构。</p> <ul style="list-style-type: none"> TRUE： 作为含有 SPLINE 控制结构的联机表单解读 FALSE： 作为不含 SPLINE 控制结构的联机表单解读 <p>若在 SPLINE 处输入了无效数值或未输入数值，则默认为：FALSE</p>
ILF_TYPE	<p>规定该联机表单的文件夹是否可添加到一个含有 SPLINE 控制结构的文件夹中。</p> <ul style="list-style-type: none"> IN_SPLINE： 文件夹可添加到 SPLINE 文件夹中。 OFF_SPLINE： 文件夹不可添加到 SPLINE 文件夹中。 ALL： 文件夹可添加到任何文件夹中。 <p>若在 ILF_TYPE 处输入了无效数值或未输入数值，则默认为：OFF_SPLINE</p>

元素	说明
STYLE	KRL 代码的格式化 <div><div>■ WYSIWYG</div><div>■ SUB</div><div>■ DSUB</div><div>■ FCT</div><div>■ DFCT</div><div>■ ASS</div><div>■ ASSAGG</div></div> 若未定义 STYLE，则默认为：WYSIWYG (>>> 6.4.9 “格式类型” 页码 32)
ONACCEPT	按下指令 OK 时所执行脚本的名称 (>>> 8.3.1 “打开和关闭联机表单之后的脚本调用” 页码 60)
ONTOUCHUP	按下 Touch Up 时所执行脚本的名称
ONOPEN	打开联机表单时所执行脚本的名称（按下更改或通过指令菜单进行选择） (>>> 8.3.1 “打开和关闭联机表单之后的脚本调用” 页码 60)

6.4.9 格式类型

6.4.9.1 格式类型 WYSIWYG

说明 KRL 代码的格式化完全符合联机表单的文本。

示例

```
DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist}
DECL INLINEFORM Off
ENDTP
```

Laser

[.]

On

WeldDataSet: DataSet1

Distance: 200 [mm]

WeavePattern: Triangle

图 6-13: 联机表单 Laser.On 示例 1

程序中的 KRL 代码:

```
4 Laser.On WeldDataSet: DataSet0, Distance: 200mm,
  ↳ WeavePattern: Triangle
```


6.4.9.2 格式类型 SUB

说明

KRL 代码的格式化相当于调用子程序：

- 参数减至字段内容，以逗号分隔，并用括号括起来。
- 工艺和联机表单名称之间的分隔点被禁用。

示例

```

DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist, STYLE SUB}
DECL INLINEFORM Off
ENDTP

```

The screenshot shows a control panel with a 'Laser' button (highlighted in blue), an 'On' button, and a dropdown menu for 'WeavePattern' set to 'Triangle'. The 'Distance' is set to 200 mm. The interface also shows 'WeldDataSet: DataSet0'.

图 6-14: 联机表单 Laser.On 示例 2

程序中的 KRL 代码：

4 LaserOn(DataSet0,200,Triangle)

6.4.9.3 格式类型 DSUB

说明

KRL 代码的格式化相当于调用子程序：

- 参数减至字段内容，以逗号分隔，并用括号括起来。
- 工艺和联机表单名称之间的分隔点被禁用。
- 参数名称和单位均作为注释添加。

示例

```

DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist, STYLE DSUB}
DECL INLINEFORM Off
ENDTP

```

The screenshot shows a control panel with a 'Laser' button (highlighted in blue), an 'On' button, and a dropdown menu for 'WeavePattern' set to 'Triangle'. The 'Distance' is set to 200 mm. The interface also shows 'WeldDataSet: DataSet0'.

图 6-15: 联机表单 Laser.On 示例 2

程序中的 KRL 代码:

```
4 LaserOn(DataSet0,200,Triangle) ;WeldDataSet:,
   Distance:[mm],WeavePattern:
```

6.4.9.4 格式类型 FCT

说明

KRL 代码的格式化相当于调用功能:

- 第一个参数值将作为功能名称使用, 与其格式无关。
- 下列参数减至字段内容, 以逗号分隔, 并用括号括起来。
- 工艺和联机表单名称之间的分隔点被禁用。
- 联机表单名称和参数表之间应设为等号。

示例

```
DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "3", DISP[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist, STYLE FCT}
DECL INLINEFORM Off
ENDTP
```

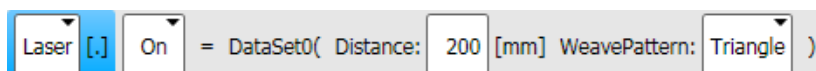


图 6-16: 联机表单 Laser.On 示例 3

程序中的 KRL 代码:

```
4 LaserOn=DataSet0(200,3)
```

6.4.9.5 格式类型 DFCT

说明

KRL 代码的格式化相当于调用功能:

- 第一个参数值将作为功能名称使用, 与其格式无关。
- 下列参数减至字段内容, 以逗号分隔, 并用括号括起来。
- 工艺和联机表单名称之间的分隔点被禁用。
- 联机表单名称和参数表之间应设为等号。
- 参数名称和单位均作为注释添加。

示例

```
DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "3", DISP[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist, STYLE DFCT}
```

```
DECL INLINEFORM Off
ENDTP
```

Laser [.] On = DataSet0(Distance: 200 [mm] WeavePattern: Triangle)

图 6-17: 联机表单 Laser.On 示例 3

程序中的 KRL 代码:

4 LaserOn=DataSet0(200,3) ;Distance:[mm],WeavePattern:

6.4.9.6 格式类型 ASS

说明

KRL 代码的格式化相当于赋值过程:

- 参数减至字段内容并通过逗号分隔。
- 工艺和联机表单名称之间的分隔点被禁用。
- 联机表单名称和参数表之间应设为等号。

示例

```
DEFTP Laser
DECL PARAM Fieldlist = {SHORTNAME[] "WeavePattern: ", _
VALUE {LIST: ITEM[1] {ITEM: VALUE[] "3", DISP[] "Triangle"}, _
ITEM[2] {ITEM: VALUE[] "Trapezoid"}, _
ITEM[3] {ITEM: VALUE[] "Sinus"}}}
DECL PARAM Fieldname = {SHORTNAME[] "WeldDataSet: ", _
VALUE {NAME: DEFAULT[] "DataSet0"}}
DECL PARAM DistanceWay = {SHORTNAME[] "Distance: ", _
VALUE {NUMBER: MIN 0, MAX 500, DEFAULT 200}, Unit[] "mm"}
DECL INLINEFORM On = {PARAM[1] Fieldname, PARAM[2] DistanceWay, _
PARAM[3] Fieldlist, STYLE ASS}
DECL INLINEFORM Off
ENDTP
```

Laser [.] On = WeldDataSet: DataSet0 Distance: 200 [mm]
WeavePattern: Triangle

图 6-18: 联机表单 Laser.On 示例 4

程序中的 KRL 代码:

4 LaserOn=DataSet0,200,3

6.4.9.7 格式类型 ASSAGG

说明

KRL 代码的格式化相当于设备赋值过程:

- 参数将减至参数名称和字段内容。
- 参数将通过逗号分隔,并用大括号扩住。
- 工艺和联机表单名称之间的分隔点被禁用。
- 联机表单名称和参数表之间应设为等号。

示例

```
DEFTP SEARCH
DECL PARAM XDir = {SHORTNAME[] "X=", VALUE {NUMBER: },UNIT[] "mm"}
DECL PARAM YDir = {SHORTNAME[] "Y=", VALUE {NUMBER: },UNIT[] "mm"}
DECL PARAM ZDir = {SHORTNAME[] "Z=", VALUE {NUMBER: },UNIT[] "mm"}
DECL INLINEFORM Direction = {PARAM[1] XDir, PARAM[2] YDir, _
```

PARAM[3] ZDir, STYLE ASSAGG}
ENDTP

SEARCH

[.]


Direction = { X= 1 [mm] Y= 1 [mm] Z= 1 [mm] }

图 6-19: 联机表单 SEARCH.Direction

程序中的 KRL 代码:

```
4 SEARCHDirection={X 1,Y 1,Z 1}
```

6.5 编程参数列表

 每个联机表单只允许一个参数列表。

一个参数列表必须在联机表单中与一个与字段类型为名称的输入栏相关联。参数列表包含一个具有不同参数的数据组。该数据组将显示在一个选项窗口中。数据组的结构和属性均须编程设定。

概览

步骤	说明
1	定义参数表的参数。 (>>> 6.4.4 “DECL PARAM” 页码 23)
2	定义数据组的结构。 (>>> 6.5.1 “DECL PARAM PL_” 页码 36)
3	在 \$CONFIG.DAT 中定义结构数据组和默认数据组。 (>>> 6.5.2 “定义结构数据类型” 页码 37)
4	定义参数表并对应输入栏。 (>>> 6.5.3 “DECL PLIST” 页码 37)

示例 (>>> 8.2 “参数列表的示例程序” 页码 56)

6.5.1 DECL PARAM PL_

说明 数据组结构的说明

```
句法 DECL PARAM PL_Name =  
{<SHORTNAME[] "String",>  
VAR[] "String",  
SHORTCUT[] "String",  
UNIT[] "String",  
VALUE {FREE: DEFAULT[] " _"}}
```

句法说明

元素	说明
DECL PARAM PL_	数据组名称 (>>> 6.5.3 “DECL PLIST” 页码 37)
SHORTNAME[]	选项窗口中所显示的数据组名称
VAR[]	数据组结构数据类型的名称 (>>> 6.5.2 “定义结构数据类型” 页码 37)

元素	说明
SHORTCUT[]	默认数据组的词头 (>>> 6.5.2 “定义结构数据类型” 页码 37)
UNIT[]	数据组的系统名称
VALUE {FREE: DEFAULT[] “_”}	数据组默认值的占位符 (>>> 6.5.2 “定义结构数据类型” 页码 37)

6.5.2 定义结构数据类型

说明 数据组的结构类型和结构数据类型的默认值必须在 C:\KRC\ROBOTER\KRC\R1\SYSTEM\$\CONFIG.DAT 文件 USER GLOBALS 折叠夹 (Fold) 中的 “Userdefined Types” 下进行定义。

句法

结构数据类型:

■ STRUC *Name*
 类型 1 结构名称参数 1,
 ...,
 类型 N 结构名称参数 N

默认值:

■ DECL 名称词头 DEFAULT =
 { 结构名称参数 1 默认 1,
 ...,
 结构名称参数 N 默认 N}

句法说明

元素	说明
名称	结构数据类型名称 (>>> 6.5.1 “DECL PARAM PL_” 页码 36)
类型 1 ... 类型 N	数据组参数的数据类型 ■ INT ■ REAL ■ BOOL
结构名称参数 1 ... 结构名称参数 N	数据组参数的结构名称
前缀	默认数据组的词头 (>>> 6.5.1 “DECL PARAM PL_” 页码 36)
默认 1 ... 默认 N	数据组参数的默认值 默认值可自由选择。

6.5.3 DECL PLIST

说明 参数表的说明

句法 DECL PLIST *Name[n]* -> 输入栏赋值
 Name[1] = *NameParam1*
 ...
 Name[n] = *NameParamN*

句法说明

元素	说明
DECL PLIST	数据组名称 (>>> 6.5.1 “DECL PARAM PL_” 页码 36)
[<i>n</i>]	数据组参数数量
->	数据组所关联的联机表单中的输入栏名称 提示： 参数列表仅可与字段类型为名称的输入栏相关联。
[1] ... [<i>n</i>]	数据组参数顺序 (= 选项窗口中的顺序)
<i>NameParam1</i> ... <i>NameParamN</i>	数据组参数名称 必须用 DECL PARAM 定义参数的布局 and 值域。 (>>> 6.4.4 “DECL PARAM” 页码 23)

6.6 工艺键编程

概览

步骤	说明
1	设计工艺键。 (>>> 6.6.1 “设计工艺键” 页码 38)
2	准备脚本。 (>>> 6.7.1 “DEFSCRIPT ... ENDSCRIPT” 页码 45)
3	定义工艺键。 (>>> 6.6.2 “DECL STATKEY” 页码 38)
4	定义工艺键栏。 (>>> 6.6.3 “DECL STATKEYBAR” 页码 44)
5	脚本编程。 (>>> 6.7 “脚本编程” 页码 45)

6.6.1 设计工艺键

通过 UserTech 可设置 smartPAD 左侧 4 个工艺键的功能。

- 设定工艺键的功能。
- 设定工艺键的布局。
- 设定工艺键的大小。
- 设定激活工艺键的条件。
- 脚本将设定调用脚本的用户操作，例如按下、松开或按住工艺键。

6.6.2 DECL STATKEY

说明

工艺键的说明

句法

```
DECL STATKEY Name =
{<TOPTTEXT[] "String",>
<CENTERTEXT[] "String",>
<BOTTOMTEXT[] "String",>
<PICTURE[] "String",>
<KEYDOWN_PICTURE[] "String",>
```

```

<KEYDOWNMINUS_PICTURE[] "String",>
<ENABLE Bool,>
<NEED_SAFETYSWITCH Bool,>
<NEED_DRIVESOK Bool,>
<NEED_PROSTATEO 激活值,>
<NEED_PROSTATE 激活值,>
<NEED_MODEOP 激活值,>
<USERMODE 标号,>
<STYLE #SWITCH|#TOGGLE,>
<ONKEYDOWN 脚本,>
<ONKEYUP 脚本,>
<ONKEYSHOW 脚本,>
<ONKEYDOWNMINUS 脚本,>
<ONKEYUPMINUS 脚本,>
<ONKEYREPEAT 脚本,>
<ONKEYREPEATMINUS 脚本,>
<NEXT Name>}

```

句法说明

元素	说明
DECL STATKEY	工艺键的名称
...	
NEXT	下一个工艺键的名称

布局:

元素	说明
TOPTEXT[]	工艺键旁显示的文本（上部）
CENTERTEXT[]	工艺键旁显示的文本（居中）
BOTTOMTEXT[]	工艺键旁显示的文本（下部）
PICTURE[]	smartHMI 上显示于工艺键旁的图示的路径说明 ())) 6.6.2.1 "图像大小" 页码 41) ())) 6.6.2.2 "布局 (KSS 8.3 以上)" 页码 41)
KEYDOWN_PICTURE[]	当按下具有 PLUS 功能的按键时, smartHMI 上显示在工艺键旁边的图示的路径说明 ())) 6.6.2.1 "图像大小" 页码 41) ())) 6.6.2.2 "布局 (KSS 8.3 以上)" 页码 41) 如果此处未给出路径, 将在 PICTURE[] 处显示给出的图示。

元素	说明
KEYDOWNMINUS_ PICTURE[]	<p>当按下具有 MINUS 功能的按键时，smartHMI 上显示在工艺键旁边的图示的路径说明</p> <p>(>>> 6.6.2.1 “图像大小” 页码 41)</p> <p>(>>> 6.6.2.2 “布局 (KSS 8.3 以上)” 页码 41)</p> <p>如果此处未给出路径，将在 PICTURE[] 处显示给出的图示。</p>
STYLE	<p>工艺键的格式</p> <ul style="list-style-type: none"> ■ #SWITCH: 简单工艺键 简单工艺键对应 1 键。该键总具有 PLUS 功能。 ■ #TOGGLE: 双重工艺键 双重工艺键对应 2 键。上部按键具有 PLUS 功能。下部按键具有 MINUS 功能。 <p>若未定义 STYLE，则默认为: #SWITCH</p>

激活选项:

元素	说明
ENABLE	<p>工艺键的激活</p> <p>TRUE: 工艺键激活。</p> <p>FALSE: 工艺键取消激活。</p> <p>若未定义 ENABLE，则默认为: TRUE</p>
NEED_SAFETY-SWITCH	TRUE : 当按下确认开关时工艺键将激活。
NEED_DRIVESOK	TRUE : 当驱动接通时工艺键将激活。
NEED_PROSTATEO	<p>类型: INT</p> <p>工艺键的激活情况取决于提交解释器的运行状态。</p> <p>默认值: 32 (= 当提交解释器运行时，工艺键将激活)。</p> <p>(>>> 6.6.2.3 “得出激活选择的十进制数值” 页码 43)</p>
NEED_PROSTATE	<p>类型: INT</p> <p>工艺键的激活情况取决于机器人解释器的运行状态。</p> <p>默认值: 30 (= 当松开启动键时，工艺键激活。)</p> <p>(>>> 6.6.2.3 “得出激活选择的十进制数值” 页码 43)</p>

元素	说明
NEED_MODEOP	<p>类型：INT</p> <p>该工艺键的激活状态取决于运行方式。</p> <p>默认值：7（= 在运行方式 T1、T2 和 AUT 下工艺键激活。）</p> <p>示例：若要在所有运行方式下以及当钥匙开关处于无效位置时激活工艺键，必须针对数值 31 进行编程。</p> <p>(>>> 6.6.2.3 “得出激活选择的十进制数值” 页码 43)</p>
USERMODE	<p>类型：INT</p> <p>用户组的标号，自此用户组开始工艺键激活</p> <ul style="list-style-type: none"> ■ 0：在任何用户组中激活 ■ 10：用户 ■ 20：专家 ■ 30：管理员 <p>若未定义 USERMODE，则默认为：0</p>

调用脚本：

元素	说明
ONKEYDOWN	当按下具有 PLUS 功能的按键时所运行脚本的名称
ONKEYUP	当松开具有 PLUS 功能的按键时所运行脚本的名称
ONKEYSHOW	当通过主菜单激活工艺键时所运行脚本的名称
ONKEYDOWNMINUS	当按下具有 MINUS 功能的按键时所运行脚本的名称
ONKEYUPMINUS	当松开具有 MINUS 功能的按键时所运行脚本的名称
ONKEYREPEAT	<p>当长时间按住具有 PLUS 功能的按键时所运行脚本的名称</p> <p>脚本将重复运行直到松开按键。时间间隔将会不断缩短。</p>
ONKEYREPEATMINUS	<p>当长时间按住具有 MINUS 功能的按键时所运行脚本的名称</p> <p>脚本将重复运行直到松开按键。时间间隔将会不断缩短。</p>

6.6.2.1 图像大小

图示中可使用图标 (*.ICO) 或位图 (*.BMP)。建议使用图标。

需要以下图像大小：

- 32 x 32 像素，用于简单工艺键 (#SWITCH)
- 32 x 96 像素，用于双重工艺键 (#TOGGLE)

6.6.2.2 布局 (KSS 8.3 以上)

说明 系统软件 8.3 版本以上须注意下列布局规则，以便 smartHMI 上工艺键旁的图示能以最优化形式显示出来。

简单工艺键 (#SWITCH):

图示必须居中位于 32 x 32 像素大小的框中。

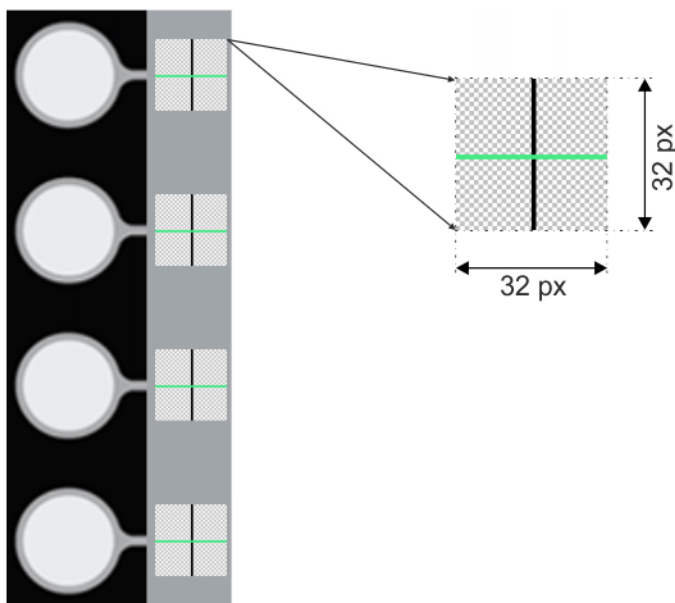


图 6-20: 简单的工艺键—图示居中



图 6-21: 简单的工艺键—布局示例

- 1 图示具有白色背景
- 2 图示具有透明或灰化背景

双重工艺键 (#TOGGLE):

32 x 96 像素的图示将分割为 3 个同样高度的组块部分, 各为 32 x 32 像素。在 smartHMI 上将独立依次显示这些组块部分。要显示在一个组块中的图示必须居中。

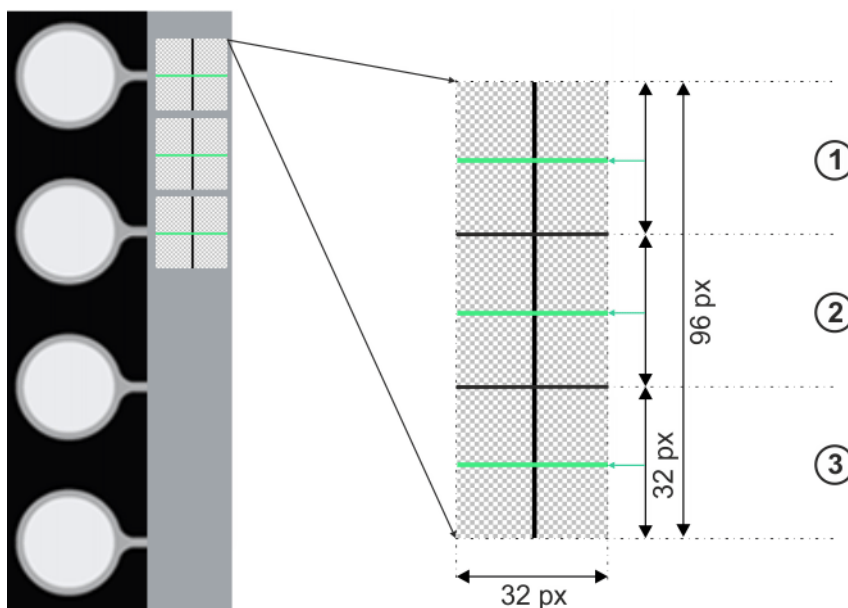


图 6-22: 双重工艺键—图示居中

- 1 图示位于上方
2 图示居中（可选）

- 3 图示位于下方

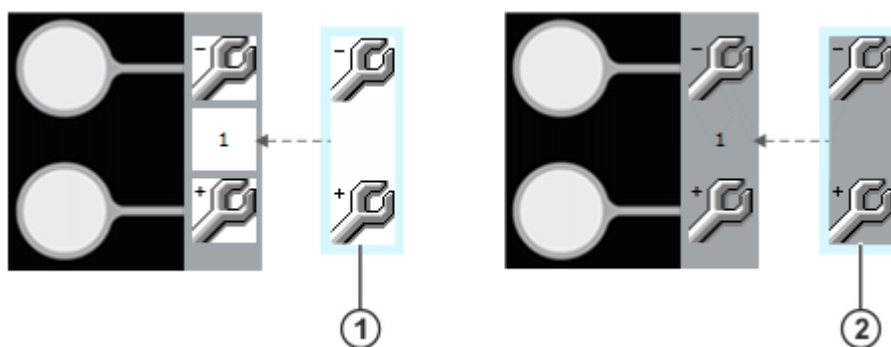


图 6-23：双重工艺键一布局示例

- 1 图示具有白色背景
2 图示具有透明或灰化背景

如果所用图示的背景颜色与 smartHMI 颜色有差异，则可在 smartHMI 上看到。如果仅须看到所用图示，而并非周边区域，则可根据文件类型有下列可能可供选择：

- 图片类型为 *.ICO：围绕图示的区域定义为透明区域。
- 图片类型为 *.BMP：围绕图示的区域中 smartHMI 的颜色灰化：
 - 灰度值 RGB：R:160, G:165, B:170

6.6.2.3 得出激活选择的十进制数值

说明

工艺键可根据提交解释器或机器人解释器的运行状态激活，也可根据运行方式激活：

- NEED_PROSTATE0（提交解释器）
- NEED_PROSTATE（机器人解释器）
- NEED_MODEOP（运行方式）

操作步骤

1. 数值为二进制值 1 的位编号，以基数 2 进行乘方。
2. 乘方叠加。

示例

NEED_PROSTATE0：当提交解释器运行时，工艺键将激活（默认）。

运行方式	位编号	二进制值	乘方	十进制数值
ACTIVE	5	1	$2^5 = 32$	32
END	4	0	0	
RESET	3	0	0	
STOP	2	0	0	
FREE	1	0	0	
UNKNOWN	0	0	0	

NEED_PROSTATE：当松开启动键时，工艺键激活（默认）。

运行方式	位编号	二进制值	乘方	十进制数值
ACTIVE	5	0	0	30
END	4	1	$2^4 = 16$	
RESET	3	1	$2^3 = 8$	
STOP	2	1	$2^2 = 4$	
FREE	1	1	$2^1 = 2$	
UNKNOWN	0	0	0	

NEED_MODEOP: 工艺键在 T1、T2 和 AUT 模式下激活（默认）。

运行方式	位编号	二进制值	乘方	十进制数值
INVALID	4	0	0	7
AUT EXT（外部自动运行）	3	0	0	
AUT（自动）	2	1	$2^2 = 4$	
T2	1	1	$2^1 = 2$	
T1	0	1	$2^0 = 1$	

NEED_MODEOP: 工艺键将在 T1、T2、AUT 和 AUT EXT 模式下以及当钥匙开关处于无效位置处时均会激活。

运行方式	位编号	二进制值	乘方	十进制数值
INVALID	4	1	$2^4 = 16$	31
AUT EXT（外部自动运行）	3	1	$2^3 = 8$	
AUT（自动）	2	1	$2^2 = 4$	
T2	1	1	$2^1 = 2$	
T1	0	1	$2^0 = 1$	

6.6.3 DECL STATKEYBAR

说明

工艺键栏的说明

句法

DECL STATKEYBAR *Name* =

{<STATKEY[1] *Name1*,>
<STATKEY[2] *Name2*,>
<STATKEY[3] *Name3*,>
<STATKEY[4] *Name4*> }

元素	说明
DECL STATKEYBAR	工艺键栏的名称
STATKEY[1] ... STATKEY[4]	工艺键的名称 <ul style="list-style-type: none">■ [1]: 从上数第一个工艺键■ ...■ [4]: 从上数最后一个工艺键

6.6.4 SET

说明 设置工艺键或工艺键栏

SET 在脚本内外均可使用。脚本内容中允许使用运行脚本之前已进行赋值的占位符，例如 %INLINEFORM。

句法 SET *Name* = ...

句法说明

元素	说明
SET	<ul style="list-style-type: none"> ■ 工艺键的名称 其他句法与 “DECL STATKEY <i>Name</i> =” 其后的句法相同。 (>>> 6.6.2 “DECL STATKEY” 页码 38) ■ 工艺键栏的名称 其他句法与 “DECL TATKEYBAR <i>Name</i> =” 其后的句法相同。 (>>> 6.6.3 “DECL STATKEYBAR” 页码 44)

6.7 脚本编程

6.7.1 DEFSCRIPT ... ENDSCRIPT

说明 脚本定义

脚本可与下列操作相关联：

- 打开或关闭联机表单。
- 更改或新建联机表单。
- 在工艺联机表单之间进行切换。
- 按下 **Touch Up**。
- 按下、松开或按住工艺键。
- 显示工艺键。



脚本不可嵌套。

句法 DEFSCRIPT *Name*

指令

ENDSCRIPT

句法说明

元素	说明
DEFSCRIPT	脚本的名称

6.7.2 SETVAR

说明 基本系统中的变量设置

句法 SETVAR (FULLPATH[] "*String*", VALUE[] "*String*")

句法说明

元素	说明
FULLPATH	<p>变量名称</p> <p>字符串中的参数引用将通过当前数值替换。 该字符串将作为变量名称写入路径说明中。</p>
VALUE	<p>变量的值</p> <p>字符串中的参数引用将通过当前数值替换。 该字符串将作为目标值解读。</p>

示例

```
DEFTP MyTech
  DEFSCRIPT Example
    SETVAR(FULLPATH[] "%OUT[20]", VALUE[] "TRUE")
  ENDSRIPT
ENDTFP
```

运行脚本之后：

- `$OUT[20]==TRUE`

6.7.3 SHOWVAR

说明

从基本系统中读取变量

句法

SHOWVAR (FULLPATH [] "String", PARAM *Name*)

句法说明

元素	说明
FULLPATH	变量名称 字符串中的参数引用将通过当前数值替换。该字符串将作为变量名称写入路径说明中。
PARAM	写入所读取数值的参数的名称

示例

```
DEFTP MyTech
  DECL PARAM MyParam = {VALUE {FREE: DEFAULT[] ""}}
  DEFSCRIPT Example
    SHOWVAR(FULLPATH[] "MODE_OP", PARAM MyParam)
    MESSAGE "$MODE_OP=%MyParam"
  ENDSRIPT
ENDTP
```

在运行方式 T1 下的选定程序处运行脚本之后将在信息窗口中给出下列信息:

- \$MODE OP=#T1

6.7.4 REDECL

说明

在基本系统中创建或覆盖一个变量



通过 REDECL 无法创建全局变量。

句法

```
REDECL (PATH[] "String", DECLARATION[] "String")
```

句法说明

元素	说明
路径 []	新变量的路径名称 字符串中的参数引用将通过当前数值替换。该字符串将作为变量名称写入路径说明中。
声明 []	新变量的声明 字符串中的参数引用将通过当前数值替换。该字符串将作为目标值解读。 提示： 全局变量无法声明。不允许关键词 GLOBAL（全局）。

示例

```

DEFTP MyTech
  DEFSCRIPT Example
    REDECL(PATH[] "%MODULE/", DECLARATION[] "DECL AXIS HOME= _
        {A1 09, A2-90, A3 90, A4 0, A5 0, A6 0}")
  ENDSSCRIPT
ENDTP

```

在选定程序上运行脚本之后将在所属的数据表中创建变量 HOME。

```

DEFDAT TEST
  EXTERNAL DECLARATIONS
  DECL AXIS HOME={A1 5.5, A2 -95.5, A3 95.5, A4 5.5, A5 5.5, A6 5.5}
ENDDAT

```

6.7.5 DO

说明

调用脚本

句法

DO *Name*

句法说明

元素	说明
DO	脚本的名称

6.7.6 MESSAGE

说明

定义显示在信息窗中的单行信息。

句法

MESSAGE "*String*"

句法说明

元素	说明
MESSAGE	将显示的信息 字符串中的参数引用将通过当前数值替换。该字符串将作为语言系统的密匙。 (>>> 6.7.10 “信息和按键的语言系统密匙” 页码 49)

6.7.7 SWITCH ... CASE (ELSE) ... ENDSWITCH

说明

调用多个可能脚本其中之一，取决于变量的查询情况。将选择结果与变量相符的 CASE 行。

当运行所属脚本后，将继续执行 ENDSWITCH 之后的程序。

如果没有结果与变量符合，则将选择 CASE ELSE 行。如果没有或未定义所属脚本，则将不会运行脚本并继续执行 ENDSWITCH 之后的程序。



SWITCH...CASE (ELSE)...ENDSWITCH 组块不可嵌套。

句法

SWITCH “ 查询字符串 ”
CASE “ 结果字符串 1 ” DO 脚本 1
<CASE “ 结果字符串 2 ” DO 脚本 2
...
<CASE ELSE 脚本 N>
ENDSWITCH

句法说明

元素	说明
SWITCH	询问的变量名称 字符串中的参数引用将通过当前数值替换。该字符串将作为变量名称写入路径说明中。 可使用下列关键词： <ul style="list-style-type: none">■ ISCHANGE 将会询问是否已有联机表单通过按下 更改 而被打开（= CASE “TRUE”）或通过菜单 指令 新添加了联机表单（= CASE “FALSE”）。 (>>> 8.3.2 “更改和创建联机表单之后的脚本调用” 页码 60)■ DOALWAYS 将询问是否在 2 个联机表单中进行了切换（= CASE “TRUE”）。 (>>> 8.3.3 “更换联机表单之后的脚本调用” 页码 62)
CASE ... DO	询问的结果；运行脚本的名称 字符串中的参数引用将通过当前数值替换。该字符串将作为变量名称写入路径说明中。
CASE ELSE	当定义的结果中没有适当内容时，所运行脚本的名称

6.7.8 SWITCH DIALOG ... CASE ... ENDSWITCH

说明

可实现程序和用户之间的对话。对话将显示在一个窗口中。用户通过可调用定义脚本的按键来回复。回复内容和所属脚本均在 CASE 行中进行定义。
当运行脚本后，将继续执行 ENDSWITCH 之后的程序。



SWITCH DIALOG...CASE...ENDSWITCH 组块不可嵌套。

句法

SWITCH DIALOG “ 提问字符串 ”
CASE “ 回答字符串 1 ” DO 脚本 1
CASE “ 回答字符串 2 ” DO 脚本 2
<CASE “ 回答字符串 3 ” DO 脚本 3
...
<CASE “ 回答字符串 7 ” DO 脚本 7
ENDSWITCH

句法说明

元素	说明
SWITCH DIALOG	显示的对话信息 字符串中的参数引用将通过当前数值替换。该字符串将作为语言系统的密匙。 (>>> 6.7.10 “信息和按键的语言系统密匙” 页码 49)
CASE	按键上显示的回答 如果未定义字符串，则不显示按键。 字符串中的参数引用将通过当前数值替换。该字符串将作为语言系统的密匙。 (>>> 6.7.10 “信息和按键的语言系统密匙” 页码 49)
DO	当按下按键时所运行脚本的名称

6.7.9 预定义脚本

名称	说明
ACCEPTINLINEFORM	联结表单将关闭，将应用所更改的参数。
CANCELINLINEFORM	联结表单将关闭，不会应用所更改的参数。
END	正在运行的脚本将会中断。
NOTHING	用于脚本的占位符

6.7.10 信息和按键的语言系统密匙

说明

信息和按键标识将通过语言系统密匙保存在当前的国家语言中。为此在目录 C:\KRC\DATA 中有一个 KXR 文件，其名称为 Menubar_TPUSER，用于 KUKA smartHMI 上所选择的语言。

在 KXR 文件中必须有一个工艺名称的模块，其中对该脚本进行了定义。这是访问语言系统的前提条件。

全局性脚本中将会访问模块 KUKATPUSER。针对信息和按键必须在语言系统中为每种语言创建适当的密匙。

如果访问语言系统失败，则将不会显示所选语言的信息或按键，而是显示密匙。作为信息发送方将显示其中定义了所执行脚本的工艺名称。

参数化输出时每个参数均必须写在大括号中，添加在密匙之后。

6.8 工艺集成于 KUKA smartHMI

概览

用户定义的联机表单和工艺键必须在配置文件 SmartHMI.User.Config 中已进行说明。SmartHMI.User.Config 位于目录 C:\KRC\USER 中。

在配置文件中已定义例如菜单元素。在此节点处添加的菜单项将不会被保存。工艺程序包或 KUKA 系统软件升级时该处所进行的更改将会丢失。



建议在软件升级之前保存 SmartHMI.User.Config。

用户定义的联机表单和工艺键必须在 SmartHMI.User.Config 中独立的菜单树内进行定义。

- 联机表单将整合进 smartHMI 的指令菜单中 (CommandMenu)。
- 工艺键将整合到 smartHMI 主菜单配置 > 状态键中 (MainMenu)。

6.8.1 配置文件 SmartHMI.User.Config

smartHMI 的菜单元素是下列元素的子元素:

- `<KukaRoboter.SmartHMI.Menu`
`type="KukaRoboter.SmartHMI.UIFramework.Menu.MenuRootImpl,`
`SmartHMI">...<KukaRoboter.SmartHMI.Menu>`

SmartHMI.User.Con
fig

```
<?xml version="1.0"?>
<configuration>
...
<!--Menu -->
<KukaRoboter.SmartHMI.Menu type= _
"KukaRoboter.SmartHMI.UIFramework.Menu.MenuRootImpl, SmartHMI">
  <MenuRoot SystemName="CommandMenu">
    <MenuFolder DisplayName="MenuBar#CmdUserTech" _
      MergeOrder="1100">
      <MenuFolder DisplayName="MenuBar#CmdMyTech" MergeOrder="100">
        <MenuItem DisplayName="MenuBar#CmdMyCmd" MergeOrder="100">
          <CommandCall Target="LegacyLoader.InvokeModuleProc">
            <Argument Name="PlugInName">INLINEFORM</Argument>
            <Argument Name="ProcID">2011</Argument>
            <Argument Name="ParamList">KUKATPUSER;MyTech; _
              MyCmd</Argument>
          </CommandCall>
        </MenuItem>
      </MenuFolder>
    </MenuFolder>
  </MenuRoot>
  <MenuRoot SystemName="MainMenu">
  ...
  <!-- Demo to implement a UserTech StatusKeyBar-->
  <!--MenuFolder DisplayName="MenuBar#ConfigStatKeys" _
    MergeOrder="210">
    <MenuItem DisplayName="MenuBar#UserTStatKeys">
      <CommandCall Target="LegacyLoader.InvokeModuleProc">
        <Argument Name="PlugInName"> _
          USERSTATKEYBAROCX</Argument>
        <Argument Name="ProcID">11</Argument>
        <Argument Name="ParamList">KUKATPUSER; _
          StatkeyTest.Test</Argument>
      </CommandCall>
    </MenuItem>
  </MenuFolder-->
  </MenuFolder>
  </MenuRoot>
  ...
</KukaRoboter.SmartHMI.Menu>
...
</configuration>
```

元素	说明
<code><MenuRoot SystemName="Name"></code> ... <code></MenuRoot></code>	<p>可通过元素 <code><MenuRoot...></code> 设定添加菜单项的菜单。</p> <ul style="list-style-type: none"> ■ CommandMenu: 菜单项将添加到菜单 指令 中。 ■ MainMenu: 菜单项将添加到主菜单中。
<code><MenuFolder DisplayName="Name" MergeOrder="Position"></code> ... <code></MenuFolder></code>	<p><code><MenuFolder...></code> 是 <code><MenuRoot...></code> 的子元素。通过该元素可将菜单结构进行图示。</p> <p>属性 DisplayName 中含有菜单中菜单项的名称。该名称作为语言系统密钥使用。</p> <p>属性 MergeOrder 将设定菜单中菜单项的位置。</p>
<code><MenuItem DisplayName="Name" MergeOrder="Position"></code> ... <code></MenuItem></code>	<p><code><MenuItem...></code> 是 <code><MenuFolder...></code> 的子元素。通过此元素可添加一个设定了功能的菜单项，例如调用联机表单或显示工艺键。</p> <p>属性 DisplayName 中含有菜单中菜单项的名称。该名称作为语言系统密钥使用。</p> <p>属性 MergeOrder 将设定菜单中菜单项的位置。</p>
<code><CommandCall Target="LegacyLoader.InvokeModuleProc"></code> ... <code></CommandCall></code>	<p><code><CommandCall...></code> 是 <code><MenuItem...></code> 的子元素。通过该元素可调用菜单项的功能。</p>
<code><ArgumentName="Name"></code> 变量 <code></Argument></code>	<p><code><Argument...></code> 是 <code><CommandCall...></code> 的子元素。 <code><CommandCall...></code> 包含 3 个自变量。</p> <ul style="list-style-type: none"> ■ PlugInName: 功能模块的名称 <ul style="list-style-type: none"> ■ INLINEFORM: 联机表单 ■ USERSTATKEYBAROCX: 工艺键栏 ■ ProcID: 功能模块的编号 <ul style="list-style-type: none"> ■ 2011: 联机表单 ■ 11: 工艺键栏 ■ ParamList: 功能模块的参数 ParamList 的全部句法: KUKATPUSER ; TechName.IIfName StatkeybarName <ul style="list-style-type: none"> ■ TechName: 其中定义了功能模块的工艺名称 ■ IIfName: 通过该菜单项调用的联机表单的名称 ■ StatkeybarName: 通过该菜单项调用的工艺键栏的名称

6.8.2 定义菜单元素

首要条件

- Windows 层面

操作步骤

1. 用 XML 编辑器打开 SmartHMI.User.Config。
2. 创建菜单树 **CommandMenu** 并在该处定义菜单元素，以便通过菜单 **指令** 调用联机表单。

```
<MenuRoot SystemName="CommandMenu">
...
</MenuRoot>
```

- 在菜单树中添加所希望的 `<MenuFolder...>` 元素，以创建菜单序列。



若要将一个子菜单集成到已存在的菜单中，则显示名称必须与现有显示名称完全相符。若名称不符，则将会创建一个新的菜单项。

- 在所希望的 `<MenuFolder...>` 元素中添加 `<MenuItem...>` 元素，以定义菜单项调用联机表单。
 - 在元素 `<MenuItem...>` 中添加 `<CommandCall...>` 元素连同所属的自变量。
3. 创建菜单树 **MainMenu** 并在该处定义菜单元素，以通过主菜单 **配置** > **状态键** 显示工艺键。

```
<MenuRoot SystemName="MainMenu">
...
</MenuRoot>
```

- 如果尚未存在，则在元素 `<MenuRoot...>` 中添加下列元素：
 - `<MenuFolder DisplayName="MenuBar#Config">`
 - `<MenuFolder DisplayName="MenuBar#ConfigStatKeys" MergeOrder="210">`
 两个 `<MenuFolder...>` 元素将构成菜单序列 **配置** > **状态键**。
 - 在第二个 `<MenuFolder...>` 元素中添加 `<MenuItem...>` 元素，以定义菜单项调用工艺键。
 - 在元素 `<MenuItem...>` 中添加 `<CommandCall...>` 元素连同所属的自变量。
4. 保存更改并重新启动机器人控制系统。

示例

(>>> 8.4 “集成工艺的示例” 页码 63)

7 存档

目标位置

可保存到以下目标位置：

- smartPAD 或机器人控制系统上的 U 盘
- 网络

注意

必须使用一个无启动功能的 U 盘。
建议使用无启动功能的库卡 U 盘。如果使用其它制造商的 U 盘，则数据可能会丢失。



更多关于存档的信息可在操作及编程指南中找到。

7.1 文件存档

说明

下列工艺相关的目录 / 文件将被存档：

- C:\KRC\USER\SmartHMI.User.Config
- C:\KRC\TP\USERTECH\TEMPLATE (KFD 文件)

首要条件

- 已连接 U 盘。
- 或已配置好用作存档路径的网络路径。

操作步骤

1. 选择主菜单**文件** > **存档** > *目标位置*，然后选择菜单项 **配置** 或 **全部**。
2. 点击**是**确认安全询问。将生成档案。当存档过程结束时，将在信息窗口中显示出来。

7.2 恢复文件

首要条件

- 如果应由 U 盘还原数据：已连接了含有该档案的 U 盘。

操作步骤

1. 选择主菜单**文件** > **恢复** > *目标位置*，然后选择菜单项 **配置** 或 **全部**。
2. 点击**是**确认安全询问。已存档的文件在机器人控制系统里重新恢复。当恢复过程结束时，屏幕出现相关的消息。
3. 如果已从 U 盘完成还原：现在可以拔下 U 盘。
4. 重启机器人控制系统。

8 示例

8.1 LASER 工艺

程序

```

1  DEFTP LASER
2
3  DECL PARAM DataSet = {SHORTNAME[] "WeldDataSet=", _
  VALUE {NAME: DEFAULT[] "DataSet1"}}
4
5  DECL PARAM DistanceWay = {SHORTNAME[] "Distance=", _
  VALUE {NUMBER: DEFAULT 200, MIN 0, MAX 500, STEP 10}, _
  UNIT[] "mm"}
6
7  DECL PARAM DelayTime = {SHORTNAME[] "Delay=", _
  VALUE {REAL:DEFAULT 0.8, MIN 0, MAX 20.3, STEP 0.3}, _
  UNIT[] "ms"}
8
9  DECL PARAM Pattern = {SHORTNAME[] "WeavePattern=", _
  VALUE {LIST: ITEM[1] {ITEM: VALUE[] "PULSE"}, _
  ITEM[2] {ITEM: VALUE[] "STEP"}, ITEM[3] {ITEM: VALUE[] "CONT"}}}
10
11 DECL FOLD LasOn[2]
12     LasOn[1] = "Laser(#ON,%DataSet/,%Pattern/)"
13     LasOn[2] = "TRIGGER WHEN DISTANCE=%DistanceWay/ _
  DELAY=0 DO LASER_ON=TRUE"
14
15 DECL FOLD LasOff[2]
16     LasOff[1] = "Laser(#OFF)"
17     LasOff[2] = "TRIGGER WHEN DISTANCE=0 _
  DELAY=%DelayTime/ DO LASER_ON=FALSE"
18
19 DECL INLINEFORM ON = {PARAM[1] DataSet, PARAM[2] DistanceWay, _
  PARAM[3] Pattern, FOLD[1] LasOn}
20
21 DECL INLINEFORM OFF = {PARAM[1] DelayTime, FOLD[1] LasOff}
22
23 ENFTP

```

说明

行	说明
1	工艺名称
3	焊接数据组输入栏说明 字段类型为名称
5	距离参数输入栏说明 字段类型为数字
7	时间参数输入栏说明 字段类型 REAL
9	说明选择焊接方式的列表栏 字段类型为列表： <ul style="list-style-type: none"> ■ PULSE ■ STEP ■ CONT
11 ... 13	折叠夹 (Fold) LasOn 的说明：接通激光器的触发器
15 ... 17	折叠夹 (Fold) LasOff 的说明：关闭激光器的触发器

行	说明
19	联机表单 LASER.ON 的说明 下列字段可供使用： <ul style="list-style-type: none">■ 用于焊接数据组的输入栏■ 用于距离参数的输入栏■ 选择焊接方式的列表栏 LASER.ON 与折叠夹（Fold）LasOn 相对应。
21	联机表单 LASER.OFF 的说明 下列字段可供使用： <ul style="list-style-type: none">■ 用于时间参数的输入栏 LASER.OFF 与折叠夹（Fold）LasOff 相对应。

联机表格

LASER [.]

OFF

Delay=

0.8 [ms]

图 8-1：联机表单 LASER.OFF

LASER [.]

ON

WeldDataSet=

DataSet1

Distance=

200 [mm]

WeavePattern=

PULSE

STEP

CONT

图 8-2：联机表单 LASER.ON

8.2 参数列表的示例程序

示例程序 MyTech 作为 KFD 文件保存在目录 C:\KRC\TP\USERTECH\TEMPLATE 中。

程序

```
1  DEFTP MyTech
2
3  DECL PARAM ParamList = {SHORTNAME[] "Val: ", _
  VALUE {Name: DEFAULT[] "VAL1"}}
4
5  DECL PARAM LSR_MAX_PWR = {SHORTNAME[] "Maximum Power", _
  VALUE {NUMBER: MIN 0, MAX 20000, STEP 100, AUTOLIMIT TRUE, _
  DEFAULT 2000}, UNIT[] "W"}
6
7  DECL PARAM LSR_MIN_PWR = {SHORTNAME[] "Minimum Power", _
  VALUE {NUMBER: MIN 0, MAX 20000, STEP 100, AUTOLIMIT TRUE, _
  DEFAULT 2000}, UNIT[] "W"}
8
9  DECL PARAM LSR_PRG = {SHORTNAME[] "Laser Program", _
  VALUE {NUMBER: MIN 1, MAX 79, STEP 1, AUTOLIMIT TRUE, _
  DEFAULT 1}}
10
11 DECL PARAM SNSR_PRG = {SHORTNAME[] "Sensor Program", _
  VALUE {NUMBER: MIN 0, MAX 3, STEP 1, AUTOLIMIT TRUE, _
```



```

12      DEFAULT 1})}
13 DECL PARAM GAS_PRESSURE = {SHORTNAME[] "Cutting Gas", _
    VALUE {NUMBER: MIN 0, MAX 8, STEP 1, AUTOLIMIT TRUE, _
    DEFAULT 2}, UNIT[] "bar"}
14
15 DECL PARAM GAS_PRE_FLOW = {SHORTNAME[] "PRE Flow Time", _
    VALUE {NUMBERMIN 0, MAX 5000, STEP 100, AUTOLIMIT TRUE, _
    DEFAULT 0}, UNIT[] "ms"}
16
17 DECL PARAM GAS_POST_FLOW = {SHORTNAME[] "POST Flow Time", _
    VALUE {NUMBER: MIN 0, MAX 5000, STEP 100, AUTOLIMIT TRUE, _
    DEFAULT 0}, UNIT[] "ms"}
18
19 DECL PARAM LSR_DLY = {SHORTNAME[] "Piercing Time", _
    VALUE {NUMBER: MIN 0, MAX 5000, STEP 100, AUTOLIMIT TRUE, _
    DEFAULT 0}, UNIT[] "ms"}
20
21 DECL PARAM PL_LsrTool = {SHORTNAME[] "PLTOOL", _
    VAR[] "LSC_TOOL_TYP", SHORTCUT[] "LT", UNIT[] "TOOL", _
    VALUE {FREE: DEFAULT[] "_"}}
22
23 DECL FOLD MyFold[1]
24     MyFold[1]= ";test"
25
26 DECL FOLD PFold[1]
27     PFold[1]= ";ParamList"
28
29 DECL PLIST LsrTool[8] -> ParamList
30     LsrTool[1] = LSR_MAX_PWR
31     LsrTool[2] = LSR_MIN_PWR
32     LsrTool[3] = LSR_PRG
33     LsrTool[4] = SNSR_PRG
34     LsrTool[5] = GAS_PRESSURE
35     LsrTool[6] = GAS_PRE_FLOW
36     LsrTool[7] = GAS_POST_FLOW
37     LsrTool[8] = LSR_DLY
38
39 DECL INLINEFORM MyCmd = {FOLD[1] MyFold}
40
41 DECL INLINEFORM ILF_With_ParamList = FOLD[1] PFold, _
    PARAM[1] ParamList}
42
43 ENDTP

```

说明

行	说明
1	工艺名称
3	激光数据组的 ParamList 输入栏说明 字段类型为名称 提示： 参数列表仅可与字段类型为名称的输入栏相关联。

行	说明
5 ... 19	激光数据组参数说明 字段类型为数字 <ul style="list-style-type: none">■ LSR_MAX_PWR （最大功率）■ LSR_MIN_PWR （最小功率）■ LSR_PRG （激光器程序）■ SNSR_PRG （传感器程序）■ GAS_PRESSURE (Cutting Gas)■ GAS_PRE_FLOW (PRE Flow Time)■ GAS_POST_FLOW (POST Flow Time)■ LSR_DLY (Piercing Time)
21	数据组结构的说明 结构类型 LSC_TOOL_TYP 和结构数据类型的默认值必须在 \$CONFIG.DAT 文件中进行定义（USER GLOBALS 折叠夹（Fold）中的“Userdefined Types”）。
23 ... 27	折叠夹（Fold）的说明
29	参数列表说明 通过指令 “-> ParamList” 可将参数列表与 ParamList 输入栏相关联。
30 ... 37	参数赋值和数据组参数顺序（= 选项窗口中的顺序）
39, 41	工艺联机表单的说明 <ul style="list-style-type: none">■ MyCmd■ ILF_With_ParamList

\$CONFIG.DAT

结构类型 LSC_TOOL_TYP 和结构数据类型的默认值必须在 \$CONFIG.DAT 文件中进行定义（USER GLOBALS 折叠夹（Fold）中的“Userdefined Types”）。

```
STRUC LSC_TOOL_TYP _  
INT LSR_MAX_PWR,INT LSR_MIN_PWR,INT LSR_Prg, _  
REAL SNSR_Prg,INT SNSR_State,REAL GAS_Pressure, _  
INT LSR_Dly,INT GAS_Pre_Flow,INT GAS_Post_Flow  
DECL LSC_TOOL_TYP LTDEFAULT = _  
{LSR_MAX_PWR 500,LSR_MIN_PWR 50,LSR_Prg 1, _  
SNSR_Prg 1.0,SNSR_State 0,GAS_Pressure 1.0, _  
LSR_Dly 100,GAS_Pre_Flow 0,GAS_Post_Flow 0}
```

联机表单

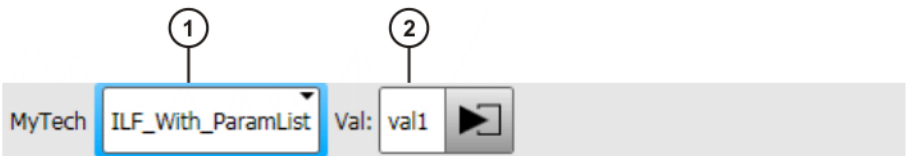


图 8-3：含有参数表的联机表单

项号	说明
1	含有工艺联机表单的列表栏 <ul style="list-style-type: none">■ MyCmd■ ILF_With_ParamList
2	输入栏 ParamList 用于激光数据组 数据组参数可在选项窗口 PLT00L 中进行设置。

参数列表

Maximum Power ① [W]

-

2000

+

■ ■

Minimum Power ② [W]

-

2000

+

■ ■

Laser Program ③

-

1

+

Sensor Program ④

-

1

+

■ ■ ■ ■ ■ ■ ■ ■

Cutting Gas ⑤ [bar]

-

2

+

■ ■ ■ ■ ■ ■

PRE Flow Time ⑥ [ms]

-

0

+

Ⓐ

PLTOOL

Laser Program ③

-

1

+

Sensor Program ④

-

1

+

■ ■ ■ ■ ■ ■ ■ ■

Cutting Gas ⑤ [bar]

-

2

+

■ ■ ■ ■ ■ ■

PRE Flow Time ⑥ [ms]

-

0

+

POST Flow Time ⑦ [ms]

-

0

+

Piercing Time ⑧ [ms]

-

0

+

Ⓑ

PLTOOL

图 8-4: 选项窗口 PLTOOL

- A A 视图

B B 视图

1 LSR_MAX_PWR

2 LSR_MIN_PWR

3 LSR_PRG
- 4 SNSR_PRG

5 GAS_PRESSURE

6 GAS_PRE_FLOW

7 GAS_POST_FLOW

8 LSR_DLY

8.3 通过操作调用脚本

概览

定义联机表单时可给出在下列操作之后调用的脚本：

- 打开或关闭联机表单。
(>>> 8.3.1 “打开和关闭联机表单之后的脚本调用” 页码 60)
- 更改或新建联机表单。
(>>> 8.3.2 “更改和创建联机表单之后的脚本调用” 页码 60)
- 在工艺联机表单之间进行切换。
(>>> 8.3.3 “更换联机表单之后的脚本调用” 页码 62)

8.3.1 打开和关闭联机表单之后的脚本调用

程序

```
1  DEFTP TEST
2
3  ;-----
4  ;----- Scripts -----
5  ;-----
6
7  DEFSCRIPT OP_Test
8    MESSAGE "Fired when ILF is opened!"
9  ENDSRIPT
10
11 DEFSCRIPT CL_Test
12   MESSAGE "Fired when ILF is closed!"
13   DO ACCEPTINLINEFORM
14 ENDSRIPT
15
16 ;-----
17 ;----- FOLds -----
18 ;-----
19
20 DECL FOLD TestFold[1]
21   TestFold[1]="/;only for testing ONOPEN-event"
22
23 ;-----
24 ;----- INLINEFORMs -----
25 ;-----
26
27 DECL INLINEFORM TestILF={FOLD[1] TestFold, ONOPEN OP_Test, _
                           ONACCEPT CL_Test}
28
29 ENDTP
```

说明

行	说明
7 ... 9	脚本 OP_Test 的定义 调用脚本时将在信息窗口中显示信息 “ <i>Fired when ILF is opened!</i> ” 。
11 ... 14	脚本 CL_Test 的定义 调用脚本时将在信息窗口中显示信息 “ <i>Fired when ILF is closed!</i> ” 。
27	联机表单的说明 <ul style="list-style-type: none">■ ONOPEN: 当联机表单打开时，调用脚本的关键词■ ONACCEPT: 当联机表单关闭时，调用脚本的关键词

8.3.2 更改和创建联机表单之后的脚本调用

程序

```
1  DEFTP TEST
2
3  ;-----
4  ;----- Scripts -----
5  ;-----
6
7  DEFSCRIPT CH_Test
8    MESSAGE "Fired only when ILF is changed!"
9  ENDSRIPT
10
```

```

11 DEFSCRIPT NEW_Test
12     MESSAGE "Fired only when ILF is created!"
13 ENDSRIPT
14
15 DEFSCRIPT OP_Test
16     SWITCH "ISCHANGE"
17         CASE "TRUE" DO CH_Test
18         CASE "FALSE" DO MEW_Test
19     ENDSWITCH
20 ENDSRIPT
21
22 DEFSCRIPT CL_Test
23     MESSAGE "Fired when ILF is closed!"
24     DO ACCEPTINLINEFORM
25 ENDSRIPT
26
27 ;-----
28 ;----- FOLDS -----
29 ;-----
30
31 DECL FOLD TestFold[1]
32     TestFold[1]="/;only for testing ONOPEN-event"
33
34 ;-----
35 ;----- INLINEFORMs -----
36 ;-----
37
38 DECL INLINEFORM TestILF={FOLD[1] TestFold, ONOPEN OP_Test, _
                           ONACCEPT CL_Test}
39
40 ENDTP

```

说明

行	说明
7 ... 9	脚本 CL_Test 的定义 调用脚本时将显示信息 “ <i>Fired only when ILF is changed!</i> ”。
11 ... 13	脚本 NEW_Test 的定义 调用脚本时将显示信息 “ <i>Fired only when ILF is created!</i> ”。
15 ... 20	脚本 OP_Test 的定义 调用脚本时将会使用关键词 ISCHANGE 询问是否已有联机表单通过按下 更改 而被打开 (= CASE “TRUE”)或通过菜单 指令 新添加了联机表单 (= CASE “FALSE”)。 <ul style="list-style-type: none"> ■ CASE “TRUE”: 对于已存在的联机表单, 将调用脚本 CH_Test。 ■ CASE “FALSE”: 对于新添加的联机表单, 将调用脚本 NEW_Test。
22 ... 25	脚本 CL_Test 的定义 调用脚本时将在信息窗口中显示信息 “ <i>Fired when ILF is closed!</i> ”。
38	联机表单的说明 <ul style="list-style-type: none"> ■ ONOPEN: 当联机表单打开时, 调用脚本的关键词 ■ ONACCEPT: 当联机表单关闭时, 调用脚本的关键词

8.3.3 更换联机表单之后的脚本调用

程序

```
1  DEFTP TEST
2
3  ;-----
4  ;----- Scripts -----
5  ;-----
6
7  DEFSCRIPT Test_1
8    MESSAGE "Fired only once!"
9  ENDSRIPT
10
11 DEFSCRIPT AL_Test
12   MESSAGE "Fired always!"
13 ENDSRIPT
14
15 DEFSCRIPT Test_2
16   SWITCH "DOALWAYS"
17     CASE "TRUE" DO AL_Test
18   ENDSWITCH
19 ENDSRIPT
20
21 ;-----
22 ;----- FOLDS -----
23 ;-----
24
25 DECL FOLD TestFold[1]
26   TestFold[1]="/;only for testing ONOPEN-event"
27
28 ;-----
29 ;----- INLINEFORMs -----
30 ;-----
31
32 DECL INLINEFORM TestILF_1={FOLD[1] TestFold, ONOPEN test_1}
33 DECL INLINEFORM TestILF_2={FOLD[1] TestFold, ONOPEN test_2}
34
35 ENDTP
```

说明

行	说明
7 ... 9	脚本 Test_1 的定义 调用脚本时将显示信息 “ <i>Fired only once!</i> ” 。
11 ... 13	脚本 AL_Test 的定义 调用脚本时将显示信息 “ <i>Fired always!</i> ” 。
15 ... 19	脚本 Test_2 的定义 调用脚本时将通过密匙 DOALWAYS 查询，是否曾在 2 个联机表单之间进行了切换。 ■ CASE “TRUE”： 当从联机表单 TestILF_1 切换为联机表单 TestILF_2 时，将调用脚本 AL_Test。
32 ... 33	联机表单 TestILF_1 和 TestILF_2 的说明 ■ ONOPEN： 当联机表单打开时，调用脚本的关键词

8.4 集成工艺的示例

8.4.1 TWINKLE 工艺

程序

```

1  DEFTP Twinkle
2
3  DECL FOLD One[5]
4      One[1] = "/;Twinkle once"
5      One[2] = "$OUT[5]=TRUE"
6      One[3] = "WAIT SEC 1.5"
7      One[4] = "$OUT[5]=FALSE"
8      One[5] = "WAIT SEC 1.5"
9
10 DECL FOLD Two[9]
11     Two[1] = "/Twinkle twice"
12     Two[2] = "$OUT[5]=TRUE"
13     Two[3] = "WAIT SEC 1.5"
14     Two[4] = "$OUT[5]=FALSE"
15     Two[5] = "WAIT SEC 1.5"
16     Two[6] = "$OUT[5]=TRUE"
17     Two[7] = "WAIT SEC 1.5"
18     Two[8] = "$OUT[5]=FALSE"
19     Two[9] = "WAIT SEC 1.5"
20
21 DECL INLINEFORM once = {FOLD[1] One}
22 DECL INLINEFORM twice = {FOLD[1] Two}
23 DECL INLINEFORM triple = {FOLD[1] One, FOLD[2] Two}
24
25 ENDTP

```

说明

行	说明
1	工艺名称
2 ... 8	折叠夹 (Fold) One 的说明
10 ... 19	折叠夹 (Fold) Two 的说明
21	联机表单 Twinkle.once 的说明 Twinkle.once 与折叠夹 (Fold) One 相对应。
22	联机表单 Twinkle.twice 的说明 Twinkle.twice 与折叠夹 (Fold) Two 相对应。
23	联机表单 Twinkle.triple 的说明 Twinkle.triple 与折叠夹 (Fold) One 和折叠夹 (Fold) Two 相对应。

8.4.1.1 将 TWINKLE 工艺集成到菜单中

说明

菜单项 **BlinkTech** 将添加到菜单 **指令 > UserTech** 中的第二个位置处，用来调用联机表单 **Twinkle.twice**。

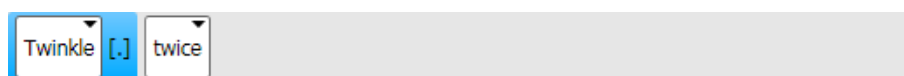


图 8-5: 联机表单 Twinkle.twice

程序

```

<?xml version="1.0"?>
<configuration>
...

```

```

<!--Menu -->
<KukaRoboter.SmartHMI.Menu type= _
"KukaRoboter.SmartHMI.UIFramework.Menu.MenuRootImpl, SmartHMI">
  <MenuRoot SystemName="CommandMenu">
    <MenuFolder DisplayName="MenuBar#CmdUserTech" _
      MergeOrder="1100">
      <MenuFolder DisplayName="MenuBar#CmdMyTech" MergeOrder="100">
        <MenuItem DisplayName="MenuBar#CmdMyCmd" MergeOrder="100">
          <CommandCall Target="LegacyLoader.InvokeModuleProc">
            <Argument Name="PlugInName">INLINEFORM</Argument>
            <Argument Name="ProcID">2011</Argument>
            <Argument Name="ParamList">KUKATPUSER;MyTech; _
              MyCmd</Argument>
          </CommandCall>
        </MenuItem>
      </MenuFolder>
    </MenuFolder>
    <MenuFolder DisplayName="MenuBar#CmdUserTech" _
      MergeOrder="1100">
      <MenuItem DisplayName="BlinkTech" MergeOrder="200">
        <CommandCall Target="LegacyLoader.InvokeModuleProc">
          <Argument Name="PlugInName">INLINEFORM</Argument>
          <Argument Name="ProcID">2011</Argument>
          <Argument Name="ParamList">KUKATPUSER;Twinkle; _
            twice</Argument>
        </CommandCall>
      </MenuItem>
    </MenuFolder>
  </MenuRoot>
  ...
</KukaRoboter.SmartHMI.Menu>
  ...
</configuration>

```

8.4.1.2 将 TWINKLE 工艺集成到子菜单中

说明

菜单项 **BlinkTech** 将添加到菜单 **指令 > UserTech** 的第二个位置处。子菜单 **BlinkTech** 中含有菜单项 **One**、**Two** 和 **Three**。

- 菜单项 **One** 可调用联机表单 **Twinkle.once**。
- 菜单项 **Two** 可调用联机表单 **Twinkle.twice**。
- 菜单项 **Three** 可调用联机表单 **Twinkle.triple**。

程序

```

<?xml version="1.0"?>
<configuration>
  ...
  <!--Menu -->
  <KukaRoboter.SmartHMI.Menu type= _
    "KukaRoboter.SmartHMI.UIFramework.Menu.MenuRootImpl, SmartHMI">
    <MenuRoot SystemName="CommandMenu">
      <MenuFolder DisplayName="MenuBar#CmdUserTech" _
        MergeOrder="1100">
      <MenuFolder DisplayName="MenuBar#CmdMyTech" MergeOrder="100">
        <MenuItem DisplayName="MenuBar#CmdMyCmd" MergeOrder="100">
          <CommandCall Target="LegacyLoader.InvokeModuleProc">
            <Argument Name="PlugInName">INLINEFORM</Argument>
            <Argument Name="ProcID">2011</Argument>
            <Argument Name="ParamList">KUKATPUSER;MyTech; _
              MyCmd</Argument>
          </CommandCall>
        </MenuItem>
      </MenuFolder>
    </MenuRoot>
    ...
  </KukaRoboter.SmartHMI.Menu>
  ...
</configuration>

```



```

    </MenuItem>
  </MenuFolder>
  <MenuFolder DisplayName="BlinkTech" MergeOrder="200">
    <MenuItem DisplayName="One" MergeOrder="210">
      <CommandCall Target="LegacyLoader.InvokeModuleProc">
        <Argument Name="PlugInName">INLINEFORM</Argument>
        <Argument Name="ProcID">2011</Argument>
        <Argument Name="ParamList">KUKATPUSER;Twinkle; _
          once</Argument>
      </CommandCall>
    </MenuItem>
    <MenuItem DisplayName="Two" MergeOrder="220">
      <CommandCall Target="LegacyLoader.InvokeModuleProc">
        <Argument Name="PlugInName">INLINEFORM</Argument>
        <Argument Name="ProcID">2011</Argument>
        <Argument Name="ParamList">KUKATPUSER;Twinkle; _
          twice</Argument>
      </CommandCall>
    </MenuItem>
    <MenuItem DisplayName="Three" MergeOrder="230">
      <CommandCall Target="LegacyLoader.InvokeModuleProc">
        <Argument Name="PlugInName">INLINEFORM</Argument>
        <Argument Name="ProcID">2011</Argument>
        <Argument Name="ParamList">KUKATPUSER;Twinkle; _
          triple</Argument>
      </CommandCall>
    </MenuItem>
  </MenuFolder>
</MenuFolder>
</MenuRoot>
...
</KukaRoboter.SmarthMI.Menu>
...
</configuration>

```

8.4.2 GLUE 工艺

程序

```

1  DEFTP Glue
2
3  DECL STATKEY Key4 = {TOPTEXT[] "Rinse", BUTTOMTEXT[] "ON", _
  PICTURE[] "C:\KRC\TP\USERTECH\TEMPLATE\PICTURES\Rinse1.BMP", _
  NEED_PROSTATEO 63}
4
5  DEFSCRIPT ScKeyUp
6  SETVAR (FULLPATH[] "$OUT[2]", VALUE[] "FALSE")
7  SET Key4 = {PICTURE[] "C:\KRC\TP\USERTECH\TEMPLATE\ _
  PICTURES\Rinse1.BMP", BUTTOMTEXT[] "ON"}
8  ENDSCRIPT
9
10 DEFSCRIPT ScKeyDown
11 SETVAR (FULLPATH[] "$OUT[2]", VALUE[] "TRUE")
12 SET Key4 = {PICTURE[] "C:\KRC\TP\USERTECH\TEMPLATE\ _
  PICTURES\Rinse2.BMP", BUTTOMTEXT[] "OFF"}
13 ENDSCRIPT
14
15 SET Key4 = {ONKEYDOWN ScKeyDown, ONKEYUP ScKeyUp}
16
17 DECL STATKEYBAR Bar4
18 SET Bar4 = {STATKEY[4] Key4}

```

19
20 ENDTP

说明

行	说明
1	工艺名称
3	工艺键 Key4 说明 NEED_PROSTATE0 63: 工艺键与提交解释器的运行状态无关, 始终处于激活状态。
5 … 8	脚本 ScKeyUp 的定义
10 … 13	脚本 ScKeyDown 的定义
15	设置工艺键 Key4 <ul style="list-style-type: none">■ 按下工艺键时将运行脚本 ScKeyDown。在工艺键旁将显示文本 ON。 运行脚本之后: \$OUT[2]==FALSE■ 松开工艺键时将运行脚本 ScKeyUp。在工艺键旁将显示文本 OFF 以及另外一个图示。 运行脚本之后: \$OUT[2]==TRUE
17	工艺键栏 Bar4 的说明
18	设置工艺键栏 Bar4 工艺键栏由单个工艺键组成 (= Key4)。工艺键位于第四个位置 (= 从上数最后一个工艺键)。

8. 4. 2. 1 将 GLUE 工艺集成到菜单中

说明

菜单项 **Rinse** 将添加到主菜单 **配置 > 状态键** 的第一个位置处, 用来调用工艺键栏。

程序

```
<?xml version="1.0"?>
<configuration>
...
<!--Menu -->
<KukaRoboter.SmarthMI.Menu type= _
"KukaRoboter.SmarthMI.UIFramework.Menu.MenuRootImpl, SmarthMI">
  <MenuRoot SystemName="CommandMenu">
...
  </MenuRoot>
  <MenuRoot SystemName="MainMenu">
    <MenuFolder DisplayName="MenuBar#Config"
    <MenuFolder DisplayName="MenuBar#ConfigStatKeys" _
      MergeOrder="210">
      <MenuItem DisplayName="MenuBar#Rinse">
        <CommandCall Target="LegacyLoader.InvokeModuleProc">
          <Argument Name="PlugInName"> _
            USERSTATKEYBAROCX</Argument>
          <Argument Name="ProcID">11</Argument>
          <Argument Name="ParamList">KUKATPUSER; _
            Glue.Bar4</Argument>
        </CommandCall>
      </MenuItem>
    </MenuFolder>
  </MenuFolder>
</MenuRoot>
...
</KukaRoboter.SmarthMI.Menu>
```

```
...
</configuration>
```

8.4.3 DISPSET 工艺

程序

```
1  DEFTP DispSet

2  DECL PARAM Bright = {VALUE {NUMBER:}}
3  DECL PARAM Cont = {VALUE {NUMBER:}}
4  DECL PARAM Temp = {VALUE {NUMBER:}}
5
6  DEFSCRIPT BrightHigh
7  SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
8  SETVAR(FULLPATH[] "$PHGBRIGHT", VALUE[] "%Bright +1")
9  SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
10 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
11 SHOWVAR(FULLPATH[] "$PHGTEMP", PARAM Temp)
12 MESSAGE "Brightness Level: %Bright Contrast Level: %Cont _
    Display Temperature: %Temp °C"
13 ENDSCRIPT
14
15 DEFSCRIPT BrightLow
16 SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
17 SETVAR(FULLPATH[] "$PHGBRIGHT", VALUE[] "%Bright -1")
18 SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
19 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
20 SHOWVAR(FULLPATH[] "$PHGTEMP", PARAM Temp)
21 MESSAGE "Brightness Level: %Bright Contrast Level: %Cont _
    Display Temperature: %Temp °C"
22 ENDSCRIPT
23
24 DEFSCRIPT ContHigh
25 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
26 SETVAR(FULLPATH[] "$PHGCONT", VALUE[] "%Cont +1")
27 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
28 SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
29 SHOWVAR(FULLPATH[] "$PHGTEMP", PARAM Temp)
30 MESSAGE "Brightness Level: %Bright Contrast Level: %Cont _
    Display Temperature: %Temp °C"
31 ENDSCRIPT
32
33 DEFSCRIPT ContLow
34 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
35 SETVAR(FULLPATH[] "$PHGCONT", VALUE[] "%Cont -1")
36 SHOWVAR(FULLPATH[] "$PHGCONT", PARAM Cont)
37 SHOWVAR(FULLPATH[] "$PHGBRIGHT", PARAM Bright)
38 SHOWVAR(FULLPATH[] "$PHGTEMP", PARAM Temp)
39 MESSAGE "Brightness Level: %Bright Contrast Level: %Cont _
    Display Temperature: %Temp °C"
40 ENDSCRIPT
41
42 DECL STATKEY Brightness = {PICTURE[] "C:\KRC\TP\USERTECH\ _
    TEMPLATE\PICTURES\PHGBRIGHT.BMP", ENABLE TRUE, USERMODE 10, _
    STYLE #TOGGLE, ONKEYDOWN BrightHigh, ONKEYDOWNMINUS BrightLow}
43
44 DECL STATKEY Contrast = {PICTURE[] "C:\KRC\TP\USERTECH\ _
    TEMPLATE\PICTURES\PHGCONT.BMP", ENABLE TRUE, USERMODE 10, _
    STYLE #TOGGLE, ONKEYDOWN ContHigh, ONKEYDOWNMINUS ContLow}
45
```

```
46  DECL STATKEYBAR Bar = {STATKEY[1] Brightness, _  
    STATKEY[3] Contrast}  
47  
48  ENDTP
```

说明

行	说明
1	工艺名称
2 ... 4	脚本中所使用参数的说明 字段类型为数字
6 ... 13	脚本 BrightHigh 的定义 调用脚本时系统变量 \$PHGBRIGHT 的值将升高 1 位。
15 ... 22	脚本 BrightLow 的定义 调用脚本时系统变量 \$PHGBRIGHT 的值将降低 1 位。
24 ... 31	脚本 ContHigh 的定义 调用脚本时系统变量 \$PHGCONT 的值将升高 1 位。
33 ... 40	脚本 ContLow 的定义 调用脚本时系统变量 \$PHGCONT 的值将降低 1 位。
42	工艺键 Brightness 的说明 使用此双重工艺键可调节亮度。 ■ 按下具有 PLUS 功能的按键将调用脚本 BrightHigh。 ■ 按下具有 MINUS 功能的按键将调用脚本 BrightLow。
44	工艺键 Contrast 的说明 使用此双重工艺键可调节对比度。 ■ 按下具有 PLUS 功能的按键将调用脚本 ContHigh。 ■ 按下具有 MINUS 功能的按键将调用脚本 ContLow。
46	工艺键栏的说明

8.4.3.1 将 DISPSET 工艺集成到菜单中

说明

菜单项 **Display** 将添加到主菜单 **配置 > 状态键** 的第二个位置处，用来调用工艺键栏。

程序

```
<?xml version="1.0"?>  
<configuration>  
...  
<!--Menu -->  
<KukaRoboter.SmarthMI.Menu type= _  
    "KukaRoboter.SmarthMI.UIFramework.Menu.MenuRootImpl, SmarthMI">  
    <MenuRoot SystemName="CommandMenu">  
...  
    </MenuRoot>  
    <MenuRoot SystemName="MainMenu">  
        <MenuFolder DisplayName="MenuBar#Config"  
        <MenuFolder DisplayName="MenuBar#ConfigStatKeys" _  
            MergeOrder="210">  
            <MenuItem DisplayName="MenuBar#Rinse" MergeOrder="100">  
                <CommandCall Target="LegacyLoader.InvokeModuleProc">  
                    <Argument Name="PlugInName"> _  
                        USERSTATKEYBAROCX</Argument>  
                    <Argument Name="ProcID">11</Argument>  
                    <Argument Name="ParamList">KUKATPUSER; _  
                        Glue.Bar4 </Argument>
```

```
</CommandCall>
</MenuItem>
<MenuItem DisplayName="MenuBar#Display" MergeOrder="200">
  <CommandCall Target="LegacyLoader.InvokeModuleProc">
    <Argument Name="PlugInName"> _
      USERSTATKEYBAROCX</Argument>
    <Argument Name="ProcID">11</Argument>
    <Argument Name="ParamList">KUKATPUSER; _
      DispSet.Bar</Argument>
  </CommandCall>
</MenuItem>
</MenuFolder>
</MenuFolder>
</MenuRoot>
...
</KukaRoboter.SmarthMI.Menu>
...
</configuration>
```


9 KUKA Service

9.1 技术支持咨询

引言 该文献将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 **提供咨询时需要以下信息：**

- 问题描述，包括故障持续时间及频率的说明
- 关于整个系统硬件和软件组件的尽可能全面的信息

以下列表提供了通常是相关信息的要点：

- 运作系统（例如机械手）的型号及序列号
- 控制系统型号及序列号
- 能量供应系统型号及序列号
- 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
- 诊断程序包 **KrcDiag**

针对 KUKA Sunrise 另外还需要：现有项目，包括应用程序

针对早于 V8 的库卡系统软件版本：软件档案（**KrcDiag** 在此尚不可用。）

- 现有的应用程序
- 现有的附加轴

9.2 库卡客户支持系统

可用性 在许多国家内均可使用库卡客户支持系统。如果您有问题，我们非常乐意为您提供咨询。

阿根廷 Ruben Costantini S.A. 公司（代理公司）
Luis Angel Huergo 13 20
Parque Industrial
2400 San Francisco (CBA)
阿根廷
电话 +54 3564 421033
传真 +54 3564 428877
ventas@costantini-sa.com

澳大利亚 KUKA Robotics Australia Pty Ltd
45 Fennell Street
Port Melbourne VIC 3207
澳大利亚
电话 +61 3 9939 9656
info@kuka-robotics.com.au
www.kuka-robotics.com.au

比利时	KUKA Automatisering + Robots N.V. (库卡自动化及机器人 N.V 公司) Centrum Zuid 1031 3530 Houthalen 比利时 电话 +32 11 516160 传真 +32 11 526794 info@kuka.be www.kuka.be
巴西	KUKA Roboter do Brasil Ltda. 库卡机器人巴西有限公司 Travessa Claudio Armando, nº 171 Bloco 5 - Galpões 51/52 Bairro Assunção CEP 09861-7630 São Bernardo do Campo - SP 巴西 电话 +55 11 4942-8299 传真 +55 11 2201-7883 info@kuka-roboter.com.br www.kuka-roboter.com.br
智利	Robotec S.A. (代理公司) Santiago de Chile 智利 电话 +56 2 331-5951 传真 +56 2 331-5952 robotec@robotec.cl www.robotec.cl
中国	库卡机器人中国有限公司 No. 889 Kungang Road Xiaokunshan Town Songjiang District 201614 Shanghai 中华人民共和国 电话 +86 21 5707 2688 传真 +86 21 5707 2603 info@kuka-robotics.cn www.kuka-robotics.com
Deutschland (德国)	KUKA Roboter GmbH (库卡机器人有限公司) Zugspitzstr. 140 86165 Augsburg Deutschland (德国) 电话 +49 821 797-4000 传真 +49 821 797-1616 info@kuka-roboter.de www.kuka-roboter.de

法国	KUKA Automatisme + Robotique SAS Techvallée 6, Avenue du Parc 91140 Villebon S/Yvette 法国 电话 +33 1 6931660-0 传真 +33 1 6931660-1 commercial@kuka.fr www.kuka.fr
印度	库卡机器人（印度）私人 有限公司 Office Number-7, German Centre, Level 12, Building No. - 9B DLF Cyber City Phase III 122 002 Gurgaon Haryana 印度 电话 +91 124 4635774 传真 +91 124 4635773 info@kuka.in www.kuka.in
意大利	KUKA Roboter Italia S.p.A. Via Pavia 9/a - int.6 10098 Rivoli (TO) 意大利 电话 +39 011 959-5013 传真 +39 011 959-5141 kuka@kuka.it www.kuka.it
日本	KUKA ロボティクスジャパン株式会社 YBP Technical Center 134 Godo-cho, Hodogaya-ku Yokohama, Kanagawa 240 0005 日本 电话 +81 45 744 7691 传真 +81 45 744 7696 info@kuka.co.jp
加拿大	KUKA Robotics Canada Ltd. 6710 Maritz Drive - Unit 4 Mississauga L5W 0A1 Ontario 加拿大 电话 +1 905 670-8600 传真 +1 905 670-8604 info@kukarobotics.com www.kuka-robotics.com/canada

韩国	库卡机器人韩国有限公司 RIT Center 306, Gyeonggi Technopark 1271-11 Sa 3-dong, Sangnok-gu Ansan City, Gyeonggi Do 426-901 韩国 电话 +82 31 501-1451 传真 +82 31 501-1461 info@kukakorea.com
马来西亚	库卡机器人自动化（泰国（M）Sdn Bhd）公司 South East Asia Regional Office No. 7, Jalan TPP 6/6 Taman Perindustrian Puchong 47100 Puchong Selangor 马来西亚 电话 +60 (03) 8063-1792 传真 +60 (03) 8060-7386 info@kuka.com.my
墨西哥	KUKA de México S. de R.L. de C.V. Progreso #8 Col. Centro Industrial Puente de Vigas Tlalnepantla de Baz 54020 Estado de México 墨西哥 电话 +52 55 5203-8407 传真 +52 55 5203-8148 info@kuka.com.mx www.kuka-robotics.com/mexico
挪威	库卡机器人公司 Sentrumsvegen 5 2867 Hov 挪威 电话 +47 61 18 91 30 传真 +47 61 18 62 00 info@kuka.no
奥地利	KUKA Roboter CEE GmbH（库卡机器人 CEE 有限公司） Gruberstraße 2-4 4020 Linz 奥地利 电话 +43 7 32 78 47 52 传真 +43 7 32 79 38 80 office@kuka-roboter.at www.kuka.at

波兰
库卡机器人（奥地利）有限公司
Spółka z ograniczoną odpowiedzialnością
Oddział w Polsce
Ul. Porcelanowa 10
40-246 Katowice
波兰
电话 +48 327 30 32 13 or -14
传真 +48 327 30 32 26
ServicePL@kuka-roboter.de

葡萄牙
KUKA Robots IBÉRICA, S.A.
Rua do Alto da Guerra n° 50
Armazém 04
2910 011 Setúbal
葡萄牙
电话 +351 265 729 780
传真 +351 265 729 782
info.portugal@kukapt.com
www.kuka.com

俄罗斯
KUKA Robotics RUS
Werbnaja ul. 8A
107143 Moskau
俄罗斯
电话 +7 495 781-31-20
传真 +7 495 781-31-19
info@kuka-robotics.ru
www.kuka-robotics.ru

瑞典
库卡自动化及机器人公司
A. Odhners gata 15
421 30 Västra Frölunda
瑞典
电话 +46 31 7266-200
传真 +46 31 7266-201
info@kuka.se

瑞士
KUKA Roboter Schweiz AG
Industriestr. 9
5432 Neuenhof
瑞士
电话 +41 44 74490-90
传真 +41 44 74490-91
info@kuka-roboter.ch
www.kuka-roboter.ch

西班牙	<p>KUKA Robots IBÉRICA, S. A. Pol. Industrial Torrent de la Pastera Carrer del Bages s/n 08800 Vilanova i la Geltrú (Barcelona) 西班牙 电话 +34 93 8142-353 传真 +34 93 8142-950 comercial@kukarob.es www.kuka.es</p>
南非	<p>Jendamark Automation 有限公司（代理公司） 76a York Road North End 6000 Port Elizabeth 南非 电话 +27 41 391 4700 传真 +27 41 373 3869 www.jendamark.co.za</p>
台湾	<p>库卡机器人自动化（台湾）有限公司 No. 249 Pujong Road Jungli City, Taoyuan County 320 台湾 电话 +886 3 4331988 传真 +886 3 4331948 info@kuka.com.tw www.kuka.com.tw</p>
泰国	<p>库卡机器人自动化（泰国（M）Sdn Bhd）公司 Thailand Office c/o Maccall System Co. Ltd. 49/9-10 Soi Kingkaew 30 Kingkaew Road Tt. Rachatheva, A. Bangpli Samutprakarn 10540 泰国 电话 +66 2 7502737 传真 +66 2 6612355 atika@ji-net.com www.kuka-roboter.de</p>
捷克	<p>库卡机器人（奥地利）有限公司 Organisation Tschechien und Slowakei Sezemická 2757/2 193 00 Praha Horní Počernice 捷克共和国 电话 +420 22 62 12 27 2 传真 +420 22 62 12 27 0 support@kuka.cz</p>

匈牙利 KUKA Robotics Hungária Kft. 公司
Fő út 140
2335 Taksony
匈牙利
电话 +36 24 501609
传真 +36 24 477031
info@kuka-robotics.hu

美国 库卡机器人公司
51870 Shelby Parkway
Shelby Township
48315-1787
Michigan
美国
电话 +1 866 873-5852
传真 +1 866 329-5852
info@kukarobotics.com
www.kukarobotics.com

英国 KUKA Robotics UK Ltd
Great Western Street
Wednesbury West Midlands
WS10 7LL
英国
电话 +44 121 505 9970
传真 +44 121 505 6589
service@kuka-robotics.co.uk
www.kuka-robotics.co.uk

索引

符号

#SWITCH 40
#TOGGLE 40
\$ 20
\$CONFIG.DAT 36, 37, 58

A

ASS 35
ASSAGG 35

D

DECL FOLD 30
DECL INLINEFORM 30
DECL PARAM 23
DECL PARAM PL_ 36
DECL PLIST 37
DECL STATKEY 38
DECL STATKEYBAR 44
DEFSCRIPT ... ENDSCRIPT 45
DEFTP ... ENDTP 22
DFCT 34
DO 47
DOALWAYS 48, 62
DSUB 33

F

FCT 34

I

IncOnOpen, 密钥 25
ISCHANGE 48, 61

K

KFD 8
KFD 文件 8
KFD 文件, 创建 22
KFD, 句法概览 17
KFD, 重要概念 18
KRL 8
KRL 代码, 生成 30
KUKA smarthMI 8
KUKA smartPAD 8

M

MESSAGE 47

N

NEED_MODEOP 41, 44
NEED_PROSTATE 40, 43
NEED_PROSTATEO 40, 43, 66

O

ONACCEPT 32, 60, 61
ONOPEN 32, 60, 61, 62
ONTOUCHUP 32

R

REDECL 46

S

Service, KUKA Roboter GmbH 71
SET 45
SETVAR 45
SHOWVAR 46
smarthMI 8
SmarthMI.User.Config 49, 50
smartPAD 8
SUB 33
SWITCH ... CASE (ELSE) ... END SWITCH 47
SWITCH DIALOG ... CASE ... END SWITCH 48

U

UserTech 界面, 更新 15
UserTech 重新初始化 (菜单项) 15

W

WYSIWYG 32

Z

安全 11
安全提示 7
安装 13
编程 17
布局 41
菜单元素, 定义 51
参数列表, 编程 36
参数列表, 示例程序 56
参数引用 19
操作 15
产品说明 9
存档 53
概览, UserTech 9
概念, 所用 8
格式类型 32
更新, UserTech 13
工艺, 编程 21
工艺, 集成于 smarthMI 49
工艺键, 编程 38
工艺键激活, 选项 40, 43
功能 9
关键词 20
技术支持咨询 71
脚本, 编程 45
脚本, 调用 59
脚本, 预定义 49
结构数据类型, 定义 37
库卡客户支持系统 71
名称 19
培训 7
软件 13
商标 8
示例 55
适用范围 20
数据类型, 简单 20
所用概念 8
特殊字符 18
提示 7

图像大小 41
文献，工业机器人 7
系统前提条件 13
卸载，UserTech 13
引言 7
语言系统，密匙 49
注释 18
字段类型 24
字符 17
字符串 19
字体 17

