

工业自动化通用组态软件

组态王

Version 7.0

用户手册

运行于 Win XP/Win7 简体中文版

重要提示

本手册仅用于提供组态王软件使用的相关信息，若有变化恕不事先通知。本手册所描述的软件是在授权或不扩散协议下完成的，本软件只能按合同规定的条款使用或拷贝。在没有**亚控公司**书面允许的前提下，除购买者自己使用外，不得为任何其他目的、使用任何方法（包括复印和录制在内的电子或机械手段）对本手册的任何部分进行复制或传播。如果你在手册中发现了某些问题，请以书面形式把它们记录下来，并与我们取得联系，亚控公司将表示衷心的感谢。亚控公司不保证手册中没有错误。

组态王是亚控公司的注册商标。

ORACLE 是 Oracle Corporation（Oracle 公司）的一个注册商标。Windows 98、Windows NT、Windows 2000、Windows XP、Excel、ODBC 是 Microsoft Corporation 的一个注册商标。所有其它的产品和标记名称都是它们各自公司的商标。

《组态王 7.0 使用手册》

2015 **亚控公司** 版权所有，保留所有权利

版本号：7.0

如果需要技术支持，请与我们联系：

电话：010-59309666

传真：010-59309600

E-Mail: support@wellintech.com sales@wellintech.com

网站: <http://www.wellintech.com/>

技术 BBS: <http://www.wellintech.com/bbs/>

目录

第一章 欢迎使用组态王 7.0.....	1-1
1.1 组态王 7.0 软件运行的软硬件环境	1-1
1.2 安装组态王 7.0	1-2
1.2.1 主程序安装	1-4
1.2.2 设备驱动程序安装	1-16
1.2.3 加密锁驱动程序的安装	1-24
1.2.3.1 KV6.5X 加密锁驱动安装	1-25
1.2.3.2 KV7.0 加密锁驱动安装	1-26
1.3 卸载组态王 7.0 主程序、驱动程序	1-28
1.4 修复或修改安装组件	1-30
1.5 加密锁驱动升级	1-31
1.6 特别说明	1-31
1.6.1 光盘里其它有价值的内容	1-31
1.6.2 组态王 7.0 版本类型	1-32
1.6.3 组态王 7.0 加密锁类型	1-32
1.7 如何得到我们的帮助	1-33
第二章 组态王 7.0 使用入门.....	2-1
2.1 软件基本构成	2-1
2.2 工程组态	2-3
2.2.1 创建组态王 7.0 工程	2-3
2.2.2 定义 IO 设备	2-7
2.2.4 数据库建点	2-15
2.2.2 组态画面	2-17
2.2.2.1 新建画面	2-18
2.2.2.2 创建图形对象	2-19
2.2.5 建立动画连接	2-21
2.2.6 运行和调试	2-26
2.3 组态王 7.0 的升级和动态分辨率转换	2-27
第三章 启动组态王 7.0 - 工程管理器	3-2

3.1 认识工程管理器	3-2
3.1.1 菜单条	3-3
3.1.1.1 文件菜单	3-3
3.1.1.2 视图菜单	3-6
3.1.1.3 工具菜单	3-7
3.1.1.4 帮助菜单	3-9
3.1.2 工具栏	3-9
3.1.3 工程信息显示区	3-10
3.1.4 状态栏	3-10
3.1.5 快捷菜单	3-10
3.2 如何管理工程	3-11
3.2.1 新建一个工程	3-11
3.2.2 如何找到一个已有的组态王 7.0 工程	3-14
3.2.3 如何找到一些已有的组态王 7.0 工程	3-16
3.2.4 如何设置一个工程为当前工程	3-17
3.2.5 如何修改当前工程的属性	3-18
3.2.6 如何清除当前不需要显示的工程	3-19
3.2.7 如何备份和恢复工程	3-20
3.2.7.1 工程备份	3-20
3.2.7.2 工程恢复	3-22
3.2.8 如何删除工程	3-25
第四章 组态王 7.0 开发环境 - 工程浏览器	4-3
4.1 工程浏览器概述	4-3
4.1.1 菜单栏	4-4
4.1.1.1 工程菜单	4-4
4.1.1.2 配置菜单	4-17
4.1.1.3 查看菜单	4-25
4.1.1.4 工具菜单	4-26
4.1.1.5 帮助菜单	4-34
4.1.2 工具栏	4-35

4.1.3	Tab 页标签.....	4-37
4.1.4	目录显示区.....	4-37
4.1.4.1	目录显示区显示项.....	4-38
4.1.4.2	目录显示区操作方法.....	4-43
4.1.5	内容显示区.....	4-43
4.1.5.1	内容显示区操作方法.....	4-43
4.1.6	状态栏.....	4-43
4.2	组态王 7.0 画面开发系统.....	4-44
4.2.1	菜单详解.....	4-44
4.2.1.1	文件菜单.....	4-45
4.2.1.2	编辑菜单.....	4-58
4.2.1.3	排列菜单.....	4-79
4.2.1.4	工具菜单.....	4-88
4.2.1.5	图库菜单.....	4-117
4.2.1.6	画面菜单.....	4-119
4.2.1.7	帮助菜单.....	4-119
4.2.2	工具箱.....	4-120
4.2.2.1	简单图素对象.....	4-120
4.2.2.2	复杂图素对象.....	4-120
4.2.3	画面.....	4-120
4.2.3.1	如何创建一个画面.....	4-120
4.2.3.2	画面属性.....	4-123
4.2.3.3	对画面的操作.....	4-124
4.2.4	组态王 7.0 开发系统菜单命令汇总及键盘快捷键.....	4-126
第五章	变量定义和管理.....	5-3
5.1	变量的类型.....	5-4
5.1.1	基本变量类型.....	5-4
5.1.2	变量的数据类型.....	5-4
5.1.3	特殊变量类型.....	5-6
5.2	基本变量的定义.....	5-9

5.2.1	基本属性定义.....	5-10
5.2.2	报警定义.....	5-15
5.2.3	记录和安全定义.....	5-16
5.3	结构变量.....	5-16
5.3.1	什么是结构变量.....	5-16
5.3.2	结构变量的定义.....	5-17
5.3.3	结构变量的使用.....	5-23
5.4	变量的属性——变量域.....	5-27
5.4.1	什么是变量的域.....	5-28
5.4.2	变量有哪些域.....	5-28
5.4.2.1	变量的基本属性域.....	5-28
5.4.2.2	变量的报警域.....	5-30
5.4.2.3	变量的历史记录起停控制域.....	5-32
5.4.2.4	报警窗口的域.....	5-33
5.4.2.5	历史趋势曲线的域.....	5-34
5.4.3	变量域的使用.....	5-34
5.5	IO 变量的转换方式.....	5-36
5.5.1	线性转换方式.....	5-36
5.5.2	开方转换方式.....	5-39
5.5.3	非线性表转换方式.....	5-40
5.5.3.1	非线性表的定义.....	5-40
5.5.3.2	非线性表的导入、导出.....	5-45
5.5.4	累计转换方式.....	5-48
5.6	变量管理工具一 —— 管理变量.....	5-51
5.6.1	如何建立变量组.....	5-51
5.6.2	如何在变量组中增加变量.....	5-56
5.6.3	变量组内变量排序.....	5-57
5.6.4	如何在变量组中删除变量.....	5-58
5.6.5	如何删除变量组.....	5-59
5.7	变量管理工具二——导入导出、更新替换、获取变量使用情况.....	5-59

5.7.1	数据词典导入导出到 Excel 中	5-59
5.7.1.1	数据词典导出到 Excel.....	5-59
5.7.1.2	从 Excel 中导入数据词典.....	5-65
5.7.2	数据词典导入导出到 Access 中	5-68
5.7.2.1	数据词典导出到 Access	5-68
5.7.2.2	从 Access 中导入数据词典	5-73
5.7.3	数据词典导入导出到工程	5-73
5.7.3.1	将数据词典导入到原工程.....	5-73
5.7.3.2	将数据词典导入到其它工程.....	5-82
5.7.4	获取变量使用情况信息和删除变量	5-83
5.7.4.1	如何得到变量使用信息.....	5-83
5.7.4.2	变量使用更新.....	5-85
5.7.4.3	删除未用的变量.....	5-85
5.7.5	变量属性修改和变量替换	5-87
5.7.5.1	查找数据库变量.....	5-87
5.7.5.2	多选修改变量属性.....	5-88
5.7.5.3	变量替换.....	5-89
5.8	关于自定义变量	5-103
5.8.1	自定义变量.....	5-104
5.8.2	自定义结构变量.....	5-106
5.9	变量指针引用	5-109
5.9.1	普通变量引用	5-109
5.9.2	结构变量引用	5-111
5.9.3	结构变量和普通变量之间的引用	5-113
5.10	变量的驱动转发	5-114
5.11	自动建变量	5-117
第六章	I/O 设备管理	6-2
6.1	设备管理	6-3
6.1.1	了解组态王 7.0 逻辑设备概念	6-4
6.1.2	组态王 7.0 逻辑设备的分类	6-7

6.1.2.1	DDE 设备	6-7
6.1.2.2	板卡类设备	6-8
6.1.2.3	串口类设备	6-9
6.1.2.4	人机界面卡	6-10
6.1.2.5	网络模块	6-11
6.1.3	如何定义 IO 设备	6-12
6.1.3.1	如何定义 DDE 设备	6-12
6.1.3.2	如何定义板卡类设备	6-16
6.1.3.3	如何定义串口类设备以及设置串口参数	6-22
6.1.3.4	如何定义 OPC 服务器设备	6-30
6.1.3.5	如何定义带网络模块的设备	6-31
6.1.3.6	如何配置组态王 7.0 作为网络设备	6-34
6.1.4	选择驱动	6-38
6.1.5	驱动查找	6-40
6.1.6	显示本地信息	6-41
6.2	组态王 7.0 提供的模拟设备—仿真 PLC	6-42
6.2.1	仿真 PLC 的定义	6-42
6.2.2	仿真 PLC 的寄存器	6-49
6.2.2.1	自动加 1 寄存器 INCREA	6-49
6.2.2.2	自动减 1 寄存器 DECREA	6-50
6.2.2.3	静态寄存器 STATIC	6-51
6.2.2.4	随机寄存器 RADOM	6-51
6.2.2.5	STRING 寄存器	6-52
6.2.2.6	CommErr 寄存器	6-52
6.2.3	仿真 PLC 使用举例	6-52
6.2.3.1	仿真 PLC 的定义	6-52
6.2.3.2	定义 I/O 变量	6-53
6.2.3.3	制作画面	6-55
6.2.3.4	运行画面程序	6-56
6.3	组态王 7.0 提供的通讯的其它特殊功能	6-56

6.3.1	开发环境下的设备通讯测试	6-56
6.3.2	如何在运行系统中判断和控制设备通讯状态	6-61
6.3.3	如何使用 MODEM 对设备进行远程拨号采集数据	6-62
6.3.3.1	拨号设置	6-62
6.3.3.2	设备拨号使用	6-63
6.3.4	如何使用 GPRS 对设备进行远程通讯	6-65
6.3.4.1	定义 GPRS--虚拟串口设备	6-66
6.3.4.2	定义 GPRS 变量	6-72
6.4	用户如何自己开发驱动程序——驱动程序开发包	6-73
第七章	图形画面与动画连接	7-2
7.1	动画连接概述	7-2
7.1.1	连接概述	7-2
7.1.2	动画连接对话框	7-3
7.2	通用控制项目	7-7
7.2.1	图形编辑工具	7-7
7.2.1.1	工具箱简介	7-7
7.2.1.2	工具箱速览	7-10
7.2.1.3	工具箱详解	7-11
7.2.1.4	画刷类型工具的使用	7-20
7.2.1.5	线形类型工具的使用	7-24
7.2.1.6	调色板的使用	7-25
7.2.2	变量浏览器的使用	7-34
7.2.3	表达式和运算符	7-40
7.2.3.1	运算符	7-41
7.2.3.2	优先级	7-42
7.2.3.3	表达式举例	7-43
7.3	动画连接详解	7-43
7.3.1	线属性连接	7-43
7.3.2	填充属性连接	7-45
7.3.3	文本色连接	7-47

7.3.4	水平移动连接	7-48
7.3.5	垂直移动连接	7-50
7.3.6	缩放连接	7-51
7.3.7	旋转连接	7-53
7.3.8	填充连接	7-55
7.3.9	模拟值输出连接	7-58
7.3.10	离散值输出连接	7-60
7.3.11	字符串输出连接	7-61
7.3.12	模拟值输入连接	7-63
7.3.13	离散值输入连接	7-65
7.3.14	字符串输入连接	7-67
7.3.15	闪烁连接	7-70
7.3.16	隐含连接	7-71
7.3.17	流动连接	7-72
7.3.18	水平滑动杆输入连接	7-74
7.3.19	垂直滑动杆输入连接	7-75
7.3.20	动画连接命令语言	7-76
7.4	动画连接向导的使用	7-78
7.4.1	水平移动动画连接向导	7-78
7.4.2	垂直移动动画连接向导	7-79
7.4.3	滑动杆输入动画连接向导	7-81
7.4.4	旋转动画连接向导	7-81
第八章	趋势曲线和其它曲线	8-2
8.1	曲线的一般介绍	8-2
8.2	实时趋势曲线	8-3
8.2.1	组态王内置实时趋势曲线	8-3
8.2.1.1	创建实时趋势曲线	8-4
8.2.1.2	实时趋势曲线属性	8-4
8.2.1.3	为实时趋势曲线建立“笔”	8-10
8.2.2	实时趋势曲线控件	8-10

8.2.2.1	创建实时趋势曲线控件.....	8-11
8.2.2.2	实时曲线控件的属性设置.....	8-12
8.2.2.3	运行时修改实时曲线属性.....	8-18
8.3	历史趋势曲线.....	8-22
8.3.1	与历史趋势曲线有关的其它必配置项.....	8-23
8.3.1.1	定义变量范围.....	8-23
8.3.1.2	对某变量作历史记录.....	8-24
8.3.1.3	定义历史库数据文件的存储目录.....	8-25
8.3.1.4	重启历史数据记录.....	8-27
8.3.2	通用历史趋势曲线.....	8-27
8.3.2.1	历史趋势曲线的定义.....	8-27
8.3.2.2	历史趋势曲线对话框.....	8-28
8.3.2.3	历史趋势曲线操作按钮.....	8-34
8.3.2.4	历史趋势曲线时间轴指示器.....	8-39
8.3.3	历史趋势曲线控件.....	8-40
8.3.3.1	创建历史曲线控件.....	8-41
8.3.3.2	设置历史曲线固有属性.....	8-42
8.3.3.3	设置历史曲线的动画连接属性.....	8-72
8.3.3.4	历史曲线属性、方法列表.....	8-75
8.3.3.5	运行时修改历史曲线属性.....	8-100
8.3.4	个性化历史趋势曲线.....	8-107
8.3.4.1	历史趋势曲线的定义.....	8-107
8.3.4.2	历史趋势曲线对话框.....	8-108
8.3.4.3	建立历史趋势曲线运行时的操作按钮.....	8-113
8.3.5	访问远程历史数据库.....	8-124
8.4	组态王内置温控曲线.....	8-124
8.4.1	在画面上放置温控曲线.....	8-124
8.4.2	温控曲线属性设置.....	8-125
8.5	超级 X-Y 曲线控件.....	8-136
8.5.1	创建超级 X-Y 曲线.....	8-137

8.5.2	设置超级 X-Y 曲线的固有属性	8-138
8.5.3	设置超级 X-Y 曲线的动画连接属性	8-144
8.5.4	超级 X-Y 曲线的使用	8-145
第九章	报警和事件系统	9-2
9.1	关于报警和事件	9-3
9.2	报警组	9-3
9.2.1	报警组概述	9-3
9.2.2	报警组定义	9-4
9.3	如何定义变量的报警属性	9-8
9.3.1	通用报警属性功能介绍	9-9
9.3.2	模拟型变量的报警类型	9-10
9.3.2.1	越限报警	9-11
9.3.2.2	偏差报警	9-16
9.3.2.3	变化率报警	9-19
9.3.2.4	报警延时和死区	9-20
9.3.3	离散型变量的报警类型	9-24
9.4	事件类型及使用方法	9-25
9.4.1	操作事件	9-25
9.4.2	用户登录事件	9-27
9.4.3	应用程序事件	9-28
9.4.4	工作站事件	9-29
9.5	如何记录、显示报警	9-29
9.5.1	报警输出显示：报警窗口	9-30
9.5.1.1	报警缓冲区大小的定义	9-30
9.5.1.2	创建报警窗口	9-31
9.5.1.3	如何配置实时和历史报警窗	9-32
9.5.1.4	运行系统中报警窗的操作	9-40
9.5.1.5	报警窗单击事件转发控件使用说明	9-43
9.5.2	报警记录输出一：文件输出	9-49
9.5.2.1	报警配置——文件输出配置	9-50

9.5.2.2	通用报警和事件记录格式配置	9-53
9.5.3	报警记录输出二：数据库	9-60
9.5.3.1	定义报警记录数据库	9-61
9.5.3.2	报警输出数据库配置	9-64
9.5.3.3	记录报警事件到数据库的格式配置	9-66
9.5.4	报警记录输出三：实时打印输出	9-71
9.6	报警相关的函数和变量的报警域	9-73
9.6.1	\$新报警变量的使用	9-73
9.6.2	报警相关的函数	9-73
9.6.3	变量的报警域	9-79
9.7	报警预置自定义函数	9-81
9.8	利用短信和 E-Mail 发送您的报警信息	9-86
9.8.1	短信功能	9-86
9.8.2	E-Mail 功能	9-90
第十章	命令语言程序	10-2
10.1	命令语言类型	10-2
10.1.1	应用程序命令语言	10-3
10.1.1.1	认识命令语言编辑器	10-3
10.1.1.3	应用程序命令语言的定义	10-15
10.1.2	数据改变命令语言	10-16
10.1.3	事件命令语言	10-18
10.1.4	热键命令语言	10-19
10.1.5	用户自定义函数	10-21
10.1.5.1	自定义函数里的概念	10-22
10.1.5.2	自定义函数的定义和使用	10-22
10.1.6	画面命令语言	10-26
10.1.7	动画连接命令语言	10-28
10.2	命令语言语法	10-30
10.2.1	运算符	10-30
10.2.2	赋值语句	10-32

10.2.3	If-Else 语句.....	10-33
10.2.4	While () 语句.....	10-35
10.2.5	命令语言程序的注释方法.....	10-36
10.3	命令语言执行中如何跟踪变量的值.....	10-37
10.4	在命令语言中使用自定义变量.....	10-38
10.5	命令语言函数及使用方法.....	10-39
第十一章	组态王运行系统.....	11-1
11.1	配置运行系统.....	11-1
11.2	运行系统菜单详解.....	11-6
11.2.1	画面菜单.....	11-7
11.2.2	特殊菜单.....	11-10
11.2.3	调试菜单.....	11-15
11.2.4	导航菜单.....	11-17
11.2.5	关于菜单.....	11-20
第十二章	组态王 7.0 信息窗口.....	12-1
12.1	如何从信息窗口中获取信息.....	12-1
12.2	如何保存信息窗口中的信息.....	12-3
12.2.1	设置保存路径.....	12-3
12.2.2	设置保存参数.....	12-4
12.2.3	信息窗口的内容保存为 txt 文件.....	12-5
12.3	如何查看历史存储信息.....	12-7
12.4	如何打印信息窗口中的信息.....	12-8
12.4.1	信息\打印设置.....	12-9
12.4.2	信息\打印.....	12-10
12.5	信息窗口其他菜单的使用.....	12-11
12.5.1	查看菜单.....	12-11
12.5.2	关于菜单.....	12-12
12.5.3	系统菜单.....	12-13
第十三章	图库.....	13-1
13.1	图库概述.....	13-1

13.2	认识图库精灵	13-2
13.3	如何创建图库精灵	13-5
13.4	如何使用图库精灵	13-9
13.5	如何管理图库	13-13
13.5.1	图库管理器菜单条	13-14
13.5.2	图库管理器工具条	13-20
13.5.3	图库显示区	13-20
13.5.4	精灵显示区	13-20
13.6	如何将图库精灵转换成普通图素	13-20
13.7	关于图库开发包	13-22
第十四章	控件	14-3
14.1	控件简介	14-4
14.1.1	什么是控件	14-4
14.1.2	控件的功能	14-4
14.1.3	组态王 7.0 支持的控件	14-5
14.2	组态王 7.0 内置控件	14-6
14.2.1	立体棒图控件	14-7
14.2.1.2	设置棒图控件的属性	14-9
14.2.1.3	如何使用棒图控件	14-15
14.2.2	温控曲线控件	14-22
14.2.3	X-Y 轴曲线控件	14-23
14.2.4	列表框和组合框控件	14-23
14.2.4.1	如何创建列表框控件	14-23
14.2.4.2	设置列表框控件的属性	14-24
14.2.4.3	如何使用列表框控件	14-26
14.2.4.4	如何使用组合框控件	14-32
14.2.5	复选框控件	14-34
14.2.5.1	如何创建复选框控件	14-35
14.2.5.2	如何设置复选框控件的属性	14-35
14.2.5.3	如何使用复选框控件	14-37

14.2.6	编辑框控件	14-38
14.2.6.1	如何创建编辑框控件	14-38
14.2.6.2	如何定义编辑框控件属性	14-39
14.2.6.3	如何使用编辑框控件	14-40
14.2.7	单选按钮控件	14-41
14.2.7.1	如何创建单选按钮控件	14-41
14.2.7.2	如何定义单选按钮控件属性	14-42
14.2.7.3	如何使用单选按钮控件	14-44
14.2.8	超级文本显示控件	14-45
14.2.8.1	如何创建超级文本显示控件	14-45
14.2.8.2	如何定义超级文本显示控件的属性	14-46
14.2.8.3	如何使用超级文本显示控件	14-47
14.2.9	多媒体控件	14-52
14.2.9.1	AVI 动画播放控件	14-52
14.2.9.2	视频动画控件	14-55
14.2.9.3	AVICapture 视频动画控件	14-58
14.3	Active X 控件	14-58
14.3.1	Active X 控件概述	14-58
14.3.2	如何在组态王 7.0 中使用 Active X 控件	14-59
14.3.2.1	创建 Active X 控件	14-59
14.3.2.2	设置 Active X 控件的固有属性	14-62
14.3.2.3	设置 Active X 控件的动画连接属性	14-64
14.3.2.4	在命令语言中调用 Active X 控件的属性和方法	14-77
14.3.2.5	一个 Active X 控件的使用例程	14-83
14.3.3	组态王 7.0 提供的 Video 视频控件	14-92
14.3.3.1	画面上创建 Video 视频控件	14-92
14.3.3.2	设置 Video 视频控件的属性	14-93
14.3.3.3	Video 视频控件的使用	14-94
14.3.4	开放型数据库访问控件	14-100
14.3.4.1	创建 KVADODBGrid 控件	14-101

14.3.4.2	设置 KVADODBGrid 控件的固有属性.....	14-103
14.3.4.3	设置 KVADODBGrid 控件的动画连接属性.....	14-110
14.3.4.4	KVADODBGrid 控件的使用	14-111
14.3.4.5	KVADODBGrid 控件的查询条件向导.....	14-135
14.3.4.6	KVADODBGrid 控件连接数据库举例.....	14-137
14.3.5	树形控件.....	14-139
14.3.5.1	创建 KingTree 控件	14-140
14.3.5.2	设置 KingTree 控件的固有属性.....	14-141
14.3.5.3	设置 KingTree 控件的动画连接属性.....	14-146
14.3.5.4	KingTree 控件的使用	14-149
14.3.6	温控曲线控件	14-152
14.3.6.1	温控表格控件概述.....	14-155
14.3.6.2	如何在组态王 7.0 中使用温控表格控件	14-155
14.3.6.3	温控曲线.....	14-178
14.3.7	旋转装置控件	14-195
14.3.7.1	控件简介.....	14-195
14.3.7.2	旋转效果配置文件说明.....	14-195
14.3.7.3	控件使用方法.....	14-196
14.3.8	PID 控件	14-198
14.3.8.1	控件简介.....	14-198
14.3.8.2	如何创建 PID 控件	14-199
14.3.8.3	如何定义 PID 属性	14-200
14.3.8.4	如何使用 PID 控件	14-207
14.3.9	位图控件.....	14-212
14.3.9.1	控件简介.....	14-212
14.3.9.2	使用方法.....	14-212
第十五章	配方管理.....	15-1
15.1	配方管理概述	15-2
15.1.1	什么是配方	15-2
15.1.2	组态王 7.0 中的配方管理	15-3

15.1.3	配方的工作原理.....	15-4
15.2	如何创建配方模板.....	15-5
15.2.1	菜单栏.....	15-7
15.2.1.1	表格菜单.....	15-8
15.2.1.2	工具菜单.....	15-10
15.2.1.3	变量菜单.....	15-12
15.2.2	创建配方模板.....	15-12
15.3	如何使用配方.....	15-14
15.3.1	配方命令语言函数.....	15-15
15.3.2	配方示例.....	15-24
第十六章	系统安全管理.....	16-1
16.1	组态王开发系统安全管理.....	16-2
16.1.1	如何对工程进行加密.....	16-2
16.1.2	如何去除工程加密.....	16-4
16.2	组态王运行系统安全管理.....	16-5
16.2.1	运行系统安全管理概述.....	16-5
16.2.2	安全管理配置.....	16-6
16.2.2.1	优先级和安全区.....	16-6
16.2.2.2	如何配置用户.....	16-7
16.2.2.3	如何设置对象的安全属性.....	16-14
16.2.3	运行时如何登录用户.....	16-19
16.2.3.1	登录用户.....	16-19
16.2.3.2	退出登录.....	16-23
16.2.4	运行时如何重新设置口令和权限.....	16-23
16.2.5	与安全管理相关的系统变量和函数.....	16-25
16.2.5.1	系统变量.....	16-25
16.2.5.2	函数.....	16-25
第十七章	报表系统.....	17-2
17.1	创建报表.....	17-3
17.1.1	如何创建报表窗口.....	17-3

17.1.2	配置报表窗口的名称及格式套用	17-4
17.2	报表组态	17-7
17.2.1	认识报表工具箱与快捷菜单	17-7
17.2.2	定义报表单元格的保护属性	17-17
17.2.3	报表的其它快捷编辑方法	17-17
17.2.4	设置报表格式	17-18
17.3	报表函数	17-25
17.3.1	报表内部函数	17-25
17.3.2	报表的单元格操作函数	17-39
17.3.3	存取报表函数	17-45
17.3.4	报表统计函数	17-47
17.3.5	报表历史数据查询函数	17-47
17.3.6	报表打印类函数	17-65
17.3.6.1	报表打印函数	17-65
17.3.6.2	报表页面设置函数	17-66
17.3.6.3	报表打印预览函数	17-66
17.3.7	报表向导函数	17-67
17.3.8	报表行、列设置和锁定函数	17-71
17.4	如何套用报表模板	17-72
17.5	如何制作实时数据报表	17-73
17.5.1	单元格中直接引用变量	17-73
17.5.2	使用单元格设置函数	17-74
17.6	如何制作历史数据报表	17-75
17.6.1	向报表单元格中实时添加数据	17-75
17.6.2	使用历史数据查询函数	17-76
17.7	如何使用报表向导	17-77
17.7.1	制作各种报表	17-78
17.7.1.1	制作亚控数据库报表	17-78
17.7.1.2	制作关系数据库报表	17-85
17.7.2	重设报表查询时间	17-90

17.8	报表锁定行列功能	17-91
17.8.1	锁定报表的行列	17-91
17.8.2	用函数锁定报表的行列	17-93
17.8.3	解除锁定	17-93
第十八章	组态王 7.0 与其它应用程序的动态数据交换 (DDE)	18-1
18.1	动态数据交换的概念	18-2
18.2	组态王 7.0 与 Excel 间的数据交换	18-3
18.2.1	组态王 7.0 访问 Excel 的数据	18-4
18.2.2	Excel 访问组态王 7.0 的数据	18-10
18.3	组态王 7.0 与 VB 间的数据交换	18-15
18.3.1	组态王 7.0 访问 VB 的数据	18-16
18.3.2	VB 访问组态王 7.0 的数据	18-21
18.4	如何重新建立 DDE 连接	18-26
18.4.1	重新建立 DDE 连接菜单命令	18-26
18.4.2	重新建立 DDE 连接函数	18-28
第十九章	组态王数据库访问 (SQL)	19-1
19.1	组态王 SQL 访问管理器	19-2
19.1.1	表格模板	19-4
19.1.2	记录体	19-7
19.2	如何配置与数据库的连接	19-10
19.2.1	定义 ODBC 数据源	19-10
19.2.2	组态王支持的数据库及配置	19-14
19.2.2.1	Oracle6 数据库	19-14
19.2.2.2	Oracle7.2 数据库	19-16
19.2.2.3	SyBase 或 MS SQLServer 数据库	19-18
19.2.2.4	dBase 数据库	19-22
19.2.2.5	MS Access 数据库	19-23
19.3	组态王 SQL 使用简介	19-24
19.3.1	如何使组态王与数据库建立连接	19-24
19.3.2	如何创建一个表格	19-25

19.3.3	如何将数据存入数据库	19-26
19.3.4	如何进行数据查询	19-27
19.3.5	SQL 函数及 SQL 函数的参数	19-35
函数的参数		19-39
19.3.6	如何得到 SQL 错误信息	19-41
19.3.7	如何将数据库里的数据读到组态王中来	19-46
第二十章	OPC 设备	20-1
20.1	OPC 简介	20-2
20.1.1	什么是 OPC	20-2
20.1.2	OPC 的适用范围	20-3
20.1.3	OPC 的基本概念	20-4
20.1.4	组态王 7.0 与 OPC	20-8
20.2	组态王 7.0 作为 OPC 客户端的使用方法	20-9
20.2.1	建立和删除 OPC 设备	20-9
20.2.1.1	建立 OPC 设备	20-9
20.2.1.2	删除 OPC 设备	20-13
20.2.2	在 OPC 服务器中定义数据项	20-13
20.2.3	OPC 服务器与组态王 7.0 数据词典的连接	20-17
20.3	组态王 7.0 作为 OPC 服务器的使用	20-19
20.3.1	组态王 7.0 OPC 服务器的功能简介	20-19
20.3.2	组态王 7.0 OPC 服务器的使用	20-20
20.3.3	组态王 7.0 为用户提供的 OPC 接口	20-23
20.4	如何使用网络 OPC 通讯	20-23
20.4.1	使用网络 OPC 前的 DCOM 配置	20-24
20.4.1.1	配置充当 OPC 服务器的机器	20-24
20.4.1.2	客户端通过 OPC 连接服务器	20-29
第二十一章	网络功能	21-2
21.1	组态王 7.0 网络结构概述	21-2
21.2	网络配置	21-6
21.2.1	认识网络配置	21-6

21.2.2	网络配置实例	21-17
21.3	网络变量使用	21-27
21.3.1	远程变量的引用	21-27
21.3.2	远程变量的回写	21-30
21.4	分布式报警引用	21-31
21.4.1	分布式报警的概念	21-31
21.4.2	分布式报警的显示和记录	21-32
21.5	分布式历史数据访问	21-34
21.6	网络登录管理	21-41
21.7	网络精灵	21-46
21.7.1	本站点信息	21-47
21.7.2	客户端网络信息	21-49
21.7.3	服务器网络信息	21-50
21.7.4	网络精灵菜单功能	21-51
21.7.5	网络精灵信息窗口操作	21-52
第二十二章	冗余系统	22-1
22.1	双设备冗余	22-2
22.1.1	双设备冗余概述	22-2
22.1.2	双设备冗余的功能	22-3
22.1.3	双设备冗余的设置	22-3
22.1.3.1	双设备冗余设置的步骤	22-3
22.1.4	双设备冗余的使用	22-7
22.2	双机热备	22-7
22.2.1	双机热备的功能	22-8
22.2.2	双机热备实现的原理	22-8
22.2.3	网络工程的冗余	22-11
22.2.4	双机热备配置	22-12
22.2.4.1	主机网络配置	22-12
22.2.4.2	从机网络配置	22-14
22.2.4.3	双机热备状态变量的使用	22-16

22.3 双网络冗余	22-17
22.3.1 双网络冗余配置	22-19
22.3.1.1 网卡的配置	22-19
22.3.1.2 组态王网络配置	22-20
第二十三章 Web 全新版	23-2
23.1 Web 全新版简介	23-2
23.1.1 Web 全新版功能介绍	23-2
23.1.2 Web 全新版与 Web 功能的比较	23-5
23.1.3 相关文件介绍	23-5
23.2 工程配置	23-8
23.2.1 数据服务器中工程的网络配置	23-10
23.2.2 Web 服务器中工程的网络配置	23-15
23.3 Web 全新版发布示例	23-24
23.3.1 Web 全新版发布准备工作	23-24
23.3.2 Web 全新版发布步骤	23-28
23.4 如何在 IE 浏览客户端浏览工	23-41
23.4.1 浏览前的准备工作	23-41
23.4.2 浏览工程	23-49
23.5 数据库数据的发布	23-53
23.6 Web 全新版支持与不支持的功能	23-70
23.6.1 Web 全新版不支持的功能	23-70
23.6.2 Web 全新版不支持的函数	23-71
23.6.3 Web 全新版支持的功能和函数	23-73
第二十四章 组态王历史库	24-1
24.1 组态王变量的历史记录属性	24-2
24.2 历史记录存储及文件的格式	24-5
24.3 历史数据的查询	24-12
24.4 网络历史库的备份合并	24-14
24.5 组态王历史库服务程序	24-18
第二十五章 其它程序访问组态王数据的方式	25-1

25.1	访问组态王实时数据	25-2
25.1.1	通过 OPC 访问组态王	25-2
25.1.2	通过 DDE 访问组态王	25-2
25.1.3	通过 ActiveX 控件访问	25-2
25.1.3.1	KvTcpiClientOcx.ocx 控件提供的方法说明	25-3
25.1.3.2	KvTcpiClientOcx.ocx 中提供的事件说明	25-12
25.1.3.3	stamp_time_hi 和 stamp_time_lo 说明	25-15
25.1.3.4	应用 KvRealDBClient 控件注意事项	25-16
25.1.3.5	KvRealDBClient 工具使用简介	25-18
25.2	访问组态王历史数据	25-26
25.2.1	AccessHist 接口	25-26
25.2.1.1	接口函数	25-27
25.2.1.2	接口函数的使用	25-33
25.2.2	KingReport	25-34
第二十六章	API 接口及对象属性导入导出	26-2
26.1	xml 导出导入接口	26-2
26.2	组态对象属性	26-6
26.2.1	设备	26-7
26.2.1.1	类型说明	26-7
26.2.1.2	设备属性	26-7
26.2.1.3	设备导入导出	26-12
26.2.2	变量	26-12
26.2.2.1	变量 xml 节点属性	26-13
26.2.2.2	变量属性	26-14
26.2.2.3	变量导入导出	26-24
26.2.3	非线性表	26-24
26.2.3.1	非线性表属性	26-24
26.2.3.2	非线性表导入导出	26-25
27.2.4	配方	26-25
26.2.4.1	配方属性	26-26

26.2.4.2	配方导入导出.....	26-27
26.2.5	系统配置对象.....	26-28
26.2.5.1	系统配置对象属性.....	26-28
26.2.5.2	系统配置导入导出.....	26-45
26.2.6	命令语言对象.....	26-45
26.2.6.1	命令语言属性.....	26-45
26.2.6.2	命令语言导入导出.....	26-51
26.2.7	关系库对象.....	26-51
26.2.7.1	关系表属性.....	26-52
26.2.7.2	关系库对象导入导出.....	26-54
26.2.8	图形对象.....	26-54
26.2.8.1	基本图素属性.....	26-54
26.2.8.2	动画连接属性.....	26-74
26.2.8.3	精灵图对象属性.....	26-86
26.2.8.4	画面导入导出.....	26-88
第二十七章	模板.....	27-1
27.1	模板管理器.....	27-2
27.2	采集模板.....	27-16
27.2.1	如何生成采集模板.....	27-16
27.2.2	通过采集模板生成设备和变量.....	27-22
27.3	画面模板.....	27-26
27.3.1	如何生成画面模板.....	27-27
27.3.2	通过画面模板生成画面.....	27-33
27.4	脚本模板.....	27-38
27.4.1	如何生成脚本模板.....	27-39
27.4.2	通过脚本模板生成命令语言.....	27-44
27.5	图形模板.....	27-48
27.5.1	如何生成图形模板.....	27-49
27.5.2	通过图形模板生成画面.....	27-54
附录 A	板卡数据类型.....	A-1

目录

附录 B 组态王支持的硬件设备	B-1
附录 C 组态王 7.0 新功能简介	C-1
附录 D 组态王 7.0 新授权使用说明	D-1

第一章 欢迎使用组态王 7.0

- ☞ 介绍组态王 7.0 软件运行的软硬件环境
- ☞ 教您如何安装及启动组态王 7.0 软件及驱动
- ☞ 教您卸载主程序及驱动程序
- ☞ 特别说明
- ☞ 如何得到我们的帮助

1.1 组态王 7.0 软件运行的软硬件环境

- ◇ CPU: P4 处理器、1GHz 以上或相当型号;
- ◇ 内存: 最少 512MB, 推荐 1G, 使用 WEB 功能或 2000 点以上推荐 2G 以上;
- ◇ 显示器: VGA、SVGA 或支持桌面操作系统的任何图形适配器。要求最少显示 256 色;
- ◇ 鼠标: 任何 PC 兼容鼠标;
- ◇ 通讯: RS-232C、100M 及以上以太网;
- ◇ 并行口或 USB 口: 用于接入组态王 7.0 加密锁;
- ◇ 操作系统: Windows XP (sp3) /Win7;



注意:

目前市面上流行的机型完全满足“组态王 7.0”的运行要求。

1.2 安装组态王 7.0

“组态王 7.0”软件存于一张光盘上。安装时插入组态王 7.0 安装光盘，光盘中的安装程序 Install.exe 会自动运行，此时出现“组态王 7.0 安装过程向导”。以 Win7 操作系统下的安装为例，介绍具体的安装步骤：

- ◇ 第一步：启动 Win7 的计算机系统。
- ◇ 第二步：在光盘驱动器中插入“组态王 7.0”软件安装盘，系统自动启动 Install.exe 安装程序，如图 1.1 所示。（用户也可通过光盘中的 Install.exe 启动安装程序）



图 1.1 启动组态王 7.0 安装程序

该安装界面显示如下几项需要用户根据安装需要来操作：

- “安装阅读”项：点击该项，用户可以在安装前阅读关于组态王 7.0 版本更新信息、授权信息、服务和支持信息等。

- “安装组态王程序”项：点击该项，进入组态王 7.0 主程序安装向导。
- “安装组态王驱动程序”项：点击该项，进入组态王 7.0 IO 设备驱动程序安装向导。
- “安装 KV6.x 系列加密锁驱动”项：点击该项，进入组态王 6.5x 加密锁驱动程序安装向导。
- “安装组态王 7.0 加密锁驱动”项：点击该项，进入组态王 7.0 加密锁驱动安装向导。
- “退出”项：点击该“退出”项，退出安装程序。



注意：

Windows XP 操作系统下的安装和 Win7 操作系统下的安装无任何差别。

1.2.1 主程序安装

1、 向导 1_ 开始安装主程序

在“安装组态王 7.0”界面点击“安装组态王程序”项，弹出“向导 1”界面，如 1.2 所示：



图 1.2 开始安装组态王 7.0

该界面为组态王 7.0 主程序的开始安装界面，通过该界面用户可以查看组态王 7.0 的版本信息。

2、向导 2_许可证协议浏览

继续安装点击“下一步”按钮，弹出“向导 2”界面，如图 1.3 所示。

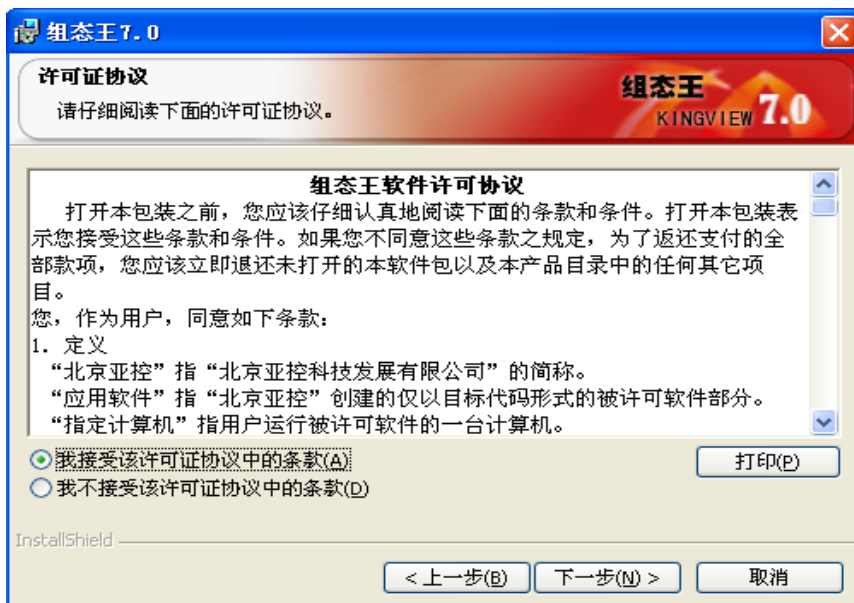


图 1.3 软件许可证协议

该界面的内容为“北京亚控科技发展有限公司”与“组态王 7.0”软件用户之间的法律约定，请用户认真阅读。如果想要继续安装，选择“我接受该许可证协议中的条款”，点击下一步按钮。

3、 向导 3_填写注册信息

当点击“下一步”按钮，弹出“向导 3”界面，如图 1.4 所示。



图 1.4 填入用户信息

通过该界面输入“用户名”和“公司名称”信息。

4、 向导 4_选择安装路径

用户信息填写完成后，点击“下一步”按钮，弹出“向导 4”界面，通过该界面选择程序的安装路径，如图 1.5 所示。

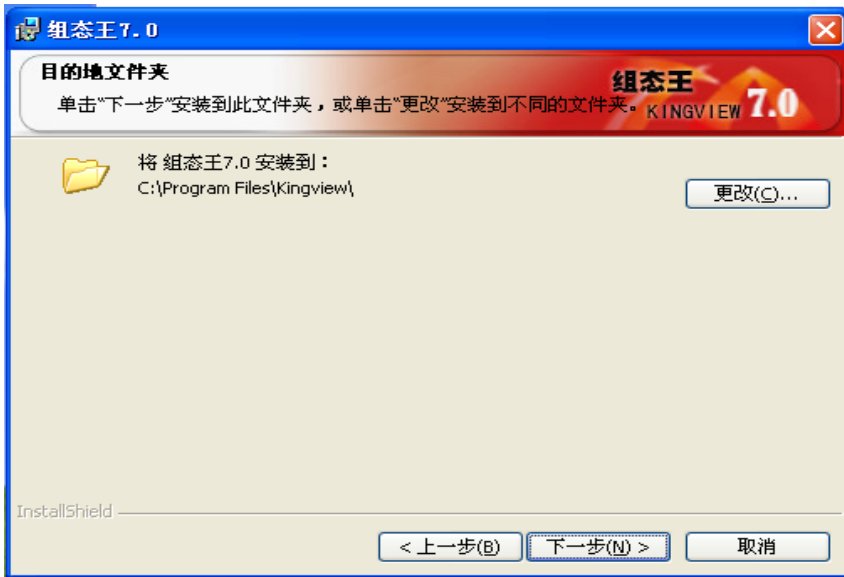


图 1.5 选择组态王 7.0 系统安装路径

由该界面设置“组态王 7.0”软件的安装目录。默认目录为 c:\ProgramFiles\kingview，若希望安装到其它目录，请单击“更改”按钮，弹出如图 1.6 所示对话框。



图 1.6 选择文件夹

在该对话框的“路径”文本框中输入或选择新的安装目录。单击“确定”按钮。返回向导 4 界面。

5、向导 5_自定义安装

单击“下一步”按钮。出现如图 1.7 所示对话框。

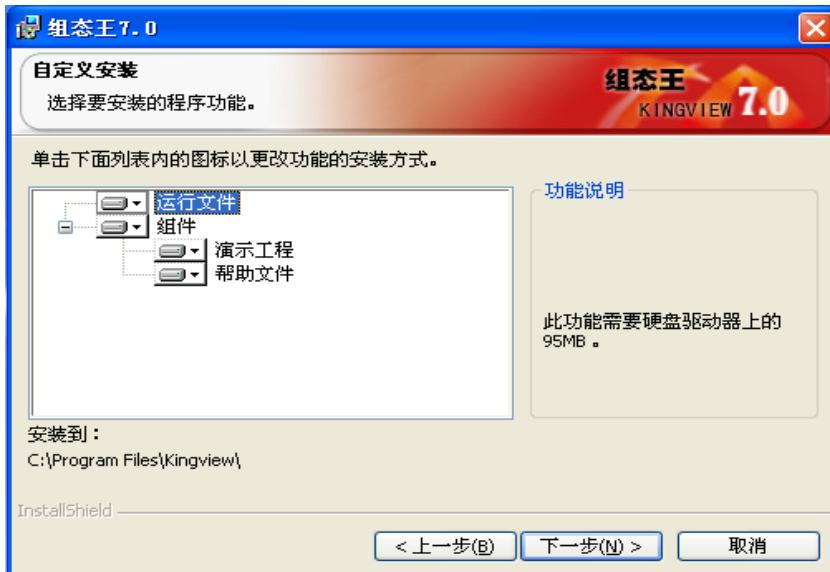


图 1.7 自定义安装

自定义安装将按用户要求安装组件。

6、向导 6_准备安装

继续安装点击“下一步”，弹出“向导 6”界面，如图 1.8 所示对话框：

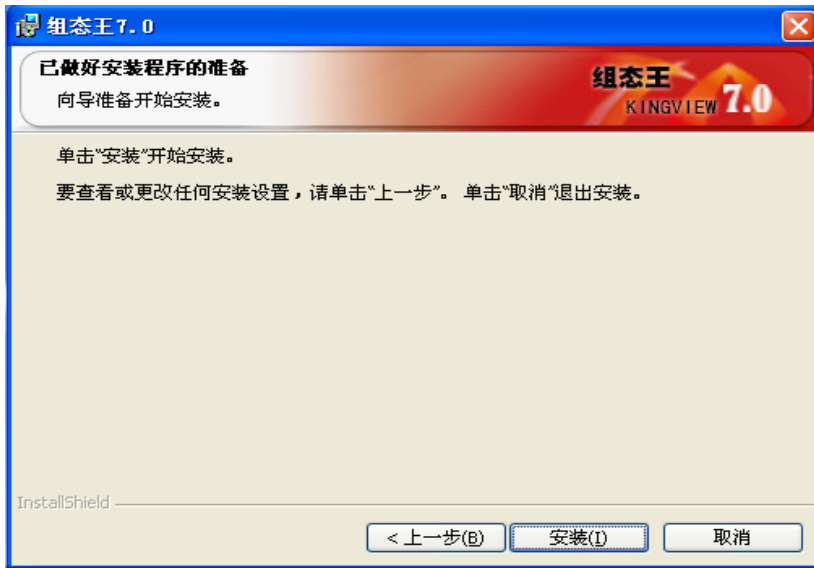


图 1.8 准备安装

7、 向导 7_执行安装

安装程序过程中有显示进度提示。



图 1.9 执行安装

8、向导 8_完成主程序安装

安装结束，弹出“向导 8”界面，如图 1.10 所示：



图 1.10 安装结束

1.2.2 设备驱动程序安装

1、 向导 1_开始安装设备驱动

在“安装组态王 7.0”界面点击“安装组态王驱动程序”项，首先弹出“向导 1”界面，如图 1.11 所示：



图1.11 驱动程序开始安装

该界面为组态王 7.0 驱动程序的开始安装界面。

2、向导 2_许可证协议浏览

继续安装点击“下一步”按钮，弹出“向导 2”界面，如图 1.12 所示。

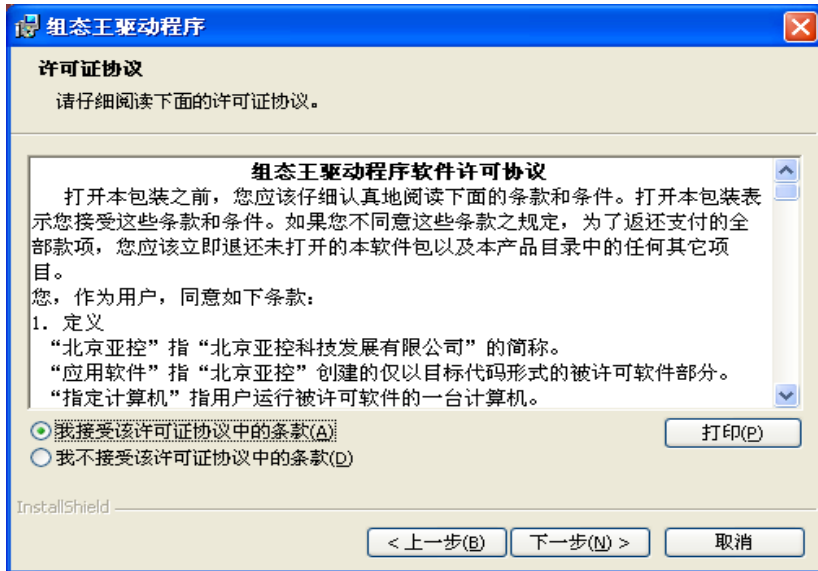


图 1.12 驱动程序软件许可协议

该界面的内容为“北京亚控科技发展有限公司”与“组态王 7.0”软件用户之间的法律约定，请用户认真阅读。

3、向导 3_填写注册信息

当点击“下一步”按钮，弹出“向导 3”界面，如图 1.13 所示。

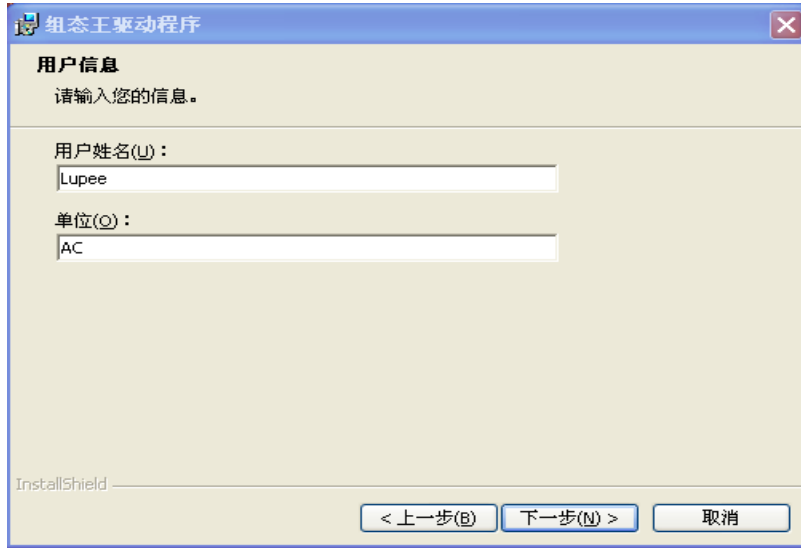


图 1.13 填写用户信息

通过该界面输入“用户名”和“公司名称”信息。

4、向导 4_选择安装路径

单击“下一步”，将出现“向导 4”界面，如图 1.14 所示：

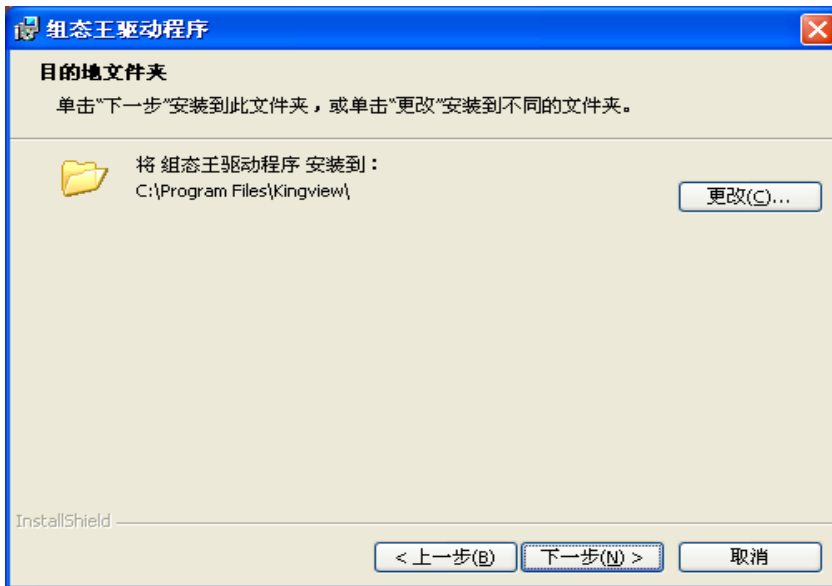


图1.14 创建路径

通过该界面，确认“组态王 7.0”系统的安装目录。系统会自动按照组态王 7.0 的安装路径列出设备驱动程序需要安装的路径。一般情况下，用户无须更改该路径。若希望更改路径，请单击“更改”，弹出对话框，如图 1.15 所示：

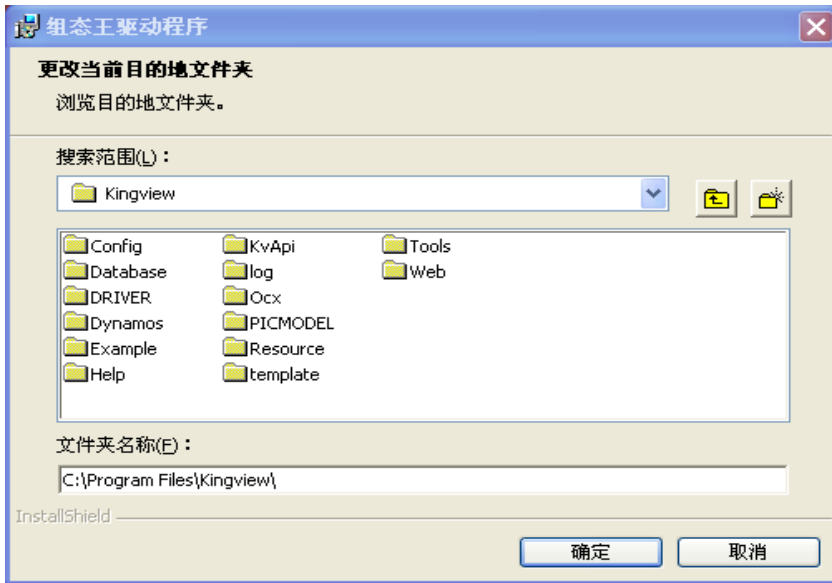


图1.15 更改路径

在对话框的“路径”中输入新的安装目录。如：c:\Program Files\Kingview 输入正确后，单击“确定”按钮。返回“向导 4”界面，此时安装目录为用户输入或者选择的新目录，如下图所示。

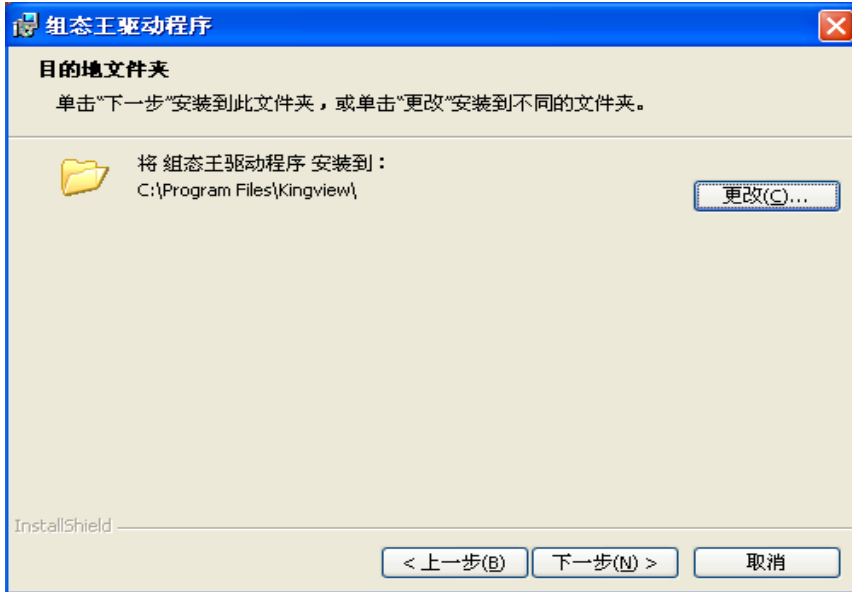


图1.16 确定路径

5、向导 5_自定义安装

继续安装点击“下一步”按钮。弹出如下对话框：

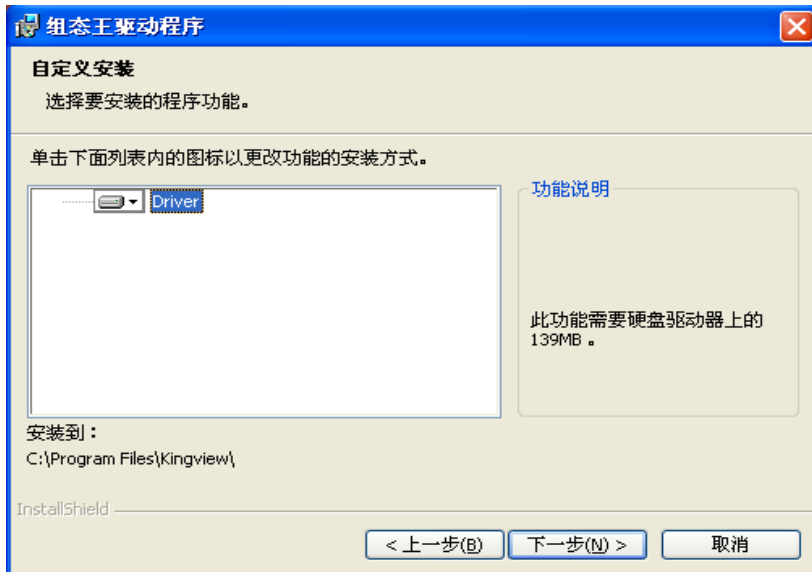


图1.17 自定义安装

用户可以根据自身的需要，选择安装设备驱动。默认状态下，安装全部驱动程序。

6、向导 6_准备安装

点击“下一步”进入准备安装界面，如 1.18 所示：

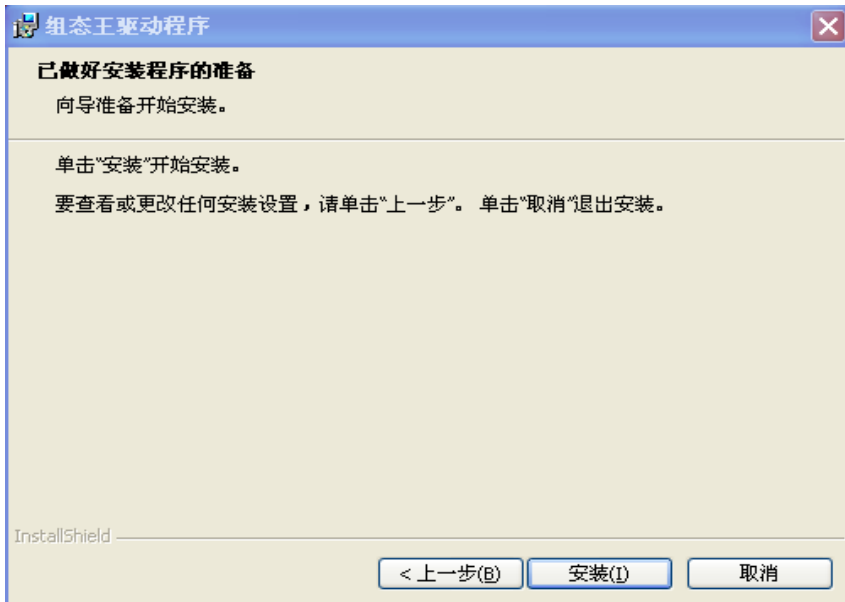


图1.18 准备安装

7、 向导 7_执行安装

执行安装点击“安装”按钮，出现如图 1.19 所示对话框：



图 1.23 驱动程序设置汇总

安装程序过程中有显示进度提示。

8、向导 8_完成驱动程序安装

安装结束，出现如下对话框，如图 1.20 所示：



图 1.20 完成安装

单击完成按钮，此次设备驱动程序的安装结束。

⚠ 注意：

为了使系统能够更好的正常运行，这里我们建议您最好选择重新启动计算机。

1.2.3 加密锁驱动程序的安装

1.2.3.1 KV6.5X 加密锁驱动安装

KV6.5X 对应的硬件加密锁，需要安装其驱动程序，安装加密锁驱动程序的优越性：

- ◇ 与打印机有更好的兼容性；
- ◇ 对加密锁有更好的保护作用；

KV6.5X 对应的硬件加密锁驱动程序在安装盘“KeyDriver\Sentinel”文件夹下。具体安装过程如下：

在“安装组态王 7.0”界面点击“安装 KV6 加密锁驱动程序”项，弹出“向导 1”界面，如图 1.21 所示：

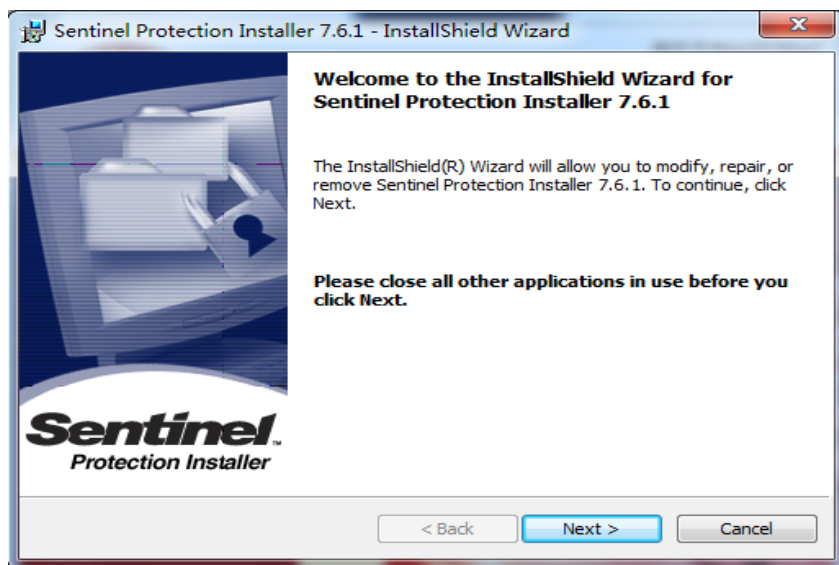


图 1.21 硬件加密锁驱动安装

根据硬件加密锁驱动安装向导安装加密锁驱动，方法与普通软件安装方法相同。驱动程序成功安装后，将包装盒中的加密锁取出，插到计算机的并口上，固定好锁上的螺丝。若需要用打印机，只需将打印机电缆接到加密锁上。加密锁的存在，将不会影响打印机的

使用，若出现问题，请与亚控公司技术支持部联系。

如果您使用的是 USB 接口加密锁，将加密锁直接插入计算机的 USB 接口即可，该型号加密锁支持即插即用。

1.2.3.2 Kv7.0 加密锁驱动安装

Kv7.0 加密锁驱动程序在安装盘“KeyDriver\LicenseDriver”文件夹下，具体安装过程如下：

点击“Kv7.0 加密锁驱动安装”项，弹出“向导 1”界面，如下图所示：

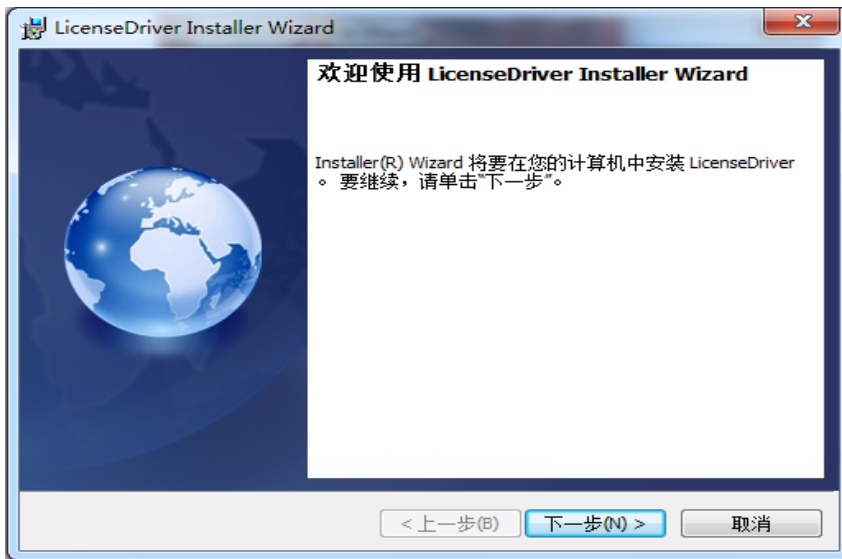


图1.22 kv7 加密锁驱动安装

点击“下一步”按钮，弹出“向导 2”界面，如下图所示：

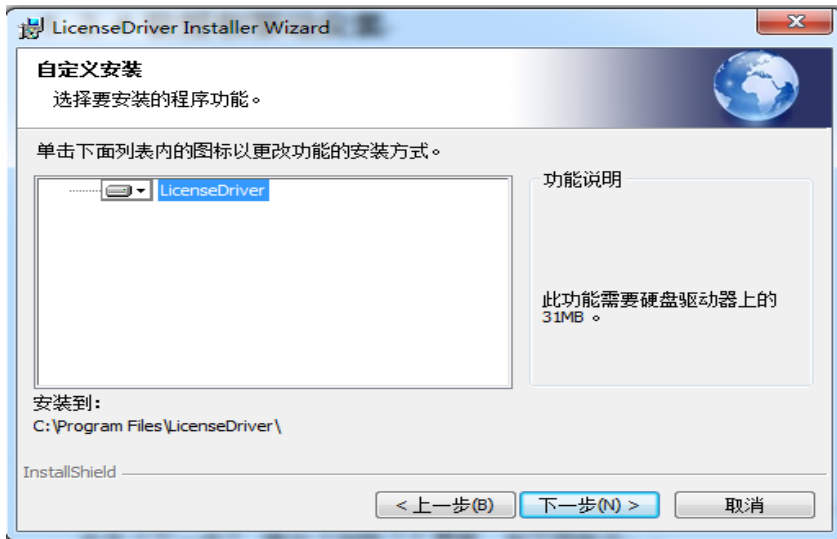


图1.23 自定义安装

点击“下一步”按钮，弹出“向导 3”界面，如下图所示：

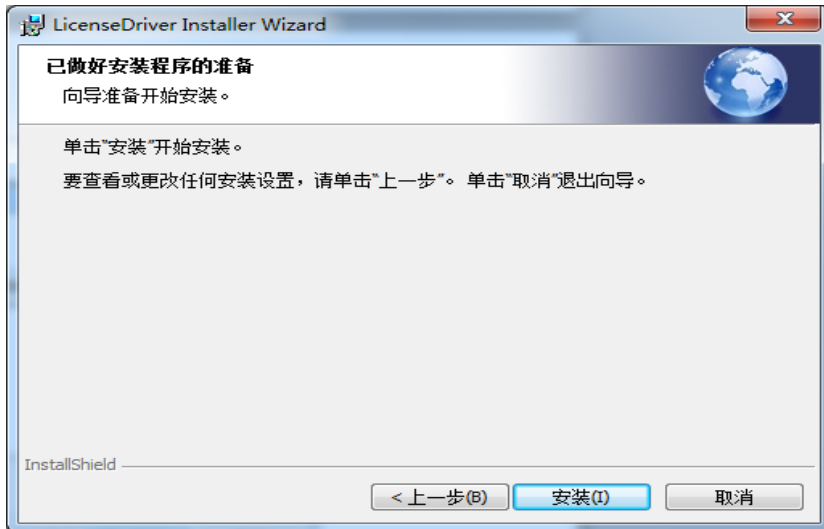


图1.24 准备安装

点击“安装”按钮，执行安装。

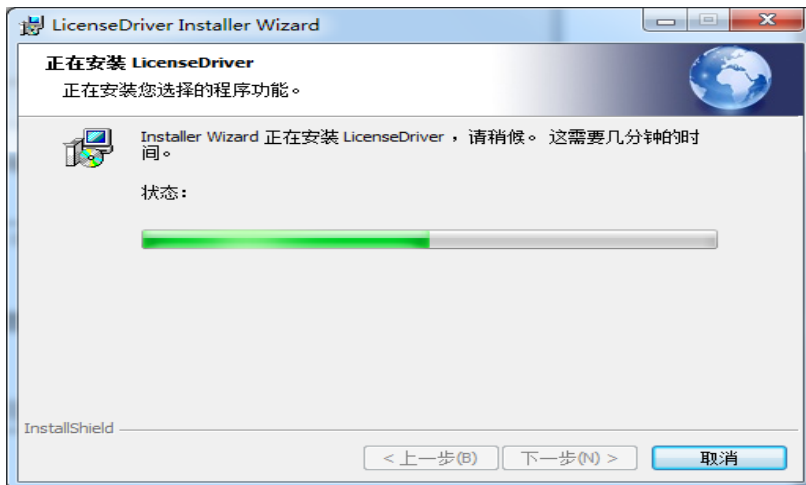


图1.25 执行安装

安装完成后，点击“完成”按钮，则退出当前安装。



图1.26 完成安装

 注意：

组态王 7.0 加密锁的使用说明参见帮助文档附录。

1.3 卸载组态王 7.0 主程序、驱动程序

卸载组态王 7.0 和设备驱动程序请按下列步骤：

- ✧ **第一步：**选择 windows 的“开始菜单”，选择“设置”－“控制面板”，进入“控制面板”界面，选择“添加/删除程序”如图 1.27 所示：



图 1.27 删除程序

- ✧ **第二步：**如果安装有组态王 7.0 驱动程序，在卸载组态王 7.0 时必须先卸载组态王 7.0 驱动程序。在上图中选择组态王 7.0 驱动程序，单击“更改/删除”按钮，系统弹出如图 1.28 所示对话框。选择“除去”选项，系统会自动删除安装的项目。

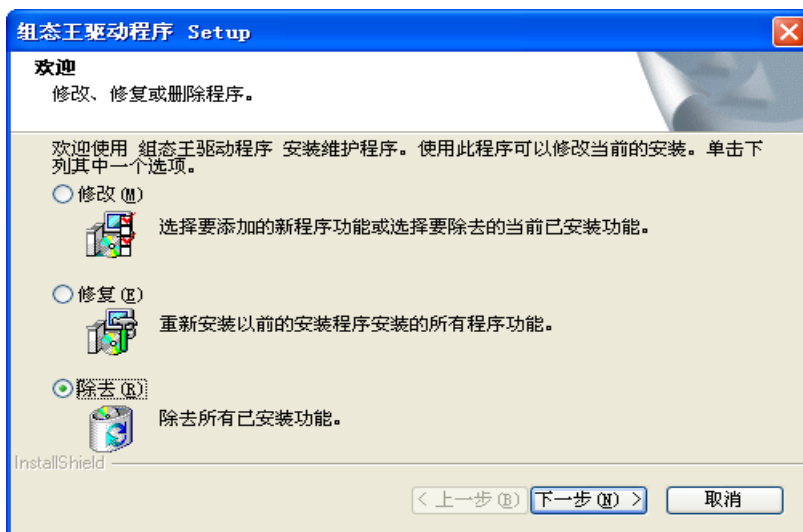



图 1.28 删除程序

 注意：

组态王 7.0 进行删除后，建议用户重新启动计算机，以保证系统删除程序完整。

1.4 修复或修改安装组件

在装有组态王 7.0 的计算机上，在控制面板中执行删除软件操作，或是使用组态王 7.0 安装程序重新安装时，安装程序自动提示“修改、修复或删除”对话框，如图 1.28 所示。如选择修改组态王 7.0 驱动程序，点击“下一步”按钮，弹出驱动组件修改对话框。可以通过选择来修改安装内容。

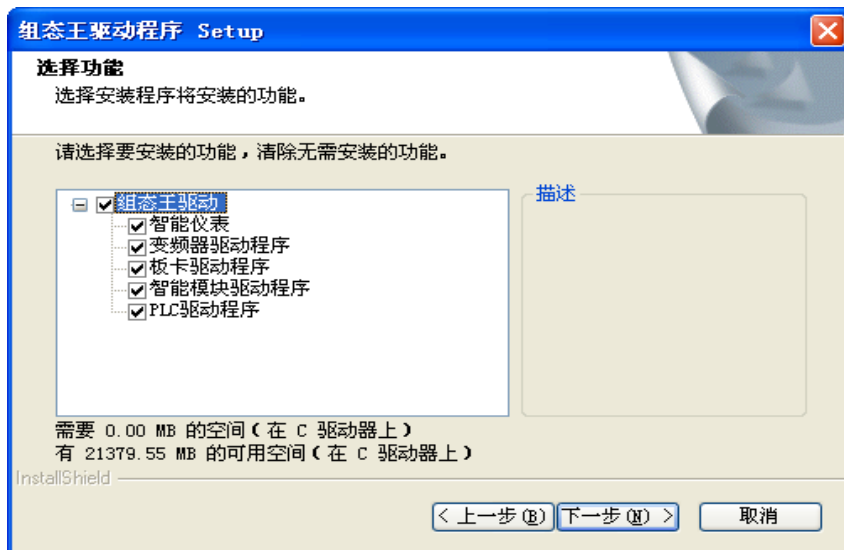


图1.29 修改程序

1.5 加密锁驱动升级

1、KV6.5X 加密锁升级

kv6.5x 硬件加密锁的升级工具在安装盘“KeyDriver\加密锁升级工具”文件夹下，具体升级过程参见《用户加密锁升级步骤.PDF》。

2、KV7.0 加密锁升级

Kv7.0 硬锁和软锁的升级工具在“KeyDriver\LicenseDriver\program files\LicenseDriver\RUS_OAIKD.exe”，具体升级过程参见《用户加密锁升级步骤.PDF》。

1.6 特别说明

1.6.1 光盘里其它有价值的内容

◇ KeyDriver:

kv6.5 系列硬件加密锁驱动程序安装: KeyDriver\Sentinel 文件夹

kv6.5 系列加密锁升级工具: KeyDriver\加密锁升级工具\工具

Kv7 系列硬件加密锁驱动程序安装: KeyDriver\LicenseDriver

Kv7 系列加密锁升级工具: KeyDriver\LicenseDriver\program files\LicenseDriver

◇ 7.0 升级工具:

工程升级工具: 参见 7.0 升级工具 \program files\Kingview 下的 UpgradeKvVersion.exe。

工程升级帮助说明: 参见 7.0 升级工具 \program files\Kingview 下的工程升级工具使用说明.pdf。

1.6.2 组态王 7.0 版本类型

◇ 试用版

◇ 单机限时版

◇ 单机不限时版

◇ 网络限时版

◇ 网络不限时版

1.6.3 组态王 7.0 加密锁类型

- ◇ 按软件保护方式分，有硬件授权和软授权：
 - 硬件授权：主要是指加密锁形式的授权，包括并口锁和 USB 锁。
 - 软授权：主要是指软件形式的授权，一般采用与计算机硬件特征绑定的电子许可证形式的加密技术，它依然需使用加密算法作为主要的保护软件的技术手段。
- ◇ 按照加密锁功能本身还分为单机锁与网络锁：
 - 单机锁：只能够同时在一台计算机上使用。
 - 网络锁：授权用户数量，可以允许网络上面的其他计算机使用，同时使用的用户数量不能够超过锁的授权用户数量。
- ◇ 按照组态王使用时限分为试用版授权和正式版授权，其中正式版授权又分为限时授权和无限时授权。
 - 试用授权：针对亚控公司发行的软件试用版本，一次性制作完成含固定期限的授权文件（demo*.V2C），并将其放入产品安装包中提供给用户试用，产品软件在试用期间、“试用参数范围”内正常运行，有试用提示，在试用期间过后不能正常运行；
 - 限时授权：针对亚控公司发行的软件正式版本，根据用户需求制作授权文件，并将其提供给授权用户使用，产品软件在限时期间、“授权参数范围”内正常运行，有限时提示，在限时期间过后不能正常运行；
 - 无限时授权：针对亚控公司发行的软件正式版本，根据用户需求制作授权文件，

并将其提供给授权用户使用，产品软件在“授权参数范围”内正常运行。

1.7 如何得到我们的帮助

您在使用组态王 7.0 软件或亚控其它产品的过程中，如果遇到疑问或困难，或需要进行咨询，请与亚控公司技术支持工程师、销售工程师联系。您可以通过电话、邮件、传真、网站（技术 BBS）等方式与我们取得联系，我们会在第一时间给您满意的答复。亚控公司在上海、广州设有分公司，在济南、西安、成都、武汉、南京、乌鲁木齐、哈尔滨等地设有办事处。

下面是亚控公司总部的联系方式，如有变更，请联系亚控总部或亚控网站，恕不另行通知。

亚控科技北京总部：

- ☞ 地址：北京市海淀区知春路 113 号银网中心 A 座 6 层
- ☞ 邮编：100086
- ☞ 电话：010-59309666 | 400 全国免费电话：400-898-5151
- ☞ 传真：010-59309600
- ☞ E-mail: support@wellintech.com
sales@wellintech.com
- ☞ 网站: <http://www.wellintech.com>
<http://en.wellintech.com>

第二章 组态王 7.0 使用入门

- ☞ 介绍组态王 7.0 软件的基本构成
- ☞ 教您如何组态一个组态王 7.0 工程
- ☞ 介绍组态王 7.0 软件的升级和动态分辨率转换

2.1 软件基本构成

“组态王 7.0”基本的程序包括：工程管理器、工程浏览器、运行系统、信息窗口。当安装完“组态王 7.0”后，会在桌面上自动生成“组态王 7.0”工程管理器的快捷方式，同时在系统的开始\程序菜单下生成名称为“组态王 7.0”的程序组，该程序组中包含内容如下：

- ◇ **工程管理器：**组态王 7.0 工程管理器程序（ProjManager）的快捷方式，用于新建工程、工程管理等；
- ◇ **工程浏览器：**组态王 7.0 单个工程管理程序的快捷方式，内嵌组态王 7.0 画面开发系统（TouchExplorer），即组态王 7.0 开发系统；
- ◇ **运行系统：**组态王 7.0 运行系统程序（TouchView）的快捷方式。工程浏览器（TouchExplorer）和运行系统（TouchView）是各自独立的 Windows 应用程序，均可单独使用；两者又相互依存，在工程浏览器的画面开发系统中设计开发的画面应用程序必须在画面运行系统（TouchView）运行环境中才能运行；

- ◇ **信息窗口：**组态王 7.0 信息窗口程序（KingMess）的快捷方式；
- ◇ **组态王 7.0 卸载：**组态王 7.0 卸载的快捷方式；
- ◇ **组态王驱动卸载：**组态王 6.5x 驱动卸载的快捷方式；
- ◇ **工具\驱动安装工具：**安装新驱动工具文件的快捷方式；
- ◇ **工具\服务注册工具：**注册服务工具文件的快捷方式；
- ◇ **工具\web 发布工具：**安装 web 发布工具文件的快捷方式；
- ◇ **工具\LicenseViewer_CN：**安装新授权查看工具文件的快捷方式；
- ◇ **帮助文档\组态王帮助：**组态王 7.0 帮助文件快捷方式；
- ◇ **帮助文档\驱动帮助：**组态王 7.0 驱动程序帮助文件快捷方式；
- ◇ **帮助文档\命令语言函数手册：**组态王 7.0 函数手册电子版文件快捷方式；
- ◇ **帮助文档\组态王 7.0 使用手册：**组态王 7.0 使用手册电子版文件快捷方式；

2.2 工程组态

工程组态人员当组态一个工程前，首先需要考虑如下三个方面：

- ◇ **图形：**用户希望用怎样的图形画面来模拟实际工业现场和相应的工控设备。
- ◇ **数据：**怎样用数据描述实际设备对象的各种属性？也就是创建一个具体的数据库，此数据库中的变量反映工控对象的各种属性，比如温度，压力等。
- ◇ **连接：**数据和图形画面中的图素的连接关系是什么？也就是画面上的图素以怎样的动画模拟现场设备运行，以及操作者如何对控制设备输入指令。

通过上述疑问，工程组态人员可以按“创建组态王 7.0 工程”、“定义设备”、“数据

库建点”、“创建组态画面”、“建立动画连接”及“运行和调试”的步骤来组态工程：

2.2.1 创建组态王 7.0 工程

要建立新的组态王 7.0 工程，首先为工程指定“工作目录”（“工程路径”）。“组态王 7.0”用“工作目录”标识工程，不同的工程应置于不同的“工作目录”。“工作目录”下的文件由“组态王 7.0”自动管理。

1、向导 1_欢迎使用本向导

启动“组态王 7.0”工程管理器 (ProjManager)，点击菜单栏的“文件\新建工程”或者直接单击“新建”按钮，弹出“向导 1”界面，如图 2.1 所示。



图 2.1 新建工程向导一

2、向导 2_选择工程路径

单击“下一步”继续。弹出“向导 2”界面，如图 2.2 所示。



图 2.2 新建工程向导二

直接在工程路径文本框输入一个有效的工程路径，或单击“浏览…”按钮，在弹出的路径选择对话框中选择一个有效的路径。单击“下一步”继续。弹出“向导 3”，如图 2.3 所示。

3、向导 3_工程名称和描述

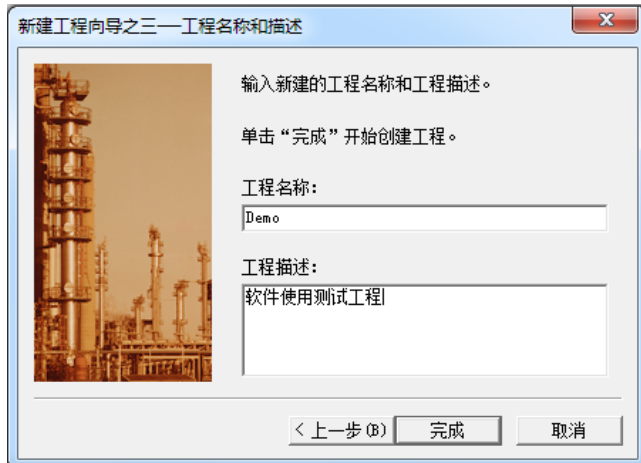


图 2.3 新建工程向导三

- ◇ **工程名称**文本框：输入工程的名称（如：Demo），该工程名称同时将被作为当前工程的路径名称，工程名称长度应小于 32 个字符。
- ◇ **工程描述**文本框：输入对该工程的描述文字（如：软件使用测试工程），工程描述长度应小于 40 个字符。
- ◇ **完成**按钮：完成工程的新建。系统会弹出对话框，询问用户是否将新建工程设为当前工程，如图 2.4 所示。

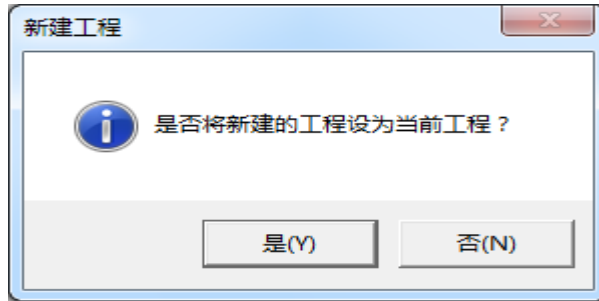


图 2.4 是否设为当前工程对话框

- **否按钮**：单击该按钮则新建工程不是工程管理器的当前工程，如果要将该工程设为新建工程，还要执行菜单下的“文件\设为当前工程”命令；
- **是按钮**：单击该按钮则将新建的工程设为“组态王 7.0”的当前工程。定义的工程信息会出现在工程管理器的信息表格中。

双击该信息条或单击“开发”按钮或选择菜单“工具\切换到开发系统”，进入组态王 7.0 的开发系统。建立的工程路径为：`C:\Users\Administrator\Desktop\demo`（组态王 7.0 画面开发系统为此工程建立目录 `C:\Users\Administrator\Desktop\demo`，并生成必要的初始数据文件。这些文件对不同的工程是不相同的。因此，不同的工程应该分置不同的目录。这些数据文件列在附录 AX 中）。具体工程管理器的使用方法，请参见“第三章 启动组态王 7.0 - 工程管理器”一章内容。

 **注意：**

建立的每个工程必须在单独的目录中。除非特别说明，不允许编辑修改这些初始数据文件。

2.2.2 定义 I/O 设备

组态王 7.0 把那些需要与之交换数据的设备或程序都作为外部设备。外部设备包括：下位机（PLC、仪表、模块、板卡、变频器等），它们一般通过串行口和上位机交换数据；其他 Windows 应用程序，它们之间一般通过 DDE 交换数据；外部设备还包括网络上的其他计算机。

只有在定义了外部设备之后，组态王 7.0 才能通过 I/O 变量和它们交换数据。为方便定义外部设备，组态王 7.0 设计了“设备配置向导”引导用户逐步完成设备的连接。

本例中使用仿真 PLC 和组态王 7.0 通信，仿真 PLC 可以模拟 PLC 为组态王 7.0 提供数据，假设仿真 PLC 连接在计算机的 COM1 口：

1、向导 1_选择设备名称及通讯方式

选择工程浏览器左侧大纲项“设备\COM1”，在工程浏览器右侧用鼠标左键双击“新建”图标，运行“设备配置向导”，如图 2.8 所示。



图 2.8 设备配置向导一

2、向导 2_为设备设置逻辑名称

选择“仿真 PLC”的“COM”项，单击“下一步”，弹出“设备配置向导”，如图 2.9 所示。



图 2.9 设备配置向导二

3、向导 3_选择 Com 口

为外部设备取一个名称，输入 PLC，单击“下一步”，弹出“设备配置向导”，如图 2.10 所示。



图 2.10 设备配置向导三

为设备选择连接串口，假设为 COM1，单击“下一步”，弹出“设备配置向导”，如图 2.11 所示。

4、向导 4_设置设备地址

填写设备地址，假设设备地址为 0。



图 2.11 设备配置向导四

5、向导 5_设置通信故障参数

单击“下一步”，弹出“设备配置向导”，如图 2.12 所示。



图 2.12 设备配置向导五

设置通信故障恢复参数(一般情况下使用系统默认设置即可), 单击“下一步”, 弹出“设备配置向导”, 如图 2.13 所示。

6、向导 6_信息总结



图 2.13 设备配置向导六

请检查各项设置是否正确，确认无误后，单击“完成”。设备定义完成后，可以在工程浏览器的右侧看到新建的外部设备“PLC”。在定义数据库变量时，只要把 I/O 变量连接到这台设备上，它就可以和组态王 7.0 交换数据了。

⚠ 注意：以仿真 PLC 和组态王 7.0 通信为例弹出上述向导界面。如果采用其它通讯方式，则向导界面会有所不同，此处不作详细介绍，可参见“第六章 I/O 设备管理”

单击工具栏中的“VIEW”按钮，把系统切换到运行状态。如果提示加载协议组件失败，如下图所示，则在信息窗口中提示出具体是哪个协议组件失败，以使用户定位和解决问题。



- 如果加载驱动组件失败，在“信息窗口”中得到如图提示信息。

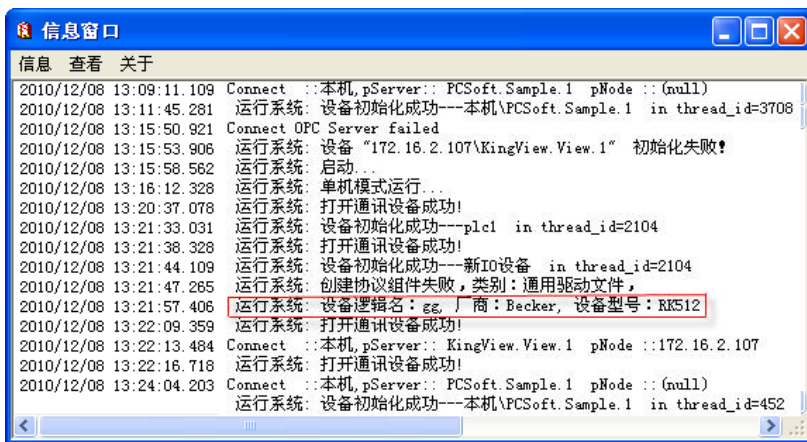


图 2.14 协议组创建失败信息一

- 如果是 OCX 控件，则在“信息窗口”列出加载失败的控件的名称。如图

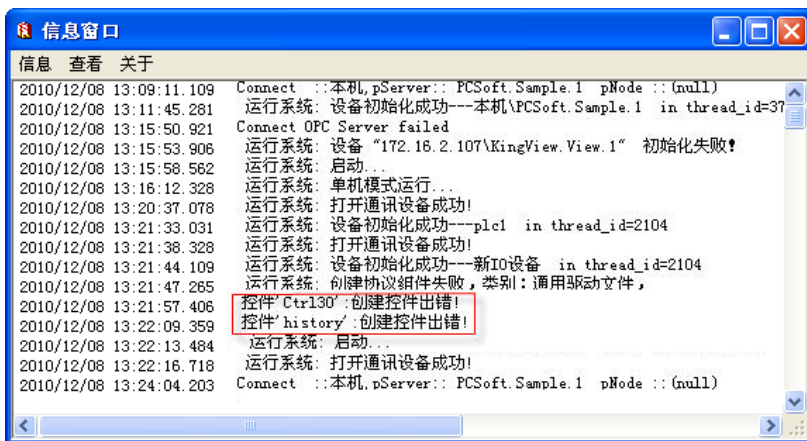


图 2.15 协议组创建失败信息二

2.2.4 数据库建点

数据库是“组态王 7.0”软件的核心部分，工业现场的生产状况要以动画的形式反映在屏幕上，操作者在计算机前发布的指令也要迅速送达生产现场，所有这一切都是以实时数据库为中介环节，所以说数据库是联系上位机和下位机的桥梁。在 TouchView 运行时，它含有全部数据变量的当前值。变量在画面制作系统组态王 7.0 画面开发系统中定义，定义时要指定变量名和变量类型，某些类型的变量还需要一些附加信息。数据库中变量的集合形象地称为“数据词典”，数据词典记录了所有用户可使用的数据变量的详细信息。



例如：

选择工程浏览器左侧大纲项“数据库\数据词典”，在工程浏览器右侧用鼠标左键双击“新建”图标，弹出“变量属性”对话框如图 2.16 所示。

1、定义内存变量

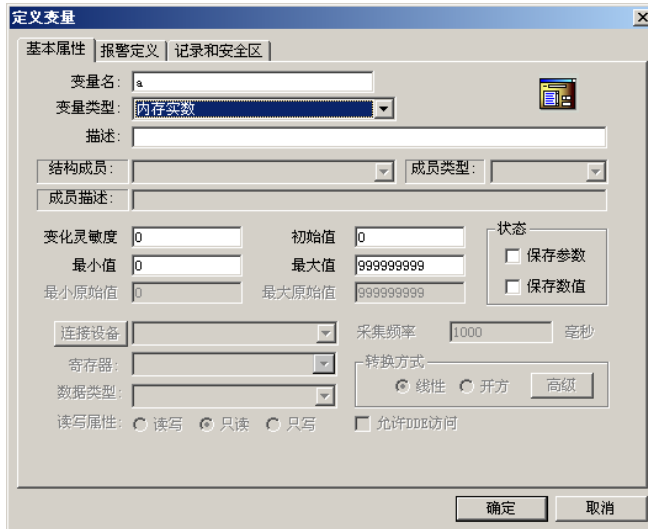


图 2.16 创建内存变量

此对话框可以对数据变量完成定义、修改等操作，以及数据库的管理工作，详细变量操作请参见“第五章 变量定义和管理”。

- ◇ **变量名：**输入变量名，如：a；
- ◇ **变量类型：**选择变量类型如：内存实数；
- ◇ **其它属性：**目前不用更改；
- ◇ **确定按钮：**单击“确定”完成属性配置；

2、定义 IO 变量



图 2.17 创建 I/O 变量

- ◇ **变量名:** 输入变量名，如：b；
- ◇ **变量类型:** 选择变量类型如：I/O 整数；
- ◇ **连接设备:** 选择先前定义好的 I/O 设备：PLC；在“寄存器”中定义为：INCREA100；在“数据类型”中定义为：SHORT 类型；
- ◇ **其它属性:** 目前不用更改；
- ◇ **确定按钮:** 单击“确定”完成属性配置；

2.2.2 组态画面

“组态王 7.0”采用面向对象的编程技术，用户可以像搭积木那样利用“组态王 7.0”提供的图形对象完成画面的组态。进入“组态王 7.0”的开发系统后，可以为每个工程建

立多个画面，在每个画面上组态相关联的静态或动态图形。

为方便在画面上组态各种图形，“组态王 7.0”开发系统中提供了矩形（圆角矩形）、直线、椭圆（圆）、扇形（圆弧）、点位图、多边形（多边线）、文本等基本的图形对象以及按钮、趋势曲线窗口、报警窗口、报表等复杂的图形对象。为方便操作这些图形对象，“组态王 7.0”还提供了图形对象在窗口内任意移动、缩放、改变形状、复制、删除和对齐操作、图形对象在画面之间任意拷贝操作、提供键盘、鼠标绘图以及提供改变图形对象的颜色、线型、填充属性的操作工具。

2.2.2.1 新建画面

进入新建的组态王 7.0 工程，选择工程浏览器左侧大纲项“文件\画面”，在工程浏览器右侧用鼠标左键双击“新建”图标，弹出对话框如图 2.5 所示。



图 2.5 新建画面

- ◇ **画面名称**文本框：输入新的画面名称，如 test。
- ◇ **其它属性**：目前不用更改（关于其它属性的设置请参见“第四章 组态王 7.0 开发环境 - 工程浏览器”）。
- ◇ **确定按钮**：点击该按钮进入内嵌组态王 7.0 画面开发系统。如图 2.6 所示。

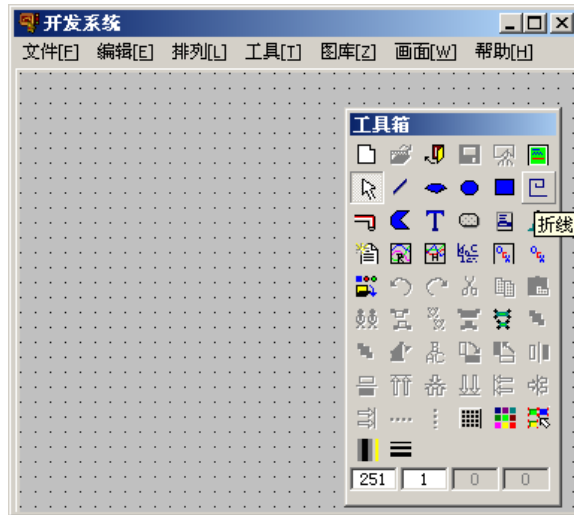


图 2.6 组态王 7.0 开发系统

2.2.2.2 创建图形对象

在“组态王 7.0”开发系统中从“工具箱”中分别选择“矩形”和“文本”图标，绘

制一个矩形对象和一个文本对象，如图 2.7 所示。

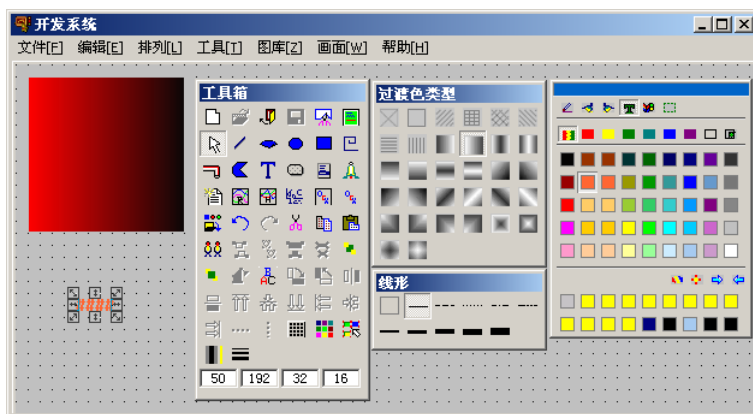


图 2.7 创建图形画面

- ◇ 在工具箱中选中“圆角矩形”，拖动鼠标在画面上画一个矩形，如上图所示。
- ◇ 用鼠标在工具箱中点击“显示画刷类型”和“显示调色板”。在弹出的“过渡色类型”窗口点击第二行第四个过渡色类型；在“调色板”窗口点击第一行第二个“填充色”按钮，从下面的色块中选取红色作为填充色，然后点击第一行第三个“背景色”按钮，从下面的色块中选取黑色作为背景色。此时就构造好了一个使用过渡色填充的矩形图形对象。
- ◇ 在工具箱中选中“文本”，此时鼠标变成“I”形状，在画面上单击鼠标左键，输入“#####”文字。
- ◇ 拖动图形对象的边线可修改大小；若需要移动位置，可以把光标定位在图形对象上，拖动鼠标即可。

2.2.5 建立动画连接

定义动画连接是指在画面的图形对象与数据库的数据变量之间建立一种关系,当变量的值改变时,在画面上以图形对象的动画效果表示出来;或者由软件使用者通过图形对象改变数据变量的值(参见“第七章 动画连接”)。“组态王 7.0”提供了 24 种动画连接方式:

属性变化	线属性变化、填充属性变化、文本色变化
位置与大小变化	填充、缩放、旋转、水平移动、垂直移动
值输出	模拟值输出、离散值输出、字符串输出
值输入	模拟值输入、离散值输入、字符串输入
特殊	闪烁、隐含、流动(仅适用于立体管道)
滑动杆输入	水平、垂直
命令语言	按下时、弹起时、按住时、鼠标进入、鼠标离开

一个图形对象可以同时定义多个连接,组合成复杂的效果,以便满足实际中任意的动画显示需要。



例如:

- 1) **对矩形关联填充动画:** 双击图形对象,即矩形,可弹出“动画连接”对话框,如图 2.18 所示。

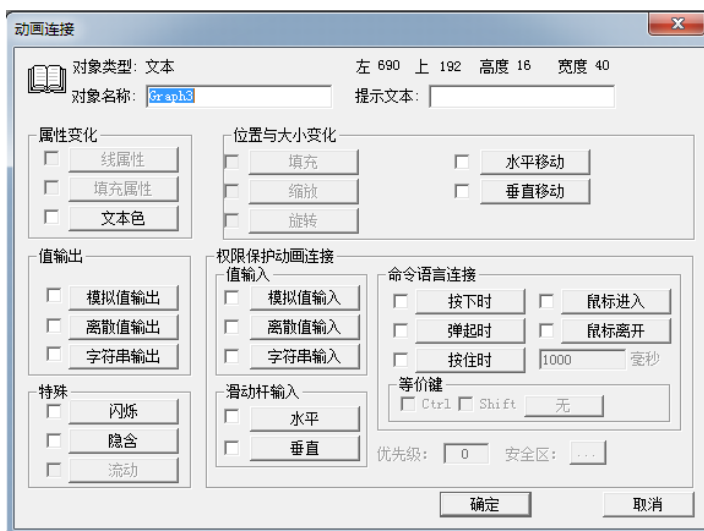


图 2.18 动画连接

各属性的设置将在“第七章 动画连接”一章中详细介绍。用鼠标单击“填充”按钮，弹出对话框如图 2.19 所示。



图 2.19 填充属性

在“表达式”处输入“a”，“缺省填充刷”的颜色改为黄色，其余属性目前不用更改，如图 2.20 所示。



图 2.20 更改填充属性

单击“确定”，再单击“确定”返回组态王 7.0 开发系统。

- 2) **输入命令语言：**为了让矩形动起来，需要使变量即 a 能够动态变化，选择“编辑\画面属性”菜单命令，弹出对话框如图 2.21 所示。



图 2.21 画面属性

单击“命令语言...”按钮，弹出画面命令语言对话框，如图 2.22 所示。

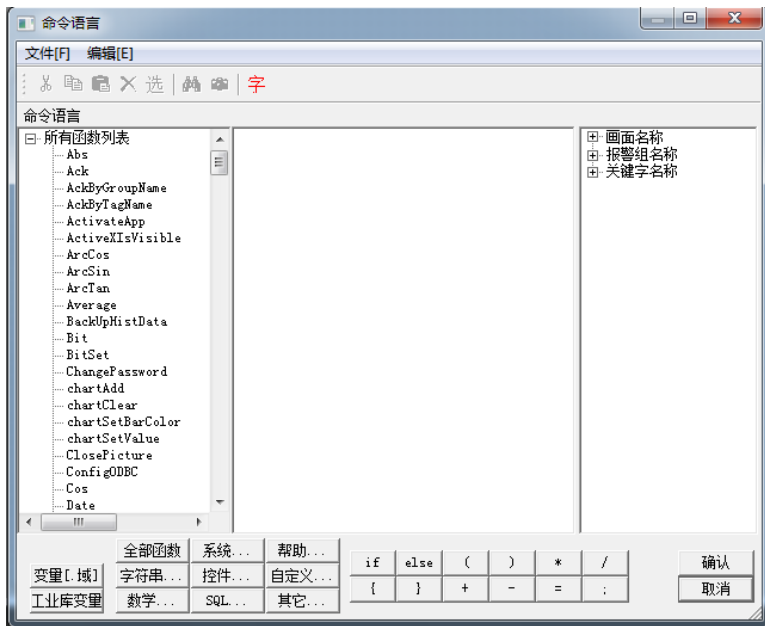


图 2.22 画面命令语言

在编辑框处输入命令语言：

```
if(a<100)
    a=a+10;
else
    a=0;
```

可将“每 3000 毫秒”改为“每 500 毫秒”，此为画面执行命令语言的执行周期。单击“确认”，及“确定”回到开发系统。

- 3) 对文本关联模拟输出动画：双击文本对象“####”，可弹出“动画连接”对话框，如图 2.23 所示。

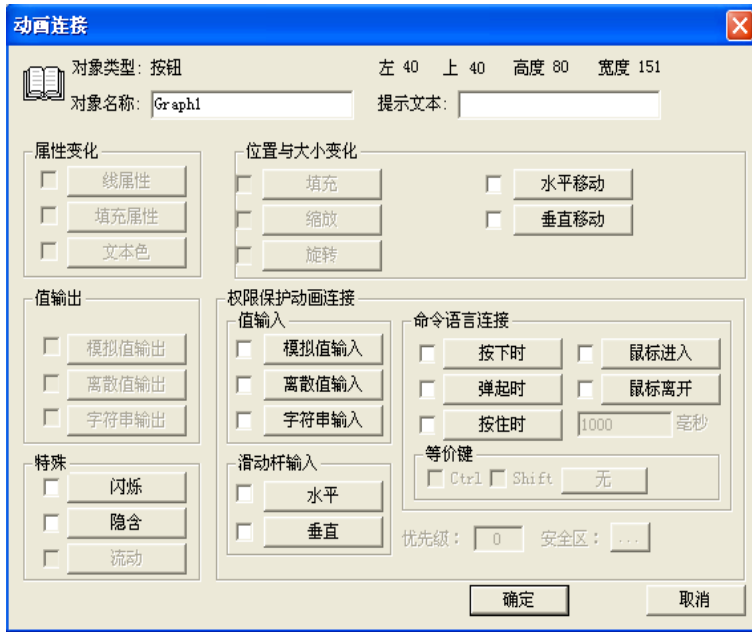


图 2.23 动画连接

用鼠标单击“模拟值输出”按钮，弹出对话框如图 2.24 所示。

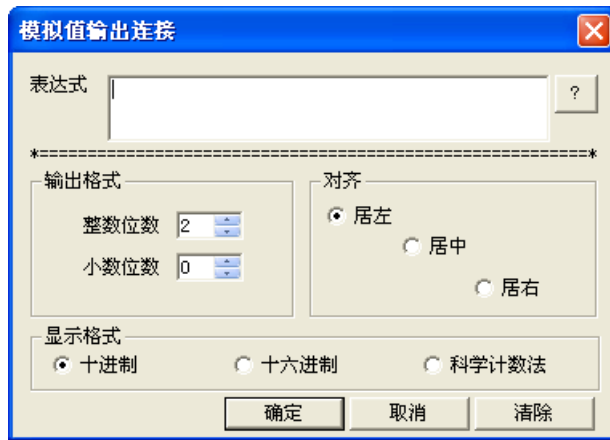


图 2.24 模拟值输出连接

在“表达式”处输入“b”，其余属性目前不用更改。单击“确定”，再单击“确定”返回组态王 7.0 开发系统。


- 4) **保存：**选择“文件\全部存”菜单命令。

2.2.6 运行和调试

组态王 7.0 工程已经初步建立起来，进入到运行和调试阶段。在组态王 7.0 开发系统中选择“文件\切换到 View”菜单命令，进入组态王 7.0 运行系统。在运行系统中选择“画面\打开”命令，从“打开画面”窗口选择“Test”画面。显示出组态王 7.0 运行系统画面，即可看到矩形框和文本在动态变化。如图 2.25 所示。



图 2.25 运行系统画面

 祝贺您已掌握了第一步!

2.3 组态王 7.0 的升级和动态分辨率转换

1、组态王 7.0 升级

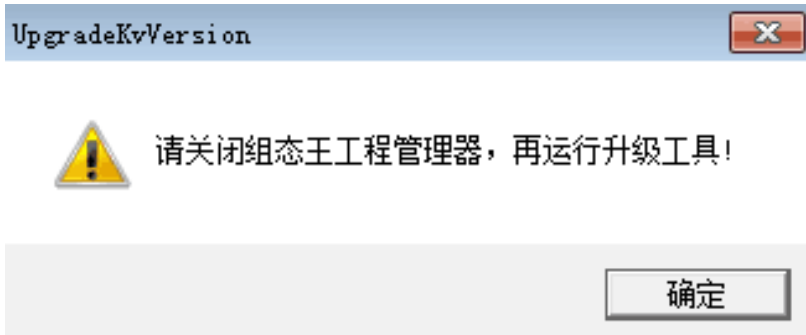
原来 6.53 升级到 6.55 采用的升级方式是在 6.55 下打开低版本工程，进入开发系统后系统将提示用户是否升级，选择“是”系统自动进行升级，这种升级方式下低版本工程一旦升级打开之后，就不能使用低版本软件打开工程，因此用户在升级工程之前需要做好工程备份。

组态王 7.0 的升级借助了工程升级工具，以该升级工具为纽带，工程升级后保证了原来老版本下的工程也不受影响。具体升级过程主要分两部分：1、通过升级工具将 kv6.55 的工程转成中间文件；2、通过 7.0 工程浏览器界面加载该中间文件。

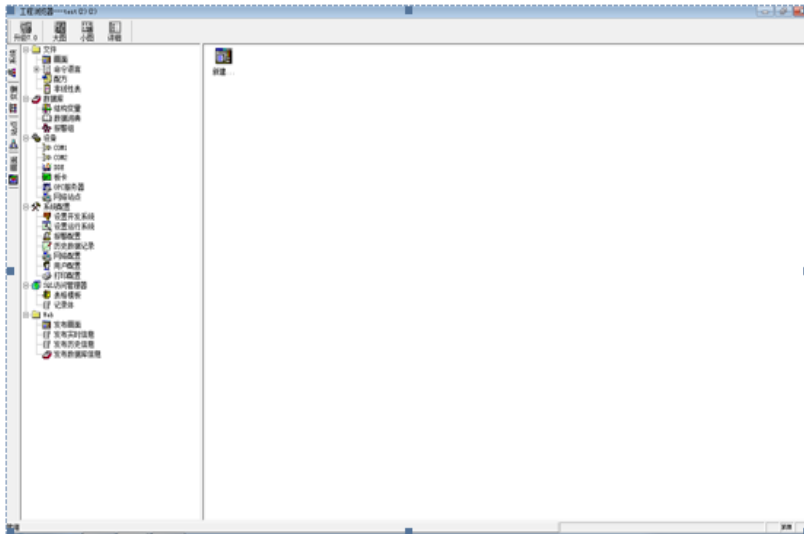
1、通过升级工具将 kv6.55 的工程转成中间文件：

前提准备工作：

- 1)、用户需要在 kv6.55 下安装升级工具，安装包 UpdateTools 由亚控公司提供。
- 2)、首先关闭 655 工程管理器或开发系统，否则会弹出提示框，如下图所示：

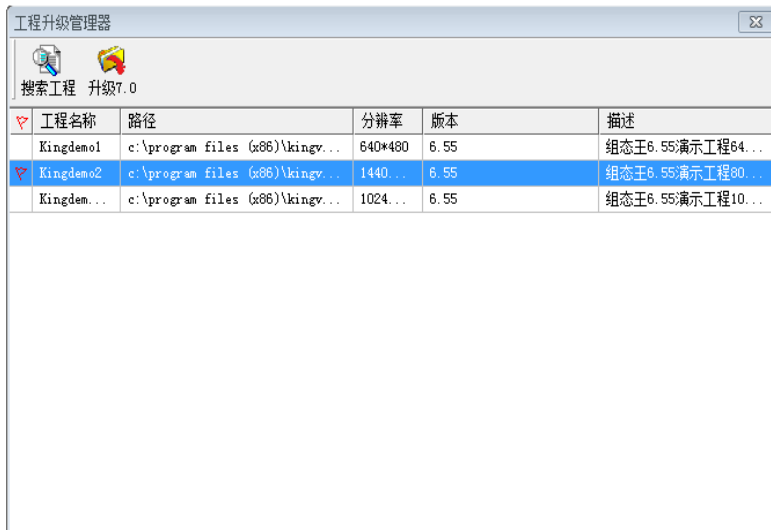


步骤一：双击运行 UpgradeKvVersion.exe, 弹出工程升级对话框，如下图所示：



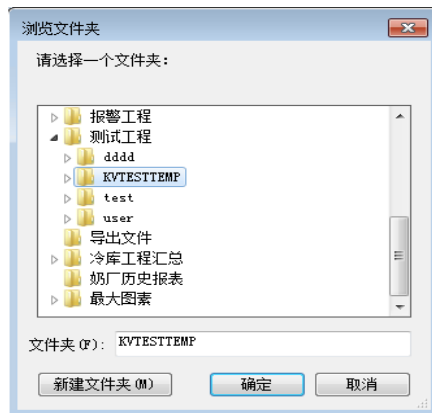
工程升级对话框

步骤二：点击工具栏的“升级 7.0”，弹出工程升级管理器界面，如下图所示：

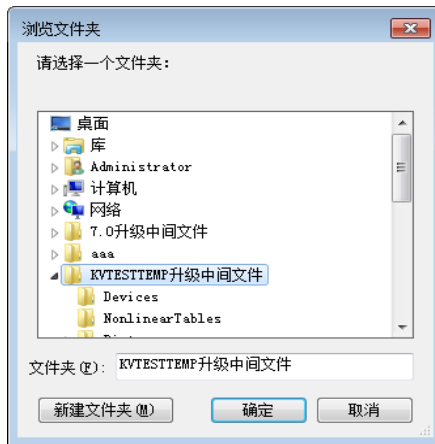


工程升级管理器

步骤三：点击工具栏的“搜索工程”选择要升级的 kv6.55 工程，如下图所示：



步骤四：点击“升级”按钮，并选择升级工程存放路径，如下图所示：



步骤五：工程升级完成后，指定路径下生成了 kv6.55 的工程升级文件，如下图所示：

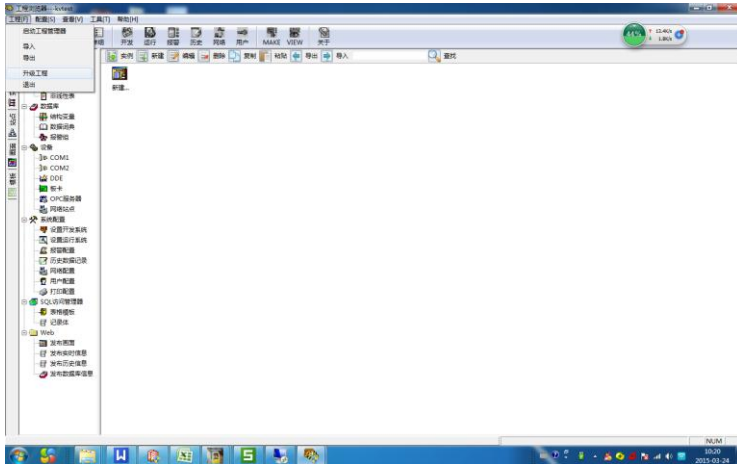


2、通过 7.0 工程浏览器界面加载该中间文件

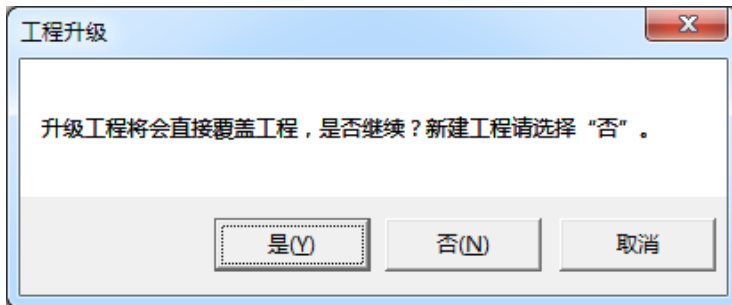
前提准备工作：

1)、kv6.5 的工程通过工程升级工具已经生成了中间文件。

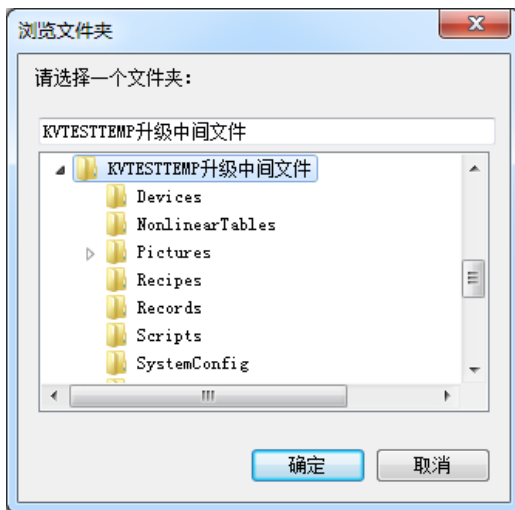
步骤一：在 7.0 中新建工程，打开工程浏览器，点击“升级工程”，如下图所示：



步骤二：提示用户升级会覆盖掉当前工程中的内容，如下图所示：



步骤三：选择“是”，弹出选择中间文件的对话框，选择 655 工程导出的中间文件，点击确定后开始升级。



步骤四：升级完成后，显示“导入信息总结”框，如下图所示：



2、动态分辨率转换

组态王 7.0 画面图形对象显示的大小与做工程时所用计算机的分辨率有关,在不同的

分辨率下对象的显示情况不相同。为了将不同分辨率的工程显示的更加完美，组态王 7.0 提供动态分辨率转换功能。

 例如：

将一个在分辨率为 1024*768 的计算机下做的工程（工程名为 Demo）拷贝到分辨率为 800*600 的计算机上（或者修改计算机的分辨率）。在工程管理器上添加完工程后（“添加工程”的使用方法，请参见“第三章 启动组态王 7.0 - 工程管理器”），在列表中“分辨率”一栏中显示的分辨率为 1024*768，如图 2.26 所示。

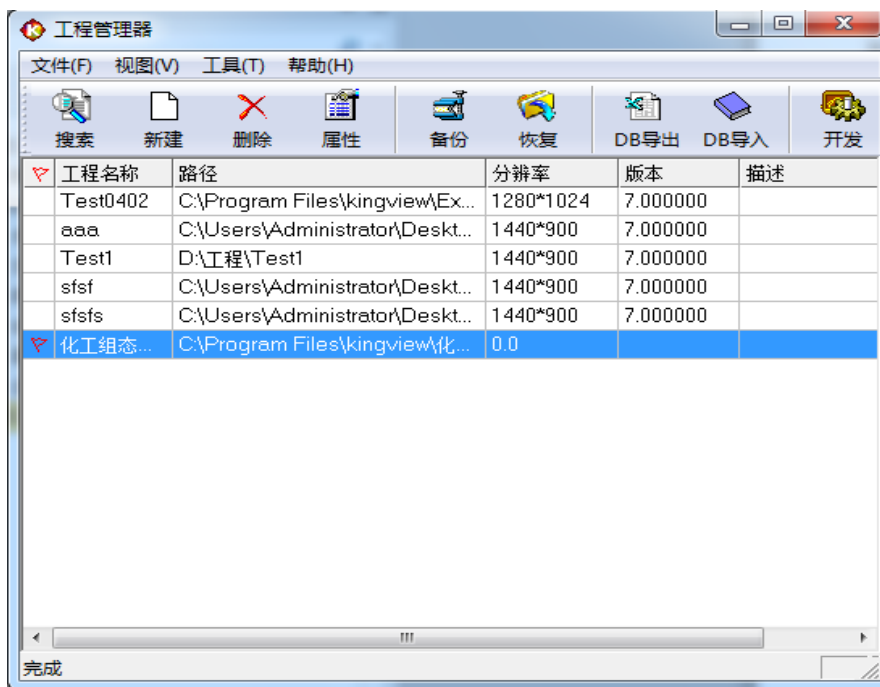


图 2.26 工程管理器分辨率显示

双击蓝色信息条或单击“开发”按钮或选择菜单“工具\切换到开发系统”，进入组态王 7.0 的开发系统。系统将弹出提示询问用户是否进行分辨率的转换，如图 2.27 所示。



图 2.27 是否进行分辨率转换对话框

单击“否”按钮，则不会进行分辨率转换，而是直接进入组态王 7.0 开发系统，画面中的图形对象将会按照 1024*768 时的状态进行显示；单击“是”按钮，则系统自动进行分辨率转换，转换结束后，画面中的图形对象将会按照比例进行缩放，使图形显示合理。

如果您想快速掌握组态王 7.0 使用方法，请阅读《组态王 7.0 培训手册》或参加亚控公司定期举办的组态王 7.0 应用培训班。在这里，有经验丰富的技术工程师为您讲座，使您更快的、全面的掌握组态王 7.0 的使用方法和使用技巧。

第三章 启动组态王 7.0 - 工程管理器

- ☞ 介绍工程管理器的相关功能
- ☞ 教您如何使用工程管理器来管理工程

对于系统集成商和用户来说，一个系统开发人员可能保存有很多个组态王 7.0 工程，对于这些工程的集中管理以及新开发工程中的工程备份等都是比较繁琐的事情，工程管理器实现了对组态王 7.0 各种版本工程的集中管理，更使用户在进行工程开发和工程的备份、数据词典的管理上方便了许多。其主要作用就是为用户集中管理本机上的所有组态王 7.0 工程。

工程管理器的主要功能包括：新建工程、删除工程，搜索指定路径下的所有组态王 7.0 工程，修改工程属性、工程的备份、恢复，数据词典的导入导出，切换到组态王 7.0 开发或运行环境等。本章首先对工程管理器各个菜单命令进行介绍，然后详细介绍每一功能实现的步骤和方法。

3.1 认识工程管理器

工程管理器界面包含菜单栏、工具栏、工程信息显示区和状态栏，如下图所示：

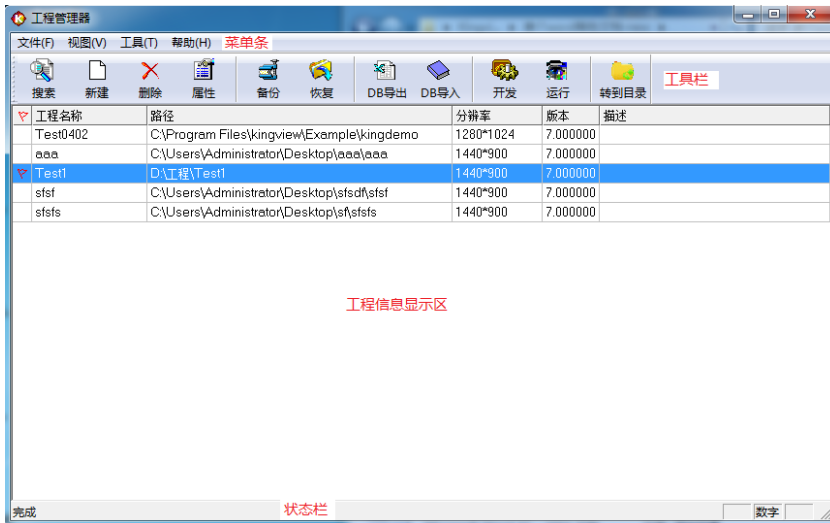


图 3.1 工程管理器界面

3.1.1 菜单条

3.1.1.1 文件菜单

单击“文件(F)”菜单，或按下{ALT}+[F]热键，弹出下拉式菜单，如图 3.2 所示。



图 3.2 “文件”菜单

◇ 文件(F)\新建工程(N)

该菜单命令为新建一个组态王 7.0 工程，但此处新建的工程，在实际上并未真正创建工程，只是在用户给定的工程路径下设置了工程信息，当用户将此工程作为当前工程并且切换到组态王 7.0 开发环境时才真正创建工程。具体如何新建一个工程请参见“如何新建一个工程”一节。

◇ 文件(F)\搜索工程(S)

该菜单命令为搜索用户指定目录下的所有组态王 7.0 工程（包括不同版本、不同分辨率的工程），将其工程名称、工程所在路径、分辨率、开发工程时用的组态王 7.0 软件版本、工程描述文本等信息加入到工程管理器中。搜索出的工程包括指定目录和其子目录下的所有工程。具体如何搜索工程请参见“如何找到一些已有的组态王 7.0 工程”一节。

◇ 文件(F)\添加工程(A)

该菜单命令主要是单独添加一个已经存在的组态王 7.0 工程,并将其添加到工程管理器中来(与搜索工程不同的是:搜索工程是添加搜索到的指定目录下的所有组态王 7.0 工程)。具体如何添加工程请参见“如何找到一个已有的组态王 7.0 工程”一节。

◇ 文件(F)\设为当前工程(C)

该菜单命令将工程管理器中选中加亮的工程设置为组态王 7.0 的当前工程。以后进入组态王 7.0 开发系统或运行系统时,系统将默认打开该工程。被设置为当前工程的工程在工程管理器信息框的表格的第一列中用一个图标(小红旗)来标识。具体如何设为当前工程请参见“如何设置一个工程为当前工程”一节。

◇ 文件(F)\删除工程(D)

该菜单命令将删除在工程管理器信息显示区中当前选中加亮的但没有被设置为当前工程的工程。具体如何删除工程请参见“如何删除工程”一节。

◇ 文件(F)\重命名(R)

该菜单命令将当前选中加亮的工程名称进行修改。如图 3.3 所示为重命名工程对话框。

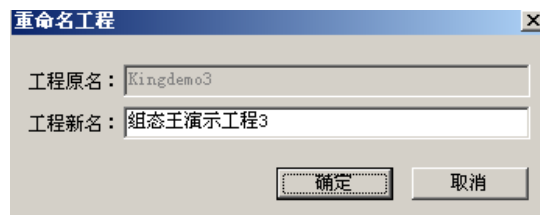


图 3.3 重命名工程对话框

在“工程原名”文本框中显示工程的原名称，该项不可修改。在“工程新名”文本框中输入工程的新名称，单击“确定”确认修改结果，单击“取消”退出工程重命名操作。

◇ 文件(F)\工程属性(P)

该菜单命令将修改当前选中加亮工程的工程属性。具体如何修改工程属性请参见“如何修改当前工程的属性”一节。

◇ 文件(F)\清除工程信息(E)

该菜单命令是将工程管理器中当前选中的高亮显示的工程信息条从工程管理器中清除，不再显示，执行该命令不会删除工程或改变工程。用户可以通过“搜索工程”或“添加工程”重新使该工程信息显示到工程管理器中。具体如何清除工程信息请参见“如何清除当前不需要显示的工程”一节。

◇ 文件(F)\转到工程目录

该菜单命令是将工程管理器中当前选中的工程转到工程所在的目录下。

◇ 文件(F)\退出(X)

退出组态王 7.0 工程管理器。

3.1.1.2 视图菜单

单击“视图(V)”菜单，或按下{ALT}+{V}热键，弹出下拉式菜单，如图 3.4 所示。

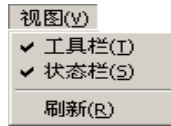


图 3.4 “视图”菜单

◇ 工具栏(T)

选择是否显示工具栏，当“工具栏”被选中时（有对勾标志），显示工具栏；否则不显示。

◇ 状态栏(S)

选择是否显示状态栏，当“状态栏”被选中时（有对勾标志），显示状态栏；否则不显示。

◇ 刷新(R)

刷新工程管理器窗口。

3.1.1.3 工具菜单

单击“工具(T)”菜单，或按下{ALT}+{T}热键，弹出下拉式菜单，如图 3.5 所示。

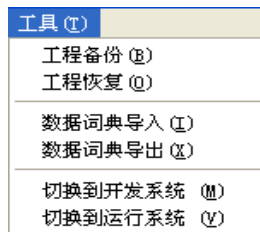


图 3.5 “工具”菜单

◇ 工具(T)\工程备份(B)

该菜单命令是将工程管理器中当前选中加亮的工程按照组态王 7.0 指定的格式进行压缩备份。具体如何备份工程请参见“如何备份和恢复工程”一节。

◇ 工具(T)\工程恢复(O)

该菜单命令是将组态王 7.0 的工程恢复到压缩备份前的状态。具体如何恢复工程请参见“如何备份和恢复工程”一节。

◇ 工具(T)\数据词典导入(I)

为了使用户更方便地使用、查看、定义或打印组态王 7.0 的变量，组态王 7.0 提供数据词典的导入导出功能。数据词典导入命令是将 Excel 中定义好的数据或将由组态王 7.0 工程导出的数据词典导入到组态王 7.0 工程中。该命令常和数据词典导出命令配合使用。

◇ 工具(T)\数据词典导出(X)

该菜单命令是将组态王 7.0 的变量导出到 Excel 格式的文件中，用户可以在 Excel 文件中查看或修改变量的一些属性，或直接在该文件中新建变量并定义其属性，然后导入到工程中。该命令常和数据词典导入命令配合使用。

◇ 工具(T)\切换到开发系统(M)

执行该命令进入组态王 7.0 开发系统，同时将自动关闭工程管理器。打开的工程为工程管理器中指定的当前工程（标有当前工程标志的工程）。

◇ 工具(T)\切换到运行系统(V)

执行该命令进入组态王 7.0 运行系统，同时将自动关闭工程管理器。打开的工程为工

程管理器中指定的当前工程（标有当前工程标志的工程）。

3.1.1.4 帮助菜单

帮助(H)\关于组态王 7.0 工程管理器(A) ...

执行该命令将弹出组态王 7.0 工程管理器的版本号和版权等信息。

3.1.2 工具栏

组态王 7.0 工程管理器工具条如图 3.6 所示



图 3.6 工程管理器工具条

- | | |
|---|--------|
|  | 搜索工程 |
|  | 新建工程 |
|  | 删除工程 |
|  | 修改工程属性 |
|  | 备份工程 |



恢复工程



导出数据词典



导入数据词典



切换到开发系统



切换到运行系统

工具条按钮功能与菜单功能相同，详见前面三节介绍。

3.1.3 工程信息显示区

管理工程的列表，该列表显示和工程有关的信息，包括：工程名称、路径、分辨率、版本及描述等信息，列表中的每个工程都支持双击、右键单击等操作。

3.1.4 状态栏

显示工程的状态信息。当选中某工程，默认显示“完成”。当鼠标右键，显示如快捷菜单中的信息。

3.1.5 快捷菜单

在工程管理器内任何一个工程信息条上单击右键，弹出快捷菜单，如图 3.7 所示。

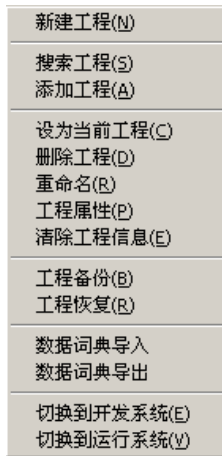


图 3.7 快捷菜单

快捷菜单功能与普通菜单功能一致，详见前面三节介绍。

3.2 如何管理工程

3.2.1 新建一个工程

1、向导 1_新建工程

组态王 7.0 提供新建工程向导。利用向导新建工程，使用户操作更简便、简单。单击菜单栏“文件\新建工程”命令或工具条“新建”按钮或快捷菜单“新建工程”命令后，弹出“新建工程向导一”对话框，如图 3.8 所示。

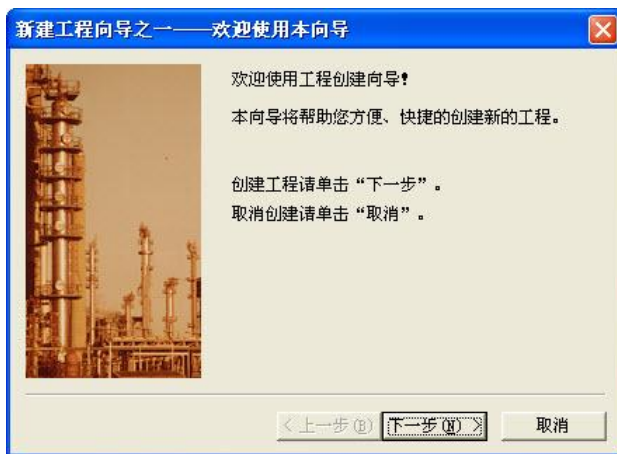


图 3.8 新建工程向导一

- ◇ **取消按钮**：单击“取消”退出新建工程向导。
- ◇ **确定按钮**：单击“下一步”进入向导 2 界面。

2、向导 2_选择工程路径

点击“下一步”弹出“新建工程向导二”对话框，如图 3.9 所示。



图 3.9 新建工程向导二

- ◇ **工程路径文本框**：在对话框的文本框中输入新建工程的路径，如果输入的路径不存

在，系统将自动提示用户。或单击“浏览”按钮，从弹出的路径选择对话框中选择工程路径（可在弹出的路径选择对话框中直接输入路径）。

✧ **上一步按钮**：单击“上一步”返回上一页向导对话框。

✧ **取消按钮**：单击“取消”退出新建工程向导。

3、向导 3_输入工程名称和描述

点击“下一步”进入新建工程向导三，如图 3.10 所示。



图 3.10 新建工程向导三

✧ **工程名称**文本框：输入新建工程的名称，名称有效长度小于 32 个字符。

✧ **工程描述**文本框：输入对新建工程的描述文本，描述文本有效长度小于 40 字符。

✧ **上一步按钮**：单击“上一步”返回向导 2。

✧ **取消按钮**：单击“取消”退出新建工程向导。

✧ **完成按钮**：单击“完成”确认新建的工程，完成新建工程操作。新建工程的路径向

导二中指定的路径，在该路径下会以工程名称为目录建立一个文件夹。完成后弹出“是否将新建的工程设为组态王 7.0 当前工程”对话框，如图 3.11 所示。

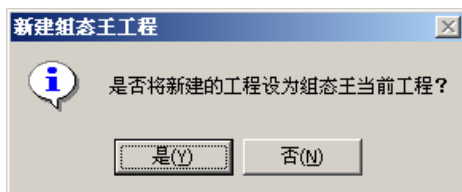


图 3.11 设为当前工程

- 单击“是”将新建的工程设置为组态王 7.0 的当前工程；
- 单击“否”不改变当前工程的设置。

完成以上操作就可以新建一个组态王 7.0 工程的工程信息了。

⚠ 注意：

只有当组态王 7.0 的开发系统或运行系统没有打开时该项有效。此处新建的工程，在实际上并未真正创建工程，只是在用户给定的工程路径下设置了工程信息，当用户将此工程作为当前工程，并且切换到组态王 7.0 开发环境时才真正创建工程。

3.2.2 如何找到一个已有的组态王 7.0 工程

在工程管理器中使用“添加工程”命令来找到一个已有的组态王 7.0 工程，并将工程的信息显示在工程管理器的信息显示区中。单击菜单栏“文件\添加工程”命令或快捷菜单“添加工程”命令后，弹出添加路径选择对话框，如图 3.12 所示。



图 3.12 添加工程路径选择对话框

选择想要添加的工程所在的路径。将选定的工程路径下的组态王 7.0 工程添加到工程管理器中，如果选择的路径不是组态王 7.0 的工程路径，则添加不了。

如果添加的工程名称与当前工程信息显示区中存在的工程名称相同，则被添加的工程将动态生成一个工程名称，在工程名称后添加序号。当存在多个具有相同名称的工程时，将按照顺序生成名称，直到没有重复的名称为止。

- ✧ **取消：**取消添加工程操作。
- ✧ **确定：**单击“确定”将指定路径下的工程添到工程管理器显示区。如图 3.13 所示。

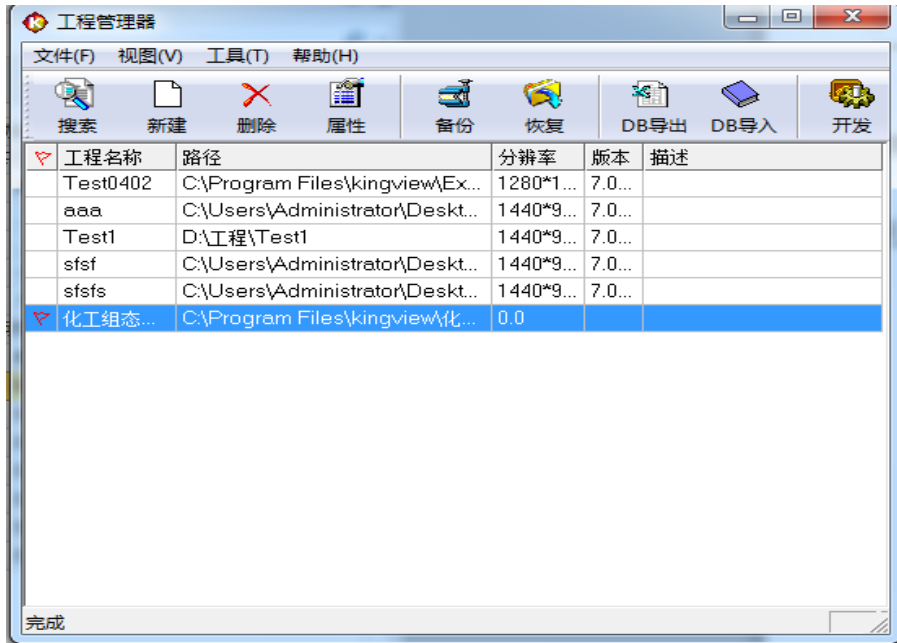


图 3.13 添加工程

3.2.3 如何找到一些已有的组态王 7.0 工程

添加工程只能单独添加一个已有的组态王 7.0 工程,要想找到更多的组态王 7.0 工程,只能使用“搜索工程”命令。单击菜单栏“文件\搜索工程”命令或工具条“搜索”按钮或快捷菜单“搜索工程”命令后,弹出选择搜索路径对话框,如图 3.14 所示。



图 3.14 搜索工程路径选择对话框

路径的选择方法与 Windows 的资源管理器相同，选定有效路径之后，单击“确定”按钮，工程管理器开始搜索工程。将搜索指定路径及其子目录下的所有工程。搜索完成后，搜索结果自动显示在管理器的信息显示区内，路径选择对话框自动关闭。单击“取消”按钮，取消搜索工程操作。

如果搜索到的工程名称与当前工程信息表格中存在的工程名称相同，或搜索到的工程中有相同名称的，在工程信息被添加到工程管理器时，将动态地生成工程名称，在工程名称后添加序号。当存在多个具有相同名称的工程时，将按照顺序生成名称，直到没有重复的名称为止。

3.2.4 如何设置一个工程为当前工程

在工程管理器工程信息显示区中选中加亮想要设置的工程，单击菜单栏“文件\设为当前工程”命令或快捷菜单“设为当前工程”命令即可设置该工程为当前工程。以后进入

组态王 7.0 开发系统或运行系统时，系统将默认打开该工程。被设置为当前工程的工程在工程管理器信息显示区的第一列中用一个图标（小红旗）来标识，如图 3.15 所示。

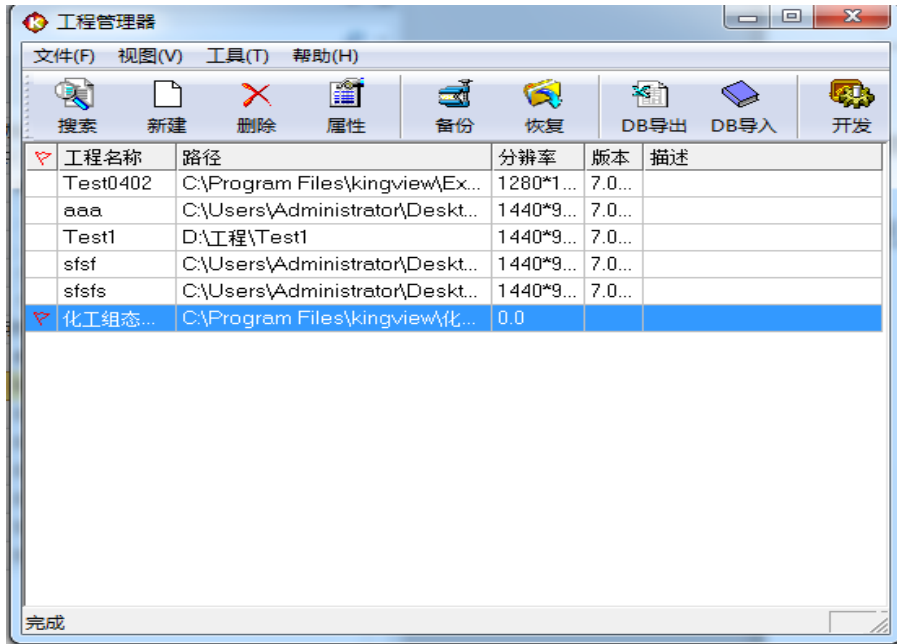


图 3.15 设置当前工程

! 注意:

只有当组态王 7.0 的开发系统或运行系统没有打开时该项有效。

3.2.5 如何修改当前工程的属性

修改工程属性主要包括工程名称和工程描述两个部分。选中要修改属性的工程，使之加亮显示，单击菜单栏“文件\工程属性”命令或工具条“属性”按钮或快捷菜单“工程

属性”命令后，弹出修改“工程属性”的对话框如图 3.16 所示。



图 3.16 修改工程属性对话框

- ◇ **工程名称**文本框：显示的为原工程名称，用户可直接修改。
- ◇ **版本、分辨率**文本框：分别显示开发该工程的组态王 7.0 软件版本和工程的分辨率。
- ◇ **工程路径**文本框：显示该工程所在的路径。
- ◇ **描述**文本框：显示该工程的描述文本，允许用户直接修改。

3.2.6 如何清除当前不需要显示的工程

选中要清除信息的工程，使之加亮显示，单击菜单栏“文件\清除工程信息”命令或快捷菜单“清除工程信息”命令后，将显示的工程信息条从工程管理器中清除，不再显示，执行该命令不会删除工程或改变工程。用户可以通过“搜索工程”或“添加工程”重新使该工程信息显示到工程管理器中。

 注意：

清除工程信息命令只能将非当前工程的信息从工程管理器中删除，对于当前工程该命令无效。

3.2.7 如何备份和恢复工程

备份命令是将选中的组态王 7.0 工程按照指定的格式进行压缩备份。恢复命令是将组态王 7.0 的工程恢复到压缩备份前的状态。下面分别讲解如何备份和恢复组态王 7.0 工程。

3.2.7.1 工程备份

选中要备份的工程，使之加亮显示。单击菜单栏“工具\工程备份”命令或工具条“备份”按钮或快捷菜单“工程备份”命令后，弹出备份工程对话框，如图 3.17 所示。

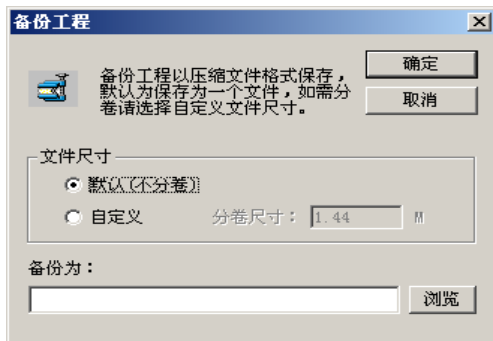


图 3.17 备份工程对话框

工程备份文件分为两种形式：不分卷、分卷。不分卷是指将工程压缩为一个备份文件，无论该文件有多大。分卷是指将工程备份为若干指定大小的压缩文件。系统的默认方式是不分卷。

- ◇ **默认（不分卷）：**选择该选项，系统将把整个工程压缩为一个备份文件。单击“浏览”按钮，选择备份文件存储的路径和文件名称，如图 3.18 所示。工程被存储成扩展名为 .cmp 的文件，如：filename.cmp。工程备份完后，生成一个 filename.cmp 文件。

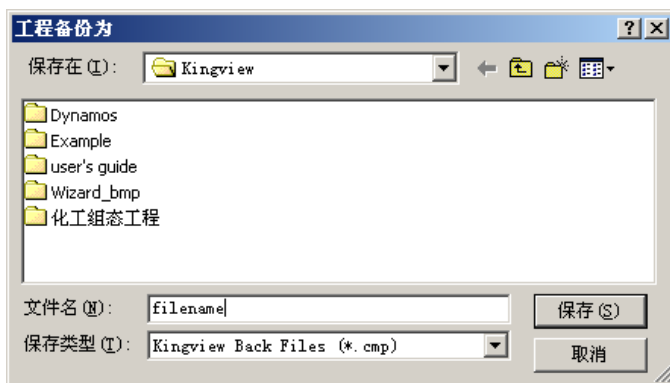



图 3.18 工程备份路径选择

- ◇ **自定义（分卷）：**选择该选项，系统将把整个工程按照给定的分卷尺寸压缩为给定大小的多个文件。“分卷尺寸”文本框变为有效，在该文本中输入分卷的尺寸，即规定每个备份文件的大小，单位为兆。分卷尺寸不能为空，否则系统会提示用户输入分卷尺寸大小。单击“浏览”按钮，选择备份文件存储的路径和文件名称。分卷文

件存储时会自动生成一系列文件，生成的第一个文件的文件名为所定义的文件名.cmp，其它依次为：文件名.c01、文件名.c02…。如：定义的文件名为 filename，则备份产生的文件为：filename.cmp、filename.c01、filename.c02…。如果用户指定的存储路径为软驱，在保存时若磁盘满则系统会自动提示用户更换磁盘。这种情况下，建议用户使用“自定义”方式备份工程。备份过程中在工程管理器的状态栏的左边有文字提示，右边有备份进度条标识当前进度。

 注意：

备份的文件名不能为空。

3.2.7.2 工程恢复

选择中要恢复的工程，使之加亮显示。单击菜单栏“工具\工程恢复”命令或工具条“恢复”按钮或快捷菜单“工程恢复”命令后，弹出“选择要恢复的工程”对话框，如图 3.19 所示。

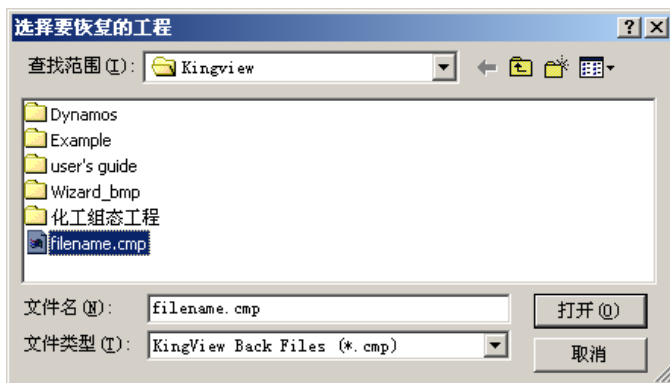


图 3.19 选择要恢复的工程

选择组态王 7.0 备份文件——扩展名为.cmp 的文件，如上例中的 filename.cmp。单击“打开”按钮，弹出“恢复工程”对话框，如图 3.20 所示。

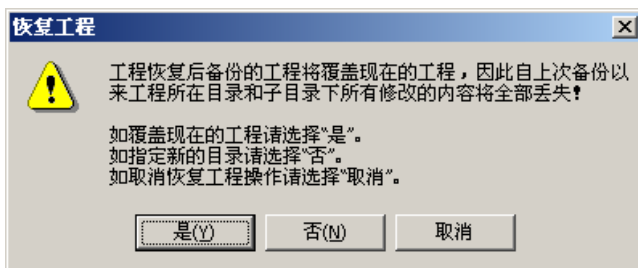


图 3.20 恢复工程选项

- ◇ **是**按钮：单击“是”按钮则以前备份的工程覆盖当前的工程。如果恢复失败，系统会自动将工程还原为恢复前的状态。恢复过程中，工程管理器的状态栏上会有文字提示信息和进度条显示恢复进度。
- ◇ **取消**按钮：单击“取消”按钮取消恢复工程操作。

- ◇ 否按钮：单击“否”按钮，则另行选择工程目录，将工程恢复到别的目录下。单击按钮后弹出路径选择对话框，如图 3.21 所示。

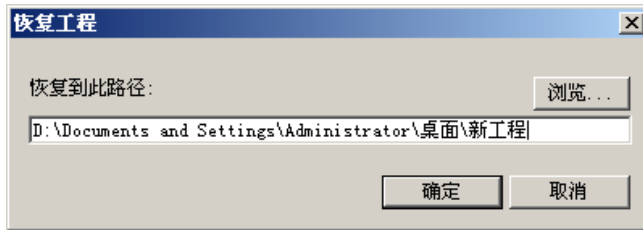


图 3.21 将工程恢复到别的目录下

- 在“恢复到此路径”文本框里输入恢复工程的新的路径。或单击“浏览...”按钮，在弹出的路径选择对话框中进行选择。如果输入的路径不存在，则系统会提示用户是否自动创建该路径。
- 路径输入完成后，单击“确定”按钮恢复工程。工程恢复期间，在工程管理器的状态栏上会有恢复信息和进度显示。
- 工程恢复完成后，弹出恢复成功与否信息框，如图 3.22 所示。

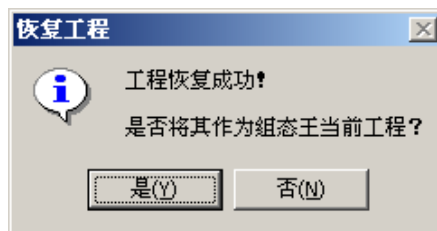


图 3.22 恢复工程成功

单击“是”将恢复的工程作为当前工程；单击“否”返回工程管理器。

恢复的工程的名称若与当前工程信息表格中存在的工程名称相同，则恢复的工程添

加到工程信息表格时将动态地生成一个工程名称，在工程名称后添加序号，如：原工程名为“Demo”，则恢复后的工程名为“Demo (2)”；恢复的工程路径为指定路径下的以备份文件名为子目录名称的路径。

 注意：

- 1、 恢复工程将丢失自备份后的新的工程信息。需要慎重操作。
 - 2、 如果用户选择的备份工程不是原工程的备份时，系统在进行覆盖恢复时，会提示工程错误。
-

3.2.8 如何删除工程

选中要删除的工程，该工程为非当前工程，使之加亮显示，单击菜单栏“文件\删除工程”命令或工具条“删除”按钮或快捷菜单“删除工程”命令后，为防止用户误操作，弹出删除工程确认对话框，提示用户是否确定删除，如图 3.23 所示。单击“是”删除工程，单击“否”取消删除工程操作。删除工程将从工程管理器中删除该工程的信息，工程所在目录将被全部删除，包括子目录。

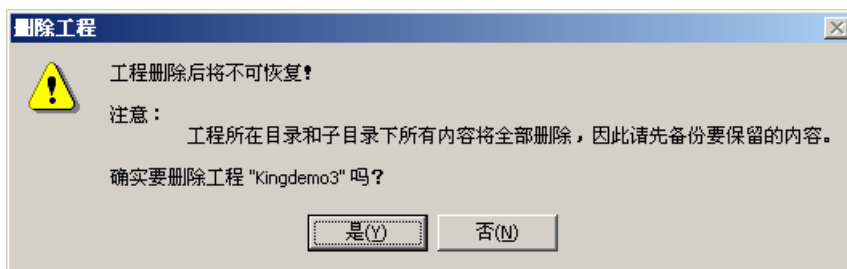


图 3.23 删除工程确认



注意：

删除工程将把工程的所有内容全部删除，不可恢复。用户应注意操作。

第四章 组态王 7.0 开发环境 - 工程浏览器

- ☞ 介绍组态王 7.0 工程浏览器
- ☞ 介绍组态王 7.0 画面开发系统

工程浏览器是组态王 7.0 的一个重要组成部分，它将图形画面、命令语言、设备驱动程序、配方、报警、网络等工程元素集中管理，工程人员可以一目了然地查看工程的各个组成部分。工程浏览器简便易学，操作界面和 Windows 中的资源管理器非常类似，为工程的管理提供了方便高效的手段。

组态王 7.0 开发系统内嵌于组态王 7.0 工程浏览器，又称为画面开发系统，是应用程序的集成开发环境，工程人员在这个环境里进行系统开发。

4.1 工程浏览器概述

组态王 7.0 的工程浏览器由 Tab 标签条、菜单栏、工具栏、工程目录显示区、目录内容显示区、状态栏组成。其中，工程目录显示区以树形结构图显示功能节点，用户可以扩展或收缩工程浏览器中所列的功能项，主要展示工程的各个组成部分。主要包括“系统”，“变量”，“站点”、“画面”和“模板管理”五部分，这五部分的切换是通过工程浏览器最

左侧的 Tab 标签实现的。“目录内容显示区”，将显示每个工程组成部分的详细内容，同时对工程提供必要的编辑修改功能。

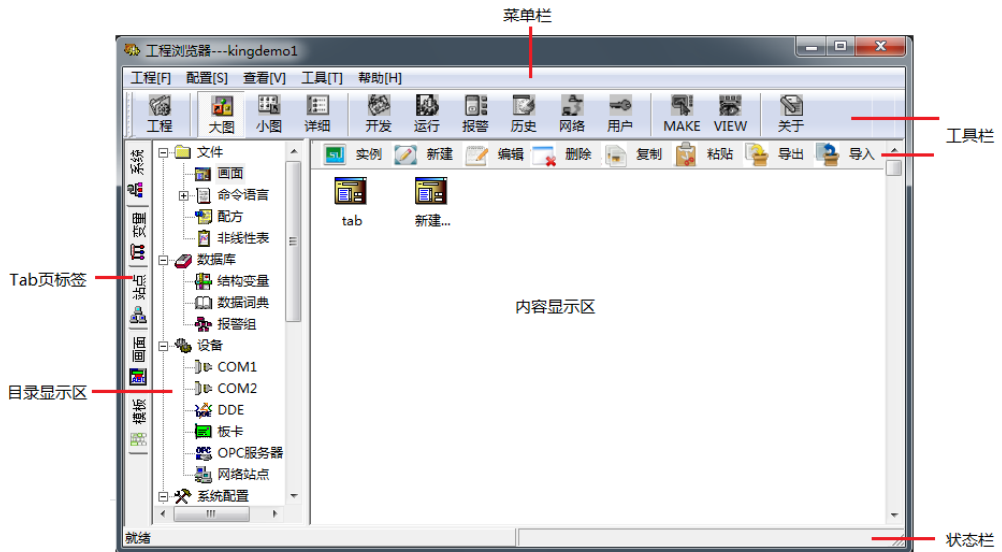


图 4.1 组态王 7.0 工程浏览器

4.1.1 菜单栏

4.1.1.1 工程菜单

用鼠标单击菜单栏上的“工程”菜单，弹出下拉式菜单，如图 4.2 所示。



图 4.2 工程菜单

1、工程\启动工程管理器

此菜单命令用于打开工程管理器，单击“工程\启动工程管理器”菜单，则弹出“工程管理器”画面，如图 4.3 所示。

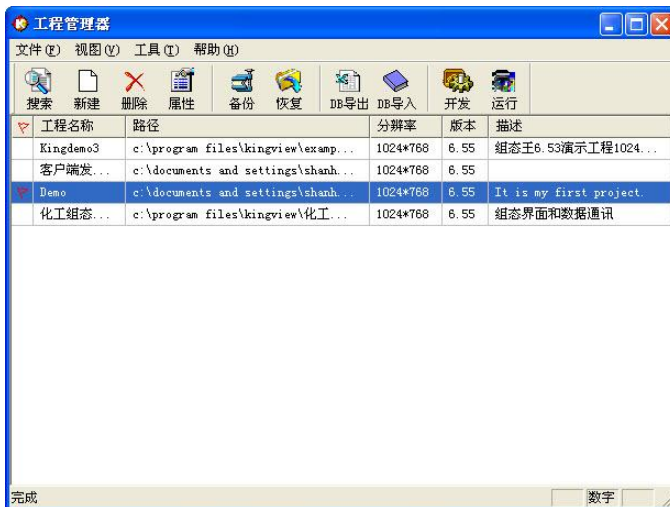


图 4.3 组态王 7.0 工程管理器

利用组态王 7.0 工程管理器可以使用户集中管理本机上的所有组态王 7.0 工程。工程

管理器的主要功能包括：新建、删除工程，对工程重命名，搜索组态王 7.0 工程，修改工程属性，工程的备份、恢复，数据词典的导入、导出，切换到组态王 7.0 开发或运行环境等。详细使用方法请参见“第三章 启动组态王 7.0- 工程管理器”。

2、工程\导入

此菜单命令用于将另一组态王 7.0 工程的画面和命令语言导入到当前工程中。

◇ 向导 1_导入：

单击“工程\导入”菜单，则弹出“画面和命令语言导入向导”画面，如图 4.4 所示。



图 4.4 画面和命令语言导入向导

◇ 向导 2_路径选择：

单击“取消”按钮退出画面和命令语言导入向导，单击“下一步”按钮：进入“选择路径”向导界面，如图 4.5 所示。



图 4.5 第一步：选择路径

在画面文本框中输入保存有组态王 7.0 画面和命令语言文件的路径。若希望对路径进行选择，单击“浏览”按钮，弹出“打开”对话框，如图 4.6 所示。

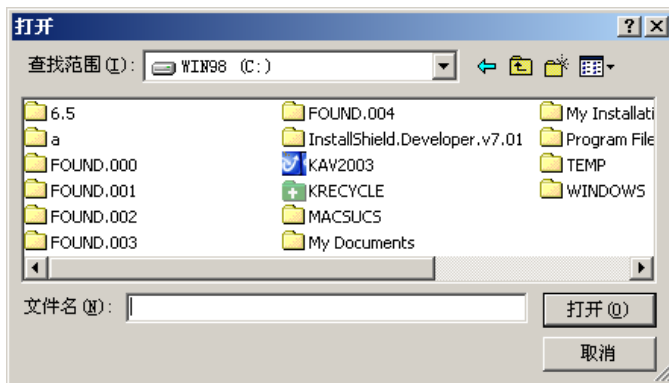


图 4.6 “打开”对话框

在对话框中选择正确的路径，如：C:\ProgramFiles\Kingview\Example\Demo，单击“打开”按钮，则返回到“选择路径”向导界面，选择的路径显示在路径文本框内。如

图 4.7 所示。



图 4.7 第一步：选择路径

◇ 向导 3_画面和命令语言选择：

单击“上一步”按钮返回“向导 2”界面；单击“下一步”按钮：进入“选择画面和命令语言”向导界面。如图 4.8 所示。



图 4.8 第二步：选择画面和命令语言

- 单击“画面”和“命令语言”后面“详细资料…”按钮可以对二者进行详细选择。单击“画面”—“详细资料…”按钮，弹出“选定画面”对话框。如图 4.9 所示。

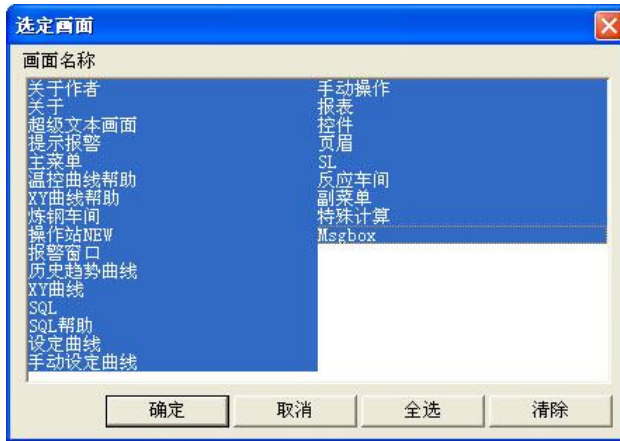


图 4.9 “选定画面”对话框

- 系统默认是全部选中画面进行导入。在对话框中选择想要导入的画面，可用鼠标对画面进行逐一选择，也可点击“全选”按钮全部选中。
- 单击“清除”按钮将清除全部选定画面。
- 单击“确定”返回到“第二步：选择画面和命令语言”对话框。
- 单击“命令语言”—“详细资料…”按钮，弹出“命令语言选项”对话框。如图 4.10 所示。

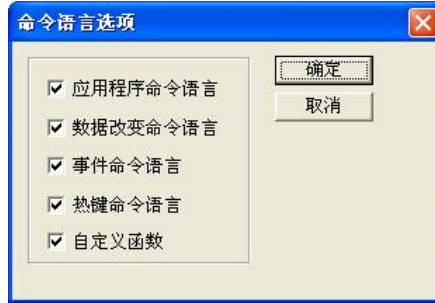


图 4.10 命令语言选项

在此对话框中对想要导入的命令语言进行选择，然后单击“确定”，返回到“选择画面和命令语言”向导界面。

◇ 向导 4_选择函数导入规则：

点击“选择画面和命令语言”向导界面中的“完成”按钮。系统首先完成对画面的导入。画面导入完成后弹出“函数导入”对话框，如图 4.11 所示。

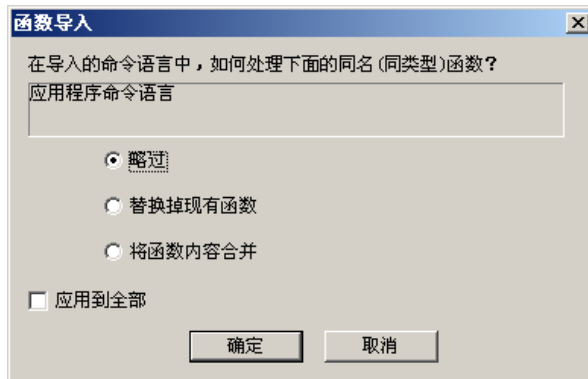


图 4.11 函数导入对话框

在此对话框中选择对应用程序命令语言导入时处理同名（同类型）函数的规则。

- **略过：**遇到同名（同类型）的函数时，不予导入。
- **替换掉现有函数：**遇到同名（同类型）的函数时，将被导入文件中的同名（同类型）函数替换现有的函数。
- **将函数内容合并：**遇到同名（同类型）的函数时，将被导入文件中的同名（同类型）函数的内容合并到现有的函数中。
- **应用到全部：**选中此复选框，则对数据改变命令语言、事件命令语言、热键命令语言和自定义函数命令语言都应用同样的规则。否则将依次出现各种命令语言的“函数导入”对话框。
- **确定：**单击“确定”，系统进行命令语言的导入。导入命令语言结束后，就将其他组态王 7.0 工程中的画面和命令语言导入到当前的组态王 7.0 工程中。

3、工程\导出

此菜单命令用于将当前组态王 7.0 工程的画面和命令语言导出到指定文件夹中。

◇ 向导 1_导出：

单击“工程\导出”菜单，则弹出“画面和命令语言导出向导”画面，如图 4.12 所示。



图 4.12 画面和命令语言导出向导

◇ 向导 2_路径选择:

单击“取消”按钮：退出画面和命令语言导出向导；单击“下一步”按钮：进入“选择路径”向导界面。如图 4.13 所示。



图 4.13 “第一步：选择路径”对话框

在画面文本框中输入要导出组态王 7.0 画面和命令语言所要保存的路径。若希望对路径进行选择，单击“浏览”按钮，弹出“打开”对话框，操作方法与导入画面和命令语言时选择路径相同。

◇ 向导 3_画面和命令语言选择：


单击“下一步”按钮，进入“选择画面和命令语言”向导界面。如图 4.14 所示。



图 4.14 第二步：选择画面和命令语言

- 单击“画面”和“命令语言”后面“详细资料...”按钮可以对二者进行详细选择。使用方法与导入画面和命令语言详细选择画面和命令语言相同。
- 单击“完成”，系统完成对选定画面和命令语言的导出。可以在相应的导出路径下看到导出的文件。在文件夹中文件有三种类型的文件：*.pic（画面中所有图素的信息文件）、*.cfg（各种命令语言的信息文件）和*.dat（画面属性的信息文件）。具体工程文件都有什么作用，请详见“附录 A 组态王 7.0 使用的数据文

件”。

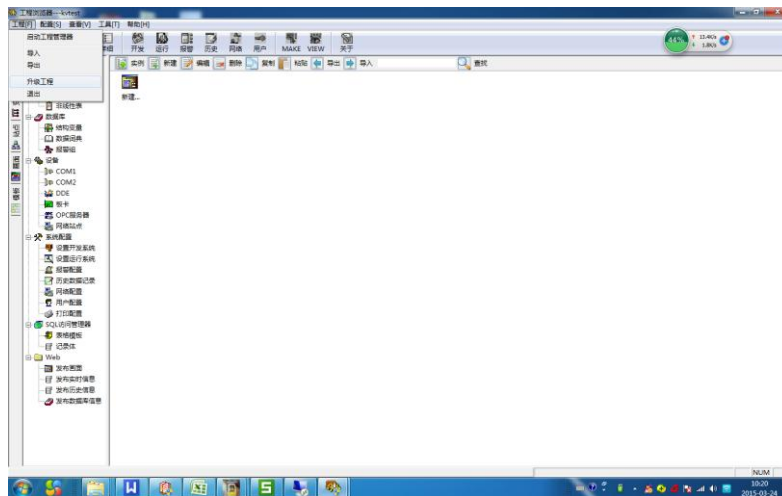
 注意：

使用工程导入\工程导出菜单命令可以重新使用旧工程中的画面和命令语言，减少工程制作人员的工作量，使组态王 7.0 工程具有可重用性。

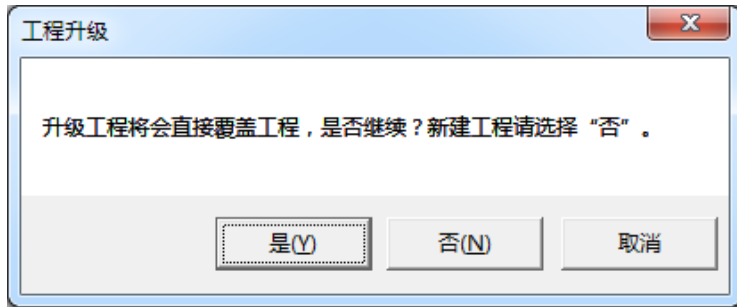
4、工程\工程升级

此菜单命令用于加载 kv6.55 工程升级后的中间文件。

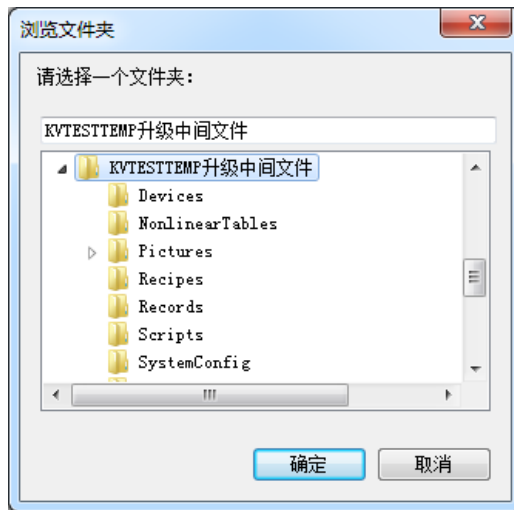
步骤一：在 7.0 中新建工程，打开工程浏览器，点击“升级工程”，如下图所示：



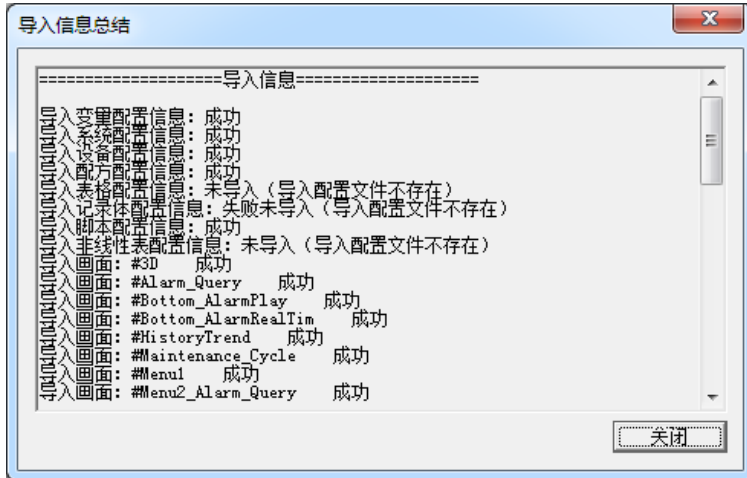
步骤二：提示用户升级会覆盖掉当前工程中的内容，如下图所示：



步骤三：选择“是”，弹出选择中间文件的对话框，选择 655 工程导出的中间文件，
点击确定后开始升级。



步骤四：升级完成后，显示“导入信息总结”框，如下图所示：



5、工程\退出

此菜单命令用于关闭工程浏览器，单击“工程\退出”菜单，则工程浏览器退出。若界面开发系统中有的画面内容被改变而没有保存，程序会提示工程人员选择是否保存。如图 4.15 所示。



图 4.15 退出工程浏览器提示

- ◇ 是按钮：如果要保存已修改的画面内容，单击“是”或单击字母键“Y”；
- ◇ 否按钮：若不保存，单击“否”或单击字母键“N”可退出组态王 7.0 工程浏览器。

◇ **取消按钮**：单击“取消”按钮取消退出操作，则不会退出工程浏览器。

4.1.1.2 配置菜单

用鼠标单击菜单栏上的“配置”菜单，弹出下拉式菜单，如图 4.16 所示。

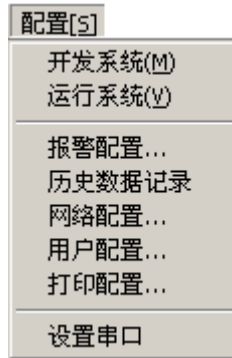


图 4.16 配置菜单

1、配置\开发系统

此菜单命令是用于对开发系统外观进行设置。单击“配置\开发系统”菜单，弹出“开发系统外观定制”画面。如图 4.17 所示。



图 4.17 开发系统外观定制

- ◇ **标题条文本:** 此编辑框用于输入组态王 7.0 画面开发系统标题栏中的标题。
- ◇ **显示工程路径:** 选择此选项使当前工程路径显示在组态王 7.0 开发系统的标题栏中。

2、配置\运行系统

此菜单命令是用于对运行系统外观、定义运行系统基准频率、设定运行系统启动时自动打开的主画面等。单击“配置\运行系统”菜单，弹出“运行系统设置”画面。如图 4.18 所示。



图 4.18 运行系统设置_运行系统外观

“运行系统设置”对话框由三个配置属性页组成：

“运行系统外观”属性页

此属性页中各项的含义和使用介绍如下：

- ◇ **启动时最大化：** TouchView 启动时占据整个屏幕。
- ◇ **启动时缩成图标：** TouchView 启动时自动缩成图标。
- ◇ **标题条文本：** 此文本框用于输入 TouchView 运行时出现在标题栏中的标题。若此内容为空，则 TouchView 运行时将隐去标题条，全屏显示。
- ◇ **系统菜单：** 选择此选项使 TouchView 运行时标题栏中带有系统菜单框。
- ◇ **最小化按钮：** 选择此选项使 TouchView 运行时标题栏中带有最小化按钮。
- ◇ **最大化按钮：** 选择此选项使 TouchView 运行时标题栏中带有最大化按钮。
- ◇ **可变大小边框：** 选择此选项使 TouchView 运行时，可以改变窗口大小。
- ◇ **标题条中显示工程路径：** 选择此选项使当前应用程序目录显示在标题栏。
- ◇ **菜单：** 选择 TouchView 运行时要显示的菜单。

“主画面配置”属性页

单击“主画面配置”标签，显示该属性页，同时属性页画面列表对话框中列出了当前工程中所有有效的画面，选中的画面加亮显示。如图 4.19 所示。此属性页指定了 TouchView 运行系统启动时自动加载的画面。如果几个画面互相重叠，最后调入的画面在前面显示。



图 4.19 运行系统设置_主画面配置

“特殊”属性页

此属性页对话框用于设置运行系统的基准频率等一些特殊属性，单击“特殊”属性页，则此属性页弹出。如图 4.20 所示。

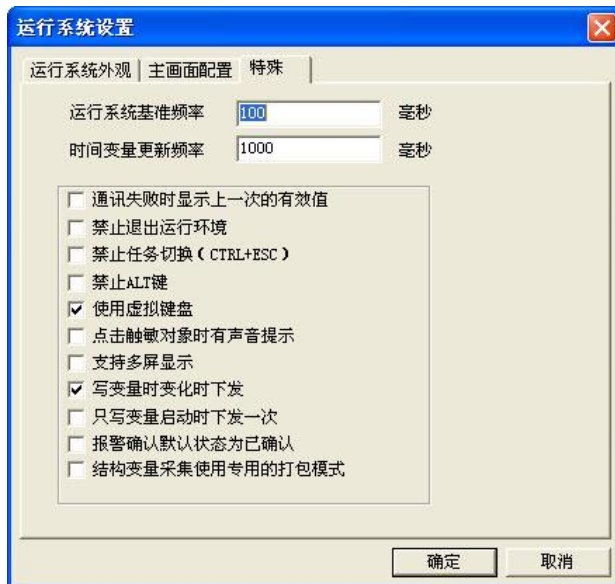


图 4.20 运行系统设置_特殊

- ◇ **运行系统基准频率：**是一个时间值。所有其它与时间有关的操作选项（如：有“闪烁”动画连接的图形对象的闪烁频率、趋势曲线的更新频率、后台命令语言的执行）都以它为单位，是它的整数倍。组态王 7.0 最高基准频率为 55 毫秒。
- ◇ **时间变量更新频率：**用于控制 TouchView 在运行时更新系统时间变量（如\$秒、\$分、\$时等）的频率。
- ◇ **通讯失败时显示上一次的有效值：**用于控制组态王 7.0 中的 IO 变量在通讯失败后在画面上的显示方式。选中此项后，对于组态王 7.0 画面上 IO 变量的“值输出”连接，在设备通讯失败时画面上将显示组态王 7.0 最后采集的数据值，否则将显示“???”。

- ◇ **禁止退出运行环境：**选择此选项后，其左边复选框内出现“✓”号。选择此选项使 TouchView 启动后，用户不能使用系统的“关闭”按钮或菜单来关闭程序，使程序退出运行。但用户可以在组态王 7.0 中使用 EXIT() 函数控制程序退出。
 - ◇ **禁止任务切换 (CTRL+ESC)：**选择此选项后，其左边小方框内出现“✓”号。选择此选项将禁止使用“<CTRL>+<ESC>”键，用户不能作任务切换。
 - ◇ **禁止 ALT 键：**选择此选项后，其左边小方框内出现“✓”号。选择此选项将禁止“<ALT>”键，用户不能用<ALT>键调用菜单命令。
-



注意：

若将上述所有选项选中，只有使用组态王 7.0 提供的内部函数 Exit(Option) 退出。

- ◇ **使用虚拟键盘：**选择此选项后，其左边小方框内出现“✓”号。画面程序运行时，当需要操作者使用键盘时，比如输入模拟值，则弹出模拟键盘窗口，操作者用鼠标在模拟键盘上选择字符即可输入。
 - ◇ **点击触敏对象时有声音提示：**选择此选项后，其左边小方框内出现“✓”号。则系统运行时，鼠标单击按钮等可操作的图素时，蜂鸣器发出声音。
 - ◇ **支持多屏显示：**选择此选项后，其左边小方框内出现“✓”号。选择此选项支持多显卡显示，可以一台主机接多个显示器，组态王 7.0 画面在多个显示器上显示。
 - ◇ **写变量时变化触发：**选择此选项后，如果变量的采集频率为 0，组态王 7.0 写变量的时候，只有变量值发生变化才写，否则不写。
 - ◇ **只写变量启动时下发一次：**对于只写变量，选择此选项后，运行组态王 7.0，将初
-

始值向下写一次，否则不写。

3、配置\报警配置

此菜单命令用于将报警和事件信息输出到文件、数据库和打印机中的配置。各属性页的使用方法在“第九章 报警和事件系统”中“9.5 如何记录、显示报警”里面详细介绍，这里就不再介绍。

4、配置\历史数据记录

此菜单命令和历史数据的记录有关，是用于对历史数据记录文件保存路径和其它参数（如数据保存天数）进行配置。从而可以利用历史趋势曲线显示历史数据。也可进行分布式历史数据配置，使本机节点中的组态王 7.0 能够访问远程计算机的历史数据。详细历史趋势曲线的使用请参见“历史趋势曲线”，“报表历史数据查询函数”，“组态王 7.0 历史数据发布”和“组态王 7.0 历史库”一章。

5、配置\网络配置

此菜单命令用于配置组态王 7.0 网络。各属性页的使用方法在“第二十一章 网络功能”中“网络配置”里面详细介绍，这里就不再介绍。

6、配置\用户配置

此菜单命令用于建立组态王 7.0 用户、用户组，以及安全区配置。使用方法在“第十六章 系统安全管理”中详细介绍，这里就不再介绍。

7、配置\打印配置

此菜单命令用于配置“画面”、“实时报警”、“报告”打印时的打印机。单击“配置\

打印配置”菜单，弹出“打印配置”画面。如图 4.21 所示。

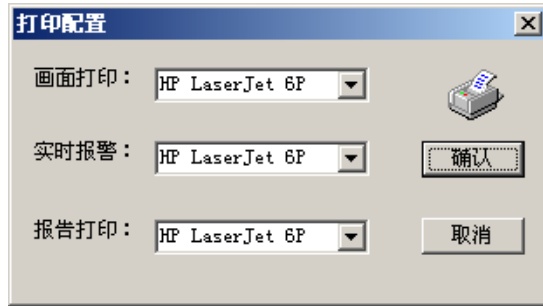



图 4.21 打印配置

其中“画面打印”指定函数 PrintWindow()使用的打印口；“实时报警”指定实时报警打印使用的打印口；“报告打印”指定报表打印函数，如：ReportPrint()使用的打印口。各个列表框中列出了本机上用户定义的打印机名称，可任选其一。

 注意：

这里的打印配置设置的是本地并口的打印机，也是为兼容组态王 7.0 早期版本而保留的。组态王 6.5 以上版本的画面打印、报表打印和通用控件的打印均可使用网络打印机，或系统默认的其它类型的打印机，不必在此处进行设置。

8、配置\设置串口

此菜单命令用于配置串口通讯参数及对 Modem 拨号的设置。单击工程浏览器“工程目录显示区”中“设备”上“COM1”或“COM2”，然后单击“配置\设置串口”菜单；或是直接双击“COM1”或“COM2”。弹出“设置串口”画面。如图 4.22 所示。

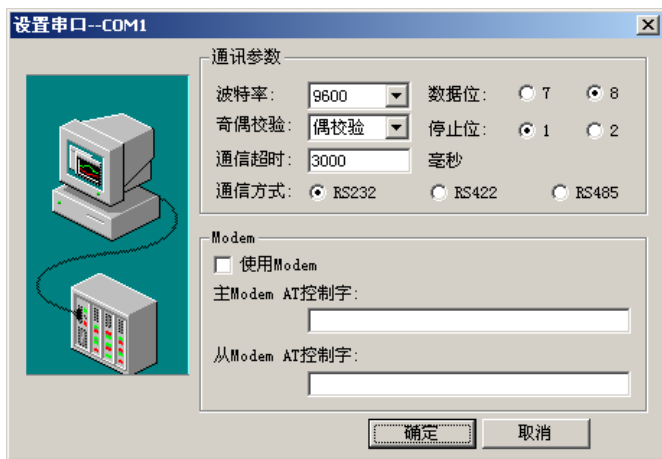


图 4.22 设置串口

具体画面中各项参数如何设置请参见“第六章 I/O 设备管理”。

4.1.1.3 查看菜单

用鼠标单击菜单栏上的“查看”菜单，弹出下拉式菜单，如图 4.23 所示。

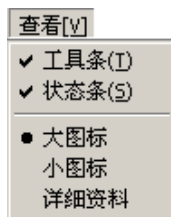


图 4.23 查看菜单

1、查看\工具条

此菜单命令用于显示\关闭工程浏览器的工具条，当工具条菜单左边出现“✓”号时，显示工具条，当工具条菜单左边没有出现“✓”号时，工具条消失。

2、查看\状态条

此菜单命令用于显示\关闭工程浏览器的状态条，当状态条菜单左边出现“√”号时，显示状态条，当状态条菜单左边没有出现“√”号时，状态条消失。

3、查看\大图标

此菜单命令用于将目录内容显示区中的内容以大图标显示。当“大图标”菜单左边出现“•”号时，显示大图标。

4、查看\小图标

此菜单命令用于将目录内容显示区中的内容以小图标显示。当“小图标”菜单左边出现“•”号时，显示小图标。

5、查看\详细资料

此菜单命令用于将目录内容显示区中各成员项所包含的全部详细内容显示出来。

4.1.1.4 工具菜单

用鼠标单击菜单栏上的“工具”菜单，弹出下拉式菜单，如图 4.24 所示。



图 4.24 工具菜单

1、工具\查找数据库变量

此菜单命令用于查找指定数据库中变量，并且显示该变量的详细情况供用户修改。单击工程浏览器“工程目录显示区”中“变量词典”项时，该菜单命令由灰色（不可用）变为黑色（可用）。弹出“查找变量”对话框。如图 4.25 所示。

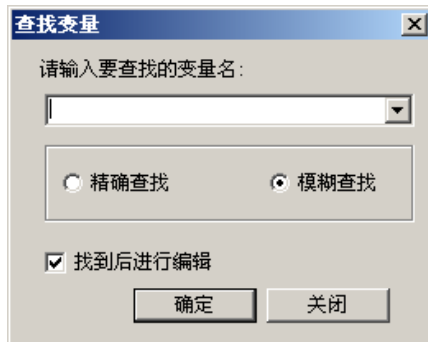


图 4.25 查找变量

使用方法请参见“第五章 变量定义和管理”5.7.3 节。

2、工具\变量使用报告

此菜单命令用于统计组态王 7.0 变量的使用情况，即变量所在的画面以及使用变量的图素在画面中的坐标位置和使用变量的命令语言的类型。单击“工具\变量使用报告”菜单，系统对变量进行统计交替弹出“调入…画面”、“统计…画面”等画面，直到统计完成，弹出“变量使用报告”画面。如图 4.26 所示。



图 4.26 变量使用报告

点击“查找”按钮可以快速查找到某个变量的位置及使用变量的图素在画面中的坐标位置和使用变量的命令语言的类型。



使用方法请参见“第五章 变量定义和管理”。

3、工具\更新变量计数

数据库采用对变量引用进行计数的办法来表明变量是否被引用，“变量引用计数”为0表明数据定义后没有被使用过。当删除、修改某些连接表达式，或删除画面，使变量引用计数变化时，数据库并不自动更新此计数值。用户需要使用更新变量计数命令来统计、更新变量使用情况。

一般情况下工程人员不需要选择此命令，在应用设计结束时作最后的清理工作时才会用到此项功能。

具体使用方法请参见“第五章 变量定义和管理”。

4、工具\删除未用变量

数据库维护的大部分工作都是由系统自动完成的，设计者需要做的是在完成的最后阶

段“删除未用变量”。在删除未用变量之前需要更新变量计数，目的是确定变量是否有动画连接或是否在命令语言中使用过，只有没使用过（变量计数=0）的变量才可以删除。更新变量计数之前要求关闭所有画面。

使用方法请参见“第五章 变量定义和管理”。

5、工具\替换变量

此菜单命令用于将已有的旧变量用新的变量名来替换，单击“工具\替换变量名称”，弹出“单个替换、批量替换”子菜单。如图 4.27 所示。

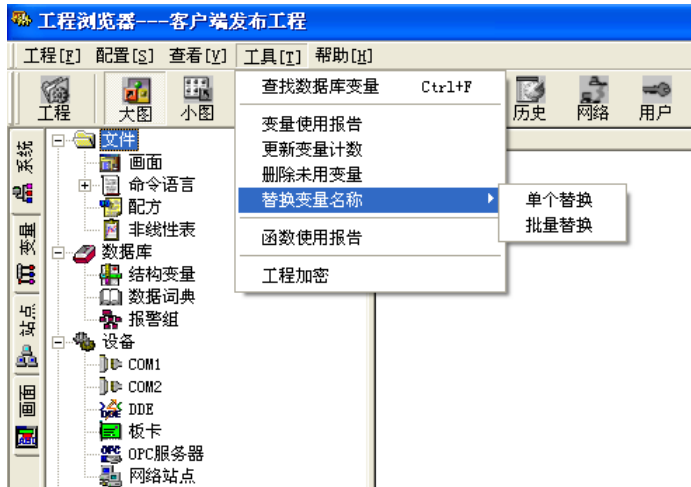


图 4.27 变量名称替换

使用方法请参见“第五章 变量定义和管理” 5.7.4 节。

6、工具\函数使用报告

函数使用报告功能为用户准确提供了工程中函数的使用情况,该功能显示的函数包括组态王 7.0 函数、控件的属性和方法,以及用户自定义函数。

在工程浏览器中,选中“工具/函数使用报告”选项,弹出如图 4.28 所示的对话框。



图 4.28 函数使用报告

上面的对话框列出了所有画面使用到的函数,并且允许将函数使用报告保存为 CSV 文件或根据函数名查找函数。单击“保存”按钮弹出图 4.29 所示的界面。

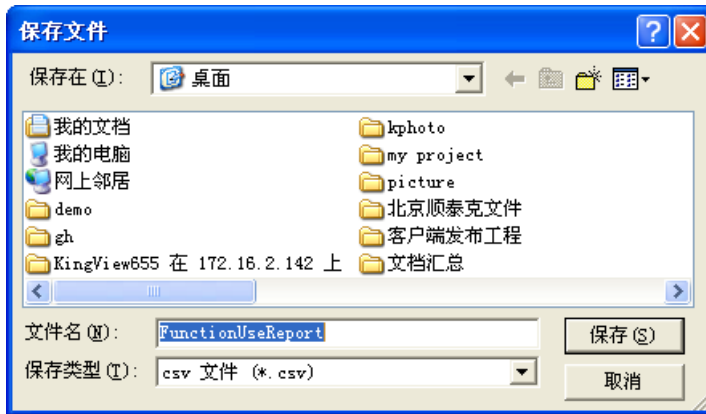


图 4.29 保存函数使用报告

选择保存路径和编辑文件名，也可以使用默认文件名 FunctionUseReport，单击“保存”按钮。打开上面保存的 CSV 文件，如图 4.30 所示。

	A	B	C	D	E	F	G	H
1	组态王函数							
2		Ack	副菜单	(27	358)	--	(378)	
3				(104	358)	--	(378)	
4			报警窗口	(80	561)	--	(588)	
5				(81	560)	--	(587)	
6				(80	526)	--	(553)	
7				(81	526)	--	(553)	
8				(80	490)	--	(517)	
9				(80	491)	--	(518)	
10				(80	452)	--	(479)	
11				(80	453)	--	(480)	
12		chartAdd	副菜单	显示时				
13		chartClear	副菜单	显示时				
14		chartSetV	副菜单	存在时				
15		ClosePictMsgbox		(117	71)	--	(197)	
16				(43	71)	--	(197)	
17			手动操作	(317	6)	--	(3636)	
18			手动设定曲线	(413	9)	--	(4737)	
19			设定曲线	(68	116)	--	(142)	
20				(20	84)	--	(9101)	
21				(19	45)	--	(962)	
22			SQL帮助	(432	567)	--	(594)	
23				(814	538)	--	(593)	
24				(813	538)	--	(593)	
25			XY曲线	(334	0)	--	(3632)	
26				(293	0)	--	(3632)	
27				(243	4)	--	(2629)	
28			vv曲线	(258	522)	--	(564)	

图 4.30 函数使用报告的 CSV 文件

根据函数名查找指定的函数。单击图 4.28 中的“查找”按钮，弹出“查找”对话框，如图 4.31 所示。

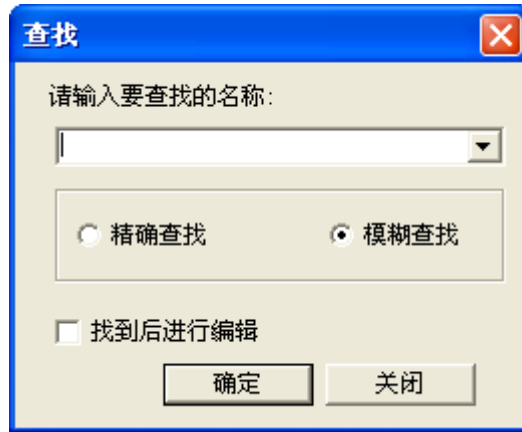


图 4.31 函数查找

输入函数名称，并选择查找类型（精确查找或模糊查找）和选择找到后是否编辑（该功能暂未实现），单击“确定”按钮开始查找函数，单击“关闭”按钮取消该操作。

若查找成功，将会在图 4.28 的对话框中展开查找到的函数，若查找不成功，将会弹出提示窗口。

7、工具\工程加密

为了防止其他人员对工程进行修改，可以对所开发的工程进行加密。也可以将加密的工程进行取消工程密码保护的操作。

使用方法请参见“组态王 7.0 开发系统安全管理”一节。

4.1.1.5 帮助菜单

用鼠标单击菜单栏上的“帮助”菜单，弹出下拉式菜单，如图 4.32 所示。



图 4.32 帮助菜单

此菜单用于弹出信息框显示组态王 7.0 的版本情况和组态王 7.0 的帮助信息。

4.1.2 工具栏

工具栏是工程浏览器中菜单命令的快捷方式。当鼠标放在工具条的任一按钮上时，立刻出现一个提示信息框标明此按钮的功能。工程浏览器的工具按钮条如图 4.33 所示。



图 4.33 工具条

工具按钮条上的每一个按钮对应着一个菜单命令，分别介绍如下：



工程

此按钮用于打开“组态王 7.0 工程管理”对话框，是“工程\启动工程管理器”菜单命令的快捷方式。



大图

此按钮用于设置目录内容显示方式为“大图标”方式，是“查看\大图标”菜单命令的快捷方式。



小图

此按钮用于设置目录内容显示方式为“小图标”方式，是“查看\小图标”菜单命令的快捷方式。



详细

此按钮用于设置目录内容显示方式为“详细资料”方式，是“查看\详细资料”菜单命令的快捷方式。



开发

此按钮用于配置组态王 7.0 开发系统 TouchExplorer 的外观，是“配置\开发系统”菜单命令的快捷方式。



运行

此按钮用于配置组态王 7.0 运行系统 TouchView 的外观，是“配置\运行系统”菜单命令的快捷方式。



报警

此按钮用于报警配置，单击此按钮后弹出“报警配置属性页”对话框，是“配置\报警配置”菜单命令的快捷方式。



历史

此按钮用于历史数据记录配置，单击此按钮后弹出“历史库配置”对话框，是“配置\历史数据记录”菜单命令的快捷方式。



网络

此按钮用于网络设置，单击此按钮后弹出“网络配置”对话框，是“配置\网络配置”菜单命令的快捷方式。



用户

此按钮用于用户和安全区的设置，单击此按钮后弹出“用户和安全区管理器”对话框，是“配置\用户配置”菜单命令的快捷方式。



MAKE

此按钮用于“切换到 Make”，即切换到组态王 7.0 画面开发系统。



VIEW

此按钮用于“切换到 View”，即切换到组态王 7.0 运行环境。



此按钮用于提供组态王 7.0 的系统帮助信息，是“帮助\关于”菜单命令的快捷方式。

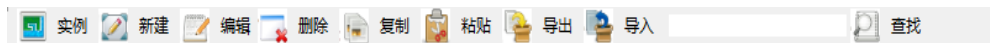


图 4.33 工具条

- 实例：从管理器中选择模板生成到组态王当前工程中；
- 新建：通过模板生成向导，创建某个类型的模板；
- 编辑：编辑选择模板的基本属性信息；
- 删除：将选择的模板文件从管理器中删除，删除后不可恢复；
- 复制、粘贴：复制选择的模板，并将其粘贴到其它的模板分组中；
- 导出：将选择的模板从模板管理器中导出，并存储到用户选择的文件夹下；
- 导入：将模板管理器外的模板文件导入到管理器中，并将其存储到管理器的工作路径下；
- 查找：在当前分组下查找符合条件的模板文件。

4.1.3 Tab 页标签

Tab 页标签主要包括“系统”、“变量”、“站点”、“画面”和“模板管理”五部分。

4.1.4 目录显示区

4.1.4.1 目录显示区显示项

4.1.4.1.1 系统标签

点击“系统”标签项，相应的目录显示区显示“文件”、“数据库”、“设备”、“系统配置”和“SQL 访问管理器”和“Web”六大项。

- ◇ **文件：**主要包括“画面”、“命令语言”、“配方”和“非线性表”。其中命令语言又包括“应用程序命令语言”、“数据改变命令语言”、“事件命令语言”、“热键命令语言”和“自定义函数命令语言”。
- ◇ **数据库：**主要包括“结构变量”、“数据词典”和“报警组”。
- ◇ **设备：**主要包括“串口 1 (COM1)”、“串口 2 (COM2)”、“DDE 设备”、“板卡”、“OPC 服务器”和“网络站点”。
- ◇ **系统配置：**主要包括“设置开发系统”、“设置运行系统”、“报警配置”、“历史数据记录”、“网络配置”、“用户配置”和“打印配置”。
- ◇ **SQL 访问管理器：**主要包括“表格模板”和“记录体”。
- ◇ **Web：**为组态王 7.0For Internet 功能画面发布工具。

4.1.4.1.2 变量标签

点击“变量”标签项，相应的目录显示区显示变量管理，包括变量组。

4.1.4.1.3 站点标签

点击“站点”标签项，相应的目录显示区显示定义的远程站点的详细信息。

4.1.4.1.4 画面标签

点击“画面”标签项，相应的目录显示区对画面进行分组管理，创建和管理画面组。

✧ 画面分组管理

当工程中的画面比较多时，开发者查找某个画面时会有一定的困难。而且开发者往往需要建立一些内容基本相同的画面，比如各 IO 数据采集站，如果工况相同，画面上包含的图素内容就大体相同，如果每个采集站的画面都需要重新制作的话，势必会增加开发者的劳动量和劳动时间。为此，组态王 7.0 提供了画面的分组管理功能，对画面进行分组管理，开发者可以根据需要，将画面放入不同的画面组中，这样在修改和选择画面时，只需到相应的组内去找就可以了，方便了用户的使用。另外，组态王 7.0 还提供了画面的复制和剪切的功能：开发者可以对画面进行复制，在某个画面组下进行粘贴，这样就节省了开发者制作画面的时间，避免了重复劳动；开发者还可以通过组态王 7.0 提供的画面剪切—粘贴的功能，根据需要将画面从一个画面组移到另一个画面组。

✧ 建立画面组

在工程浏览器框架窗口上放置有四个标签：“系统”，“变量”，“站点”和“画面”，选

择“画面”标签，进入“画面”标签页。如图 4.37 所示。在“画面”标签页的左侧视窗中显示“画面”文件夹，在右侧的视图区显示画面。

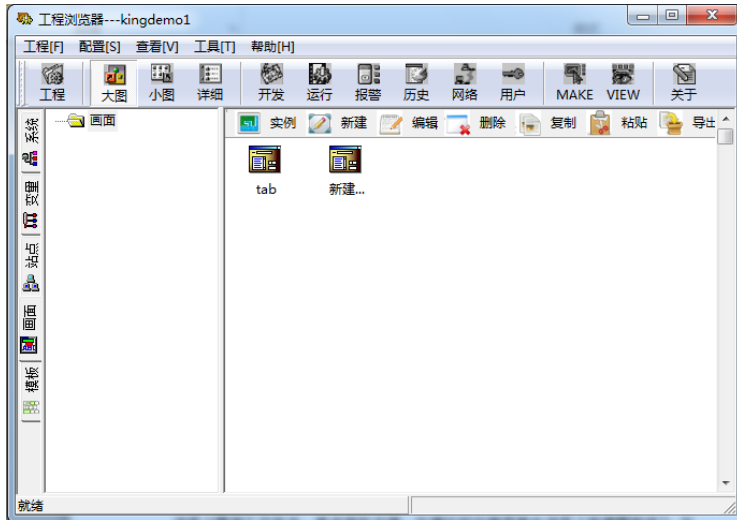


图 4.37 组态王 7.0 “画面”标签页

选择“画面”文件夹，单击鼠标右键，在弹出的右键菜单中选择“新建画面组”，在“画面”文件夹下生成一个新的画面组，并且系统自动给出默认的组名。用户可以使用这个默认的组名，也可以自己输入新的组名。

◇ 建立子画面组

在画面组下可以新建子画面组。选择某一画面组，单击鼠标右键，在弹出的右键菜单中选择“新建画面组”，则在画面组下生成一个子画面组。注意，在画面组下创建的子画面组层数不能超过 9 层，画面组在每一层最多可以创建 256 个子画面组，总的画面组数不能超过 1000 个。建立了画面组和子画面组的组态王 7.0 界面如下图 4.38。

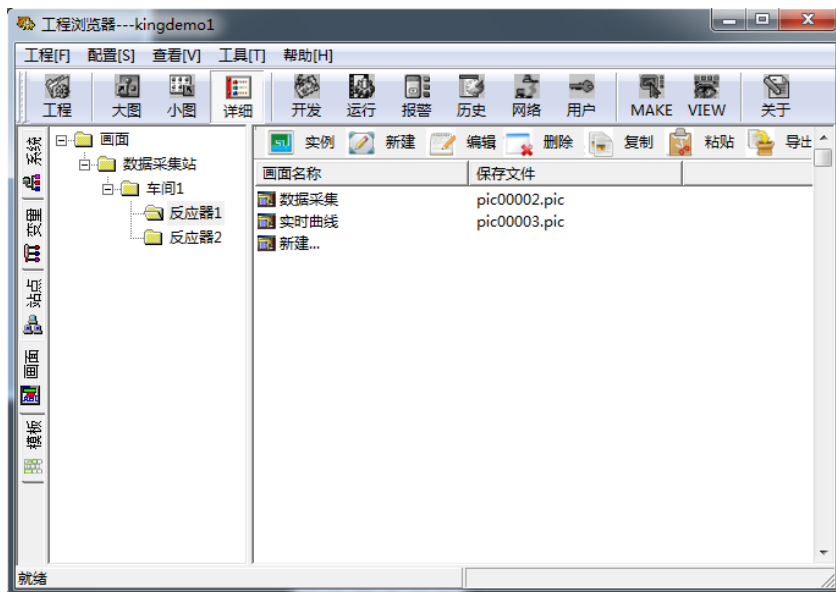


图 4.38 建立了画面组的组态王 7.0 界面

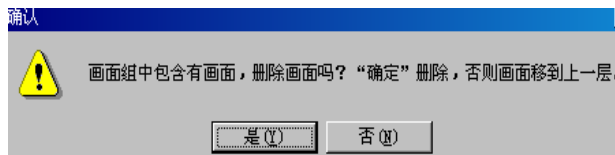
画面组和子画面组的名称要符合命名规则，如果输入的组名称错误，系统会弹出对话框“名称无效”，提示组名称错误。组名的命名规则是：

- 组态王 7.0 新建的画面组名称不能超过 31 个字符。
- 画面组名首字符只能是汉字或标准字符，不能是数字等非法字符，名称中间不容许有空格，算术符号等非法字符存在。
- 同一层下的画面组不能重名，画面组名称区分大小写。

◇ 删除画面组

组态王 7.0 可以删除不包含子画面组的画面组。当画面组下有子画面组存在时，删除该画面组，系统会提示“画面组下包含有画面组，不能删除”。必须先删除画面组下

的子画面组，该画面组才能被删除。当画面组下有画面存在时，删除该画面组，系统会弹出如下对话框，提示选择是否删除画面。



如果选择“是”，删除画面组和画面组下的画面；如果选择“否”，画面移到上一级目录下，画面组被删除。

 注意：

- 1、“画面”标签页中的“画面”文件夹不能进行更名和删除。
 - 2、不能在工程之间进行画面的复制和剪切操作。
-

4.1.4.1.5 模板标签

当点击“模板”标签项，相应的目录显示区显示“默认工作空间”下的所有模板组及相应的模板。“默认工作空间”的路径为组态王安装目录下的 Template 文件夹。当用户左键点击“默认工作空间”节点，该节点下显示所有预定义的模板组。允许用户在“默认工作空间”下再创建新的模板组及相应的模板，也允许用户在“默认工作空间”下的预定义模板组下再创建新的模板，创建后的模板将保存到默认工作空间对应的路径下。

4.1.4.2 目录显示区操作方法

✧ 打开功能配置对话框

双击功能项节点，则工程浏览器扩展该项的成员显示出来。如果选中某一个节点——如“应用程序命令语言”，在目录内容显示区中显示“请双击这儿进入<应用程...”图标，则可双击该节点，即可打开该功能的配置对话框。

✧ 扩展大纲节点

单击大纲项前面的“+”号，则工程浏览器扩展该项的成员并显示出来。

✧ 收缩大纲节点

单击大纲项前面的“-”号，则工程浏览器收缩该项的成员并只显示大纲项。

4.1.5 内容显示区

4.1.5.1 内容显示区操作方法

组态王 7.0 支持鼠标右键的操作，合理使用鼠标右键将大大提高您使用组态王 7.0 的效率。如果在工程目录显示区选中某一个成员名后（比如“画面”成员名），在目录内容显示区中显示“新建”图标，则可在目录内容显示区中的任何位置单击右键，弹出相应浮动式菜单进行操作；用户也可以在目录内容显示区中选中“新建”图标，然后双击，则弹出相应对话框。

4.1.6 状态栏

状态栏主要显示工程浏览器的状态信息。

4.2 组态王 7.0 画面开发系统

4.2.1 菜单详解

组态王 7.0 画面开发系统内嵌于组态王 7.0 工程浏览器中，又称为界面开发系统，是应用程序的集成开发环境，工程人员在这个环境里进行系统开发。

单击工程浏览器工具条“MAKE”按钮或右键单击工程浏览器空白处从显示的快捷菜单中选择“切换到 Make”命令，进入组态王 7.0 “开发系统”。如图 4.40 所示。



图 4.40 组态王 7.0 界面开发系统

此时开发系统没有画面打开，菜单栏只有“文件”和“帮助”两栏。当打开或新建一个画面时，开发系统菜单与图 4.40 显示不同。如图 4.41 所示。



图 4.41 组态王 7.0 界面开发系统菜单

4.2.1.1 文件菜单

文件菜单各命令用于对画面进行建立、打开、保存、删除等操作。若某一菜单条为灰色，表明此菜单命令当前无效，其他菜单命令为灰色时，意义相同。用鼠标单击“文件”菜单，弹出下拉式菜单。如图 4.42 所示。



图 4.42 文件菜单

1、文件\新画面

此菜单命令用于新建画面，单击“文件\新画面”菜单，则弹出“新画面”对话框，如图 4.43 所示。



图 4.43 新画面

在对话框中可定义画面的名称、大小、位置、风格，及画面在磁盘上对应的文件名。该文件名可由“组态王 7.0”自动生成，工程人员可以根据自己的需要进行修改。输入完成后单击“确定”按钮使当前操作有效，或单击“取消”按钮放弃当前操作。

- ✧ **画面名称** 在此编辑框内输入新画面的名称，画面名称最长为 20 个字符。如果在画面风格里选中“标题杆”选择框，此名称将出现在新画面的标题栏中。
- ✧ **对应文件** 此编辑框输入本画面在磁盘上对应的文件名，也可由“组态王 7.0”自动生成缺省文件名。工程人员也可根据自己需要输入。对应文件名称最长为 8 个字符。画面文件的扩展名必须为“.pic”。

- ◇ **注释** 此编辑框用于输入与本画面有关的注释信息。注释最长为 49 个字符。
- ◇ **画面位置** 输入六个数值决定画面显示窗口位置、大小和画面大小。
- ◇ **左边、顶边** 左边和顶边位置形成画面左上角坐标。
- ◇ **显示宽度、显示高度** 指显示窗口的宽度和高度。以像素为单位计算。
- ◇ **画面宽度、画面高度** 指画面的大小，是画面总的宽度和高度，总是大于或等于显示窗口的宽度和高度。

可以通过对画面属性中显示窗口大小和画面大小的设置来实现组态王 7.0 的大画面漫游功能。大画面漫游功能也就是组态王 7.0 制作的画面不再局限于屏幕大小，可以绘制任意大小的画面，通过拖动滚动条来查看，并且在开发和运行状态都提供画面移动和导航功能。

画面的最大宽度和高度为 8000×8000，最小宽度和高度为 50×50。如指定的画面宽度或高度小于显示窗口的大小，则自动设置画面大小为显示窗口大小。画面的显示高度和显示宽度设置分别不能大于画面的高度和宽度设置。

当定义画面的大小小于或者等于显示窗口大小时，不显示窗口滚动条；当画面宽度大于显示窗口宽度时显示水平滚动条；当画面高度大于显示窗口高度时，显示垂直滚动条。可用鼠标拖动滚动条，拖动滚动条时画面也随之滚动。当画面滚动时，如选择“工具\显示导航图”命令，则在画面的右上方有一个小窗口出现，此窗口为导航图，在导航图中标志当前显示窗口在整个画面中相对位置的矩形也随之移动。如图 4.44 所示。

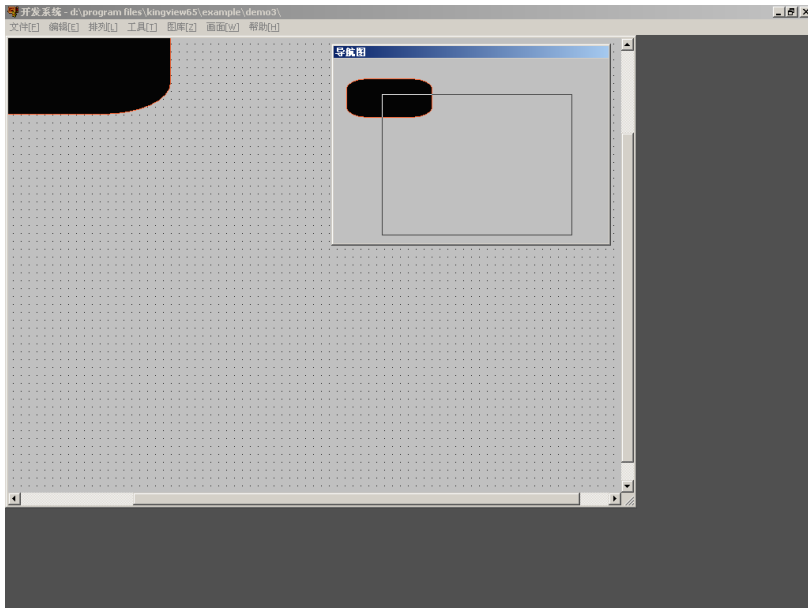


图 4.44 大画面漫游功能

组态王 7.0 开发系统会自动记录滚动条的位置，也就是说当下次再切换到此画面时，仍然是上次编辑的状态。当工程关闭后，再打开时仍然保持关闭前的状态。

通过鼠标拖动画面右下角可设置画面显示窗口大小，拖动画面左上角可设置显示窗口的位置。当显示窗口大小拖动后大于画面大小时，画面大小自动设置为显示窗口大小。

通过鼠标拖拉画面右下角，并同时按下 Ctrl 键可设置画面显示窗口和画面实际大小相等，以显示窗口的大小为准。

- ✧ **画面风格 标题杆** 此选择用于决定画面是否有标题杆。若有标题杆，选中此选项在其前面的小方框中有“√”号显示，开发系统画面标题杆上将显示画面名称。

- ◇ **画面风格 大小可变** 此选择用于决定画面在开发系统（TouchExplorer）中是否可由工程人员改变大小。改变画面大小的操作与改变 WINDOWS 窗口相同。鼠标挪动到画面边界时，鼠标箭头变为双向箭头，拖动鼠标，可以修改画面的大小。
-



注意：

修改画面大小时，如果不按下<Ctrl>键，则画面只改变显示大小，不改变画面本身的大小。如果同时按下<Ctrl>键，则同时保持画面显示大小与画面被拖动后的大小一致。

- ◇ **画面风格 类型** 主要指在运行系统中，有三种画面类型可供选择。
 - **“覆盖式”**：新画面出现时，它重叠在当前画面之上。关闭新画面后被覆盖的画面又可见；
 - **“替换式”**：新画面出现时，所有与之相交的画面自动从屏幕上和内存中删除，即所有画面被关闭。建议使用“替换式”画面以节约内存；
 - **“弹出式”**：“弹出式”画面被打开后，始终显示为当前画面，只有关闭该画面后才能对其它组态王 7.0 画面进行操作。“弹出式”画面的使用注意事项：
 - 画面类型选择“弹出式”时，“画面风格”下的“标题杆”选项只对开发系统起作用，也就是说，无论是否选择该项，组态王 7.0 运行系统都显示标题杆。
 - 一个组态王 7.0 工程中可以包含多个“弹出式”画面，但是在组态王 7.0 开发系统下进行运行系统主画面配置时最多只能选择一个“弹出式”

画面。在组态王运行系统中最多也只能打开一个“弹出式”画面。

- 如果运行系统打开的画面中包含“弹出式”画面，那么该“弹出式”画面始终显示为当前画面。
- 在组态王 7.0 运行系统中，如果打开了“弹出式”画面，那么运行系统的所有系统菜单都变为不可用状态，不能通过菜单或命令语言来关闭、打开、隐藏其它组态王 7.0 画面。可以通过点击“弹出式”画面标题栏上的关闭按钮或使用命令语言函数来关闭“弹出式”画面。“弹出式”画面关闭后，系统将恢复打开“弹出式”画面前的状态。注意，隐藏画面的函数 HidePicture 对“弹出式”画面无效。
- 在组态王 7.0 运行系统中，如果打开了“弹出式”画面，运行系统的关闭按钮也处于不可用状态。如果想退出运行系统，可以先关闭“弹出式”画面，也可以使用快捷键 { “ALT” + “F4” }，或者使用命令语言函数 Exit(0)。

- ◇ **画面风格 边框** 画面边框的三种样式，可从中选择一种。只有当“大小可变”选项没被选中时该选项才有效，否则灰色显示无效。
- ◇ **画面风格 背景色** 此按钮用于改变窗口的背景色，按钮中间是当前缺省的背景色。用鼠标按下此按钮后出现一个浮动的调色板窗口，可从中选择一种颜色。
- ◇ **命令语言（画面命令语言）** 根据程序设计者的要求，画面命令语言可以在画面显示时执行、隐含时执行或者在画面存在期间定时执行。如果希望定时执行，还需要指定时间间隔。单击“命令语言”按钮，弹出“画面命令语言”对话框如图 4.45 所

示。

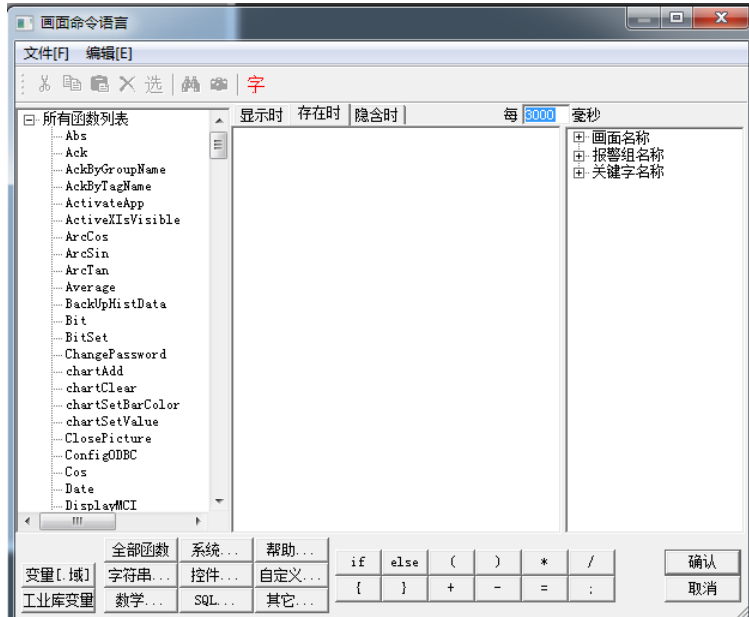


图4.45 画面命令语言

执行画面命令语言的方式有三种：**显示时**、**存在时**、**隐含时**。这三种执行方式的含义如下：

- **显示时**：每当画面由隐含变为显示时，则“显示时”编辑框中的命令语言就被执行一次。
- **存在时**：只要该画面存在，即画面处于打开状态，则“存在时”编辑框中的命令语言按照设置的频率被反复执行。
- **隐含时**：每当画面由显示变为隐含时，则“隐含时”编辑框中的命令语言就被执行一次。

命令语言的使用方法请参见“第十章 命令语言程序”。



例如：

下面使用“新画面”命令来创建一个组态画面，定义“新画面”对话框如图4.46所示。

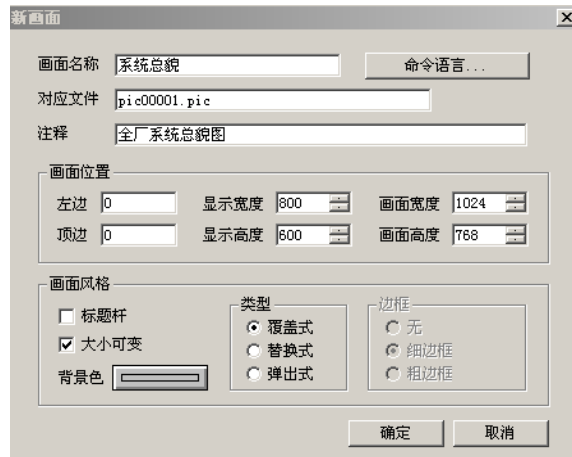


图 4.46 创建新画面

单击“确定”按钮，新建画面完成，进入“开发系统”，如图 4.47 所示。

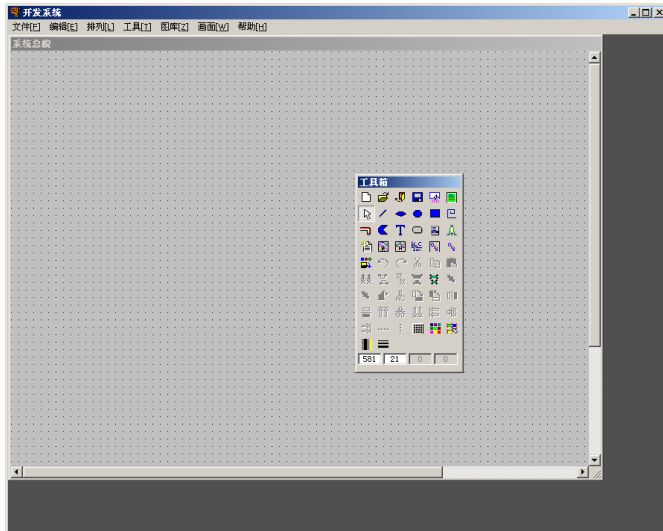


图 4.47 新画面

由于定义的画面大小为 1024×768 ，画面显示大小为 800×600 ，所以画面出现窗口滚动条。使用鼠标拖动横、纵滚动条可以显示画面的其他部分。

2、文件\打开

此菜单命令用于打开画面，单击“文件\打开”菜单，则弹出“打开画面”画面，如图 4.48 所示。

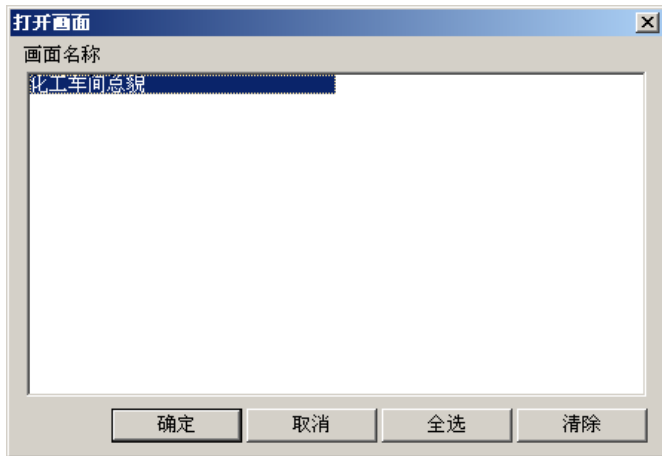


图 4.48 打开画面

在画面中显示所有当前工程路径没有打开的画面。可用鼠标或空格键选择一个或多个画面，或单击“全选”按钮选中所有画面，选中的画面加亮显示。“清除”按钮用于撤消所有选中画面。然后单击“确定”打开所有选中的画面。单击“取消”放弃当前操作。

3、文件\关闭

此菜单命令用于关闭画面，单击“文件\关闭”菜单，则弹出“关闭画面”画面，如图 4.49 所示。



图 4.49 关闭画面

画面中列出所有已经打开的画面。可以用鼠标或空格键选择其中一个或多个画面，或单击“全选”按钮选中所有画面，选中的画面加亮显示。“清除”按钮用于撤消所有选中画面。然后单击“确定”关闭所有选中的画面；单击“取消”放弃当前操作。

如果工程人员改变了画面内容而没有存入，关闭画面时将弹出对话框让工程人员选择是否存入画面，图 4.50 所示。



图 4.50 保存画面提示

4、文件\存入

此菜单命令用于保存画面，单击“文件\存入”菜单，则弹出“存入画面”画面，如图 4.51 所示。

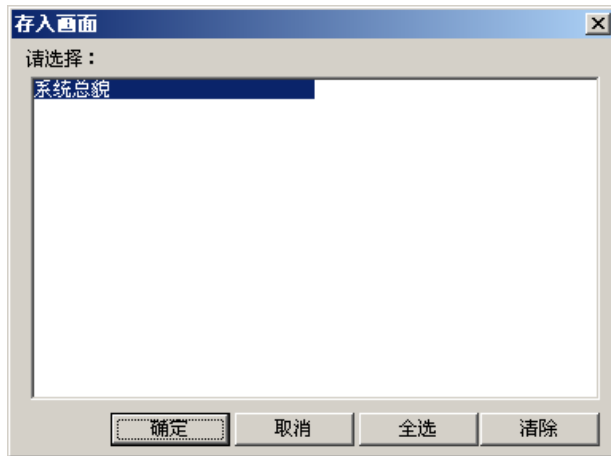


图 4.51 存入画面

列出所有已经打开并且内容发生改变的画面清单。可以用鼠标或空格键选择其中一个或多个画面，或单击“全选”按钮选中所有画面，选中的画面加亮显示。“清除”按钮用于撤消所有选中画面。单击“确定”存入所有选中的画面，单击“取消”不做保存。

5、文件\全部存

此菜单命令用于保存全部画面。单击“文件\全部存”菜单，“组态王 7.0”将所有已经打开并且内容发生改变的画面存入对应的文件。

6、文件\删除

此菜单命令用于删除画面，单击“文件\删除”菜单，则弹出“删除画面”画面，如图 4.52 所示。

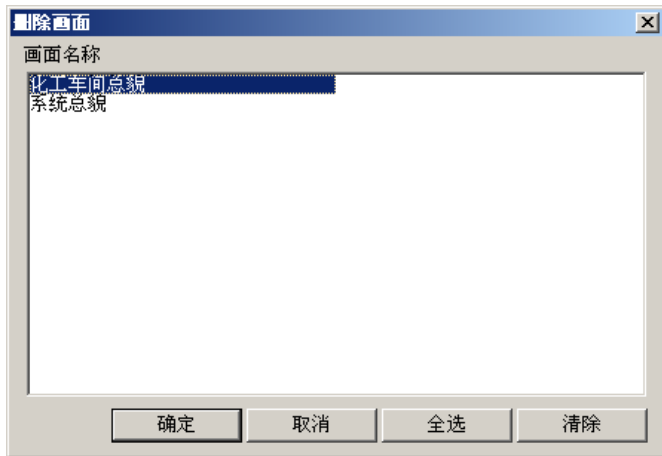


图 4.52 删除画面

列出了当前工程路径下所有画面的清单。可以用鼠标或空格键选择其中一个或多个画面，或单击“全选”按钮选中所有画面，选中的画面加亮显示。“清除”按钮用于撤消所有选中画面。单击“确定”删除所有选中的画面，单击“取消”不做删除。

组态王7.0在实际删除每个画面之前会弹出对话框，让工程人员确认是否进行删除操作，如图4.53所示。



图 4.53 删除画面确认

单击“是”或单击字母键“Y”后执行删除操作；单击“否”或单击字母键“N”后放弃删除操作。

7、文件\切换到 View

此菜单命令用于从画面制作系统直接进入画面运行系统。

8、文件\切换到 Explorer

此菜单命令用于从画面制作系统直接进入工程浏览器。

9、文件\退出

此菜单命令将组态王 7.0 开发系统制作程序最小化并回到工程浏览器。

4.2.1.2 编辑菜单

编辑菜单各命令用于对图形对象进行编辑的命令。用鼠标单击“编辑”菜单，弹出下拉式菜单。如图 4.54 所示。

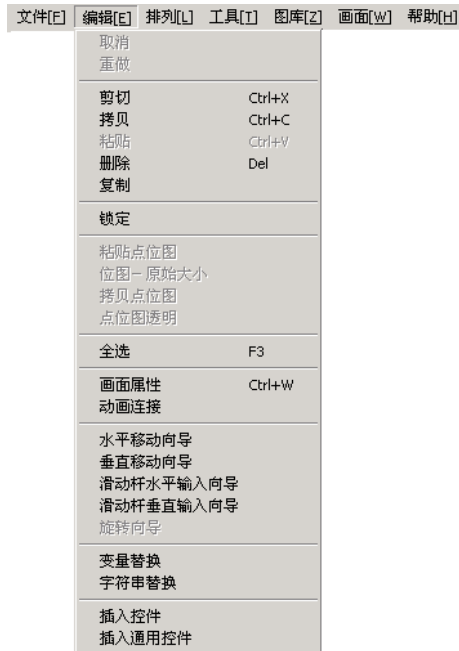


图 4.54 编辑菜单

为了使用这些命令，应首先选中要编辑的图形对象（对象周围出现 8 个小矩形），然后选择编辑菜单中合适的命令。菜单命令变成灰色表示此命令对当前图形对象无效。

1、编辑\取消

此菜单命令用于取消以前执行过的命令，从最后一次操作开始。

2、编辑\重做

此菜单命令用于恢复取消的命令，从最后一次操作开始。

3、编辑\剪切

此菜单命令将选中的一个或多个图形对象从画面中删除，并复制到粘贴缓冲区中。

剪切命令与拷贝命令的相同之处是都把当前选中的一个或多个图形对象复制到粘贴缓冲区中，不同之处是剪切命令删除当前画面中选中的一个或多个图形对象，而拷贝命令保留当前画面中选中的一个或多个图形对象，其操作方式与拷贝命令完全相同。

4、编辑\拷贝

此菜单命令将当前选中的一个或多个图形对象拷贝到粘贴缓冲区中，当选中一个或多个图形对象时（对象周围出现 8 个小矩形），灰色的拷贝命令将变为正常的显示颜色，表示此命令可对当前选中的所有图形对象进行拷贝操作，执行该命令，将把选中的图形对象拷贝到粘贴缓冲区中。



例如：

比如用鼠标选中椭圆图形对象，图形对象周围出现 8 个小矩形，点击编辑菜单，显示的下拉式菜单中部分菜单命令由灰色变成正常显示颜色，如下图 4.55 所示。此时编辑菜单下的拷贝命令有效。

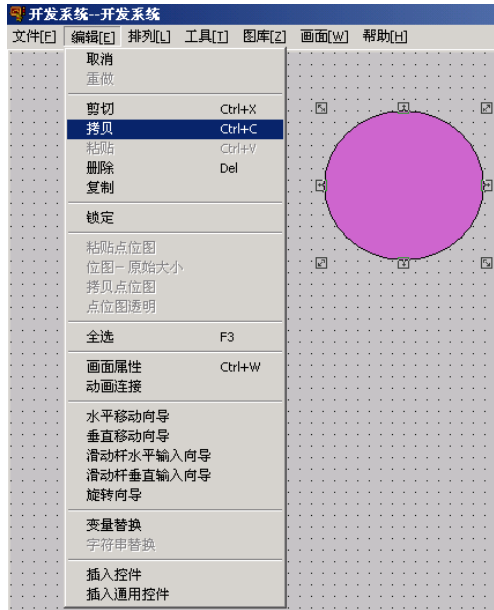



图 4.55 拷贝图形对象

 注意：


组态王 7.0 开发系统此时使用的粘贴缓冲区不是 WINDOWS 系统的裁剪版，所以图形对象的复制只能在组态王 7.0 开发系统画面制作系统内进行

5、编辑\粘贴

此菜单命令将当前粘贴缓冲区中的一个或多个图形对象复制到指定位置。只有执行了拷贝命令或剪切命令后此命令才有效，这时“粘贴”项由灰色变成正常颜色，如图 4.56 所示。



图 4.56 粘贴命令

执行粘贴命令，鼠标变成“”形状，将鼠标移到指定位置，单击左键该椭圆图形被粘贴到该位置上。

6、编辑\删除

此菜单命令用于删除一个或多个选中的图形对象。只有选中图形对象后，删除命令才由灰色变成正常颜色，此时命令有效。

7、编辑\复制

此菜单命令将当前选中的一个或多个图形对象直接在画面上进行复制，而不需要送到粘贴缓冲区中。当选中一个或多个图形对象时，灰色的“复制”命令将变为正常的显示颜色。

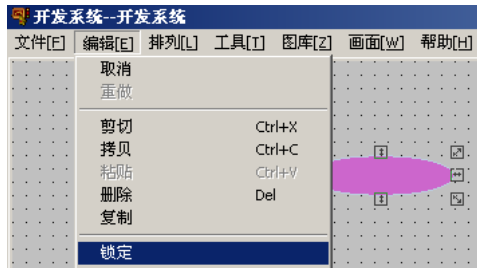
色，表示此命令可对当前选中的一个或多个图形对象进行复制操作。执行“复制”命令后，在原先图形对象上面出现一个新复制出来的图形对象。

8、编辑\锁定

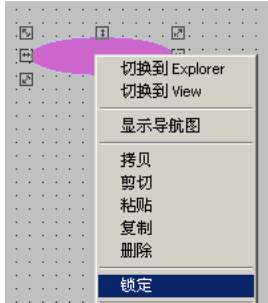
此菜单命令用于锁定，解锁图素。当图素锁定时，不能对图素的位置和大小进行操作，而复制、粘贴、删除、图素前移后移等操作不会受到影响。

图素锁定：

- ✧ **方法 1：** 打开开发系统，首先选中要锁定的图素，然后选择菜单栏的编辑/锁定，如下图所示：



- ✧ **方法 2：** 打开开发系统，首先选中要锁定的图素，点击右键，弹出浮动菜单，选择锁定，如下图所示：



画面中的图素锁定后其位置和大小将不能改变。

图素解锁：

在图素锁定后，菜单栏锁定前显示对勾，如果需要解锁，只需要再次点击锁定，对勾消失，图素被解锁。

9、编辑\粘贴点位图

此菜单命令用于将剪贴板中的点位图复制到当前选中的点位图对象中，并且复制的点位图将进行缩放以适应点位图对象的大小。组态王 7.0 中可以嵌入各种格式的图片：如 Bmp、Jpg、Jpeg、Png、gif 等。图形的颜色只受显示系统的限制。向组态王 7.0 点位图中加载图片有两种方法：

- ✧ 打开图片文件，选择所要加载的图片部分，使用“复制”命令或热键<Ctrl>+<C>将选择的图片部分复制到 Windows 的剪贴板中。在组态王 7.0 中进入开发系统画面，单击工具箱中的“点位图”按钮，在画面上绘制图片区域，然后使用“粘贴点位图”命令，将图片粘贴到组态王 7.0 画面中。
- ✧ 在组态王 7.0 的开发系统画面中，单击工具箱中的“点位图”命令在画面上绘制图

片区域。然后在该区域上单击鼠标右键弹出快捷菜单，如图 4.57 所示。



图 4.57 加载图片文件

从弹出的菜单中选择“从文件中加载”命令，弹出文件选择对话框，如图 4.58 所示。用户可以从该对话框中选择一个要加载的图片文件，单击“打开”按钮，将整个图片加载到组态王 7.0 的点位图对象中。

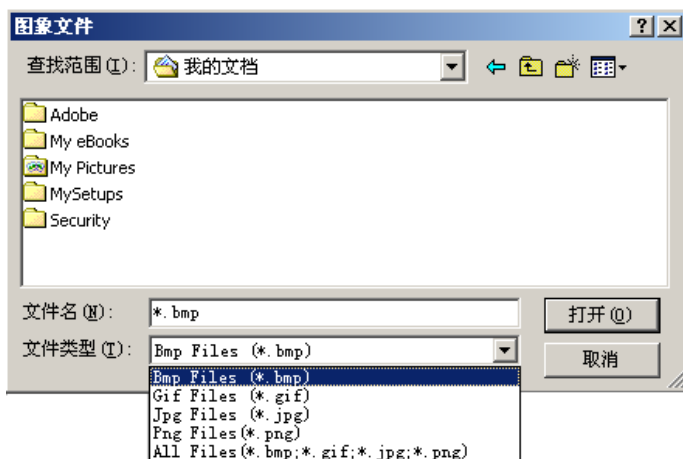


图 4.58 打开图像文件

这两种加载图片的方法各有缺点：

- ◇ 第一种方法可全部加载整个图片，也可以加载图片的某一个部分。但操作步骤较多。
- ◇ 第二种方法只能加载整个图片，但操作简便。

10、编辑\位图-原始大小

此菜单命令使选中的点位图对象中的点位图恢复到与图片本身一样的原有尺寸，而不管点位图对象矩形框的大小。点位图恢复到原有尺寸是为了避免缩放引起的图像失真。

11、编辑\拷贝点位图

此菜单命令将当前选中的点位图对象中的点位图复制到剪贴板中。只有选中点位图对象后，拷贝点位图命令才有效。

12、编辑\点位图透明

此菜单命令用于对点位图进行透明和不透明切换，在组态王 7.0 开发系统中，当从

WINDOWS 的剪切板中粘贴点位图时，设定点位图透明颜色，使用菜单命令“点位图透明”，则点位图中设定的颜色被透明，从而使被点位图覆盖的画面背景透明显示。

💡 例如：*关于点位图的各种操作*

复制粘贴点位图：

◇ 第一步：准备点位图

使用作图工具（如：Windows 的画图）画一张点位图。如图 4.59 所示。



图 4.59 准备点位图

在图上选定一部分或全部，执行<ctrl>+<c>或选择“编辑\复制”。

◇ 第二步：准备粘贴点位图

进入组态王 7.0 开发系统，单击工具箱中“点位图”图标，移动鼠标，在画面上画出一个矩形方框。如图 4.60 所示。

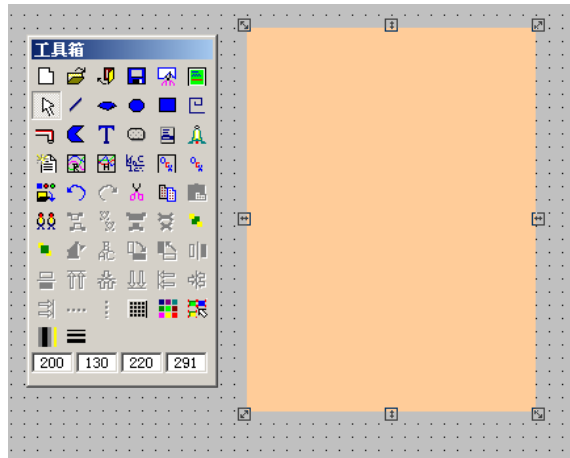


图 4.60 画点位图对象

选中该点位图对象，单击鼠标右键，弹出浮动式菜单，如图 4.61 所示。

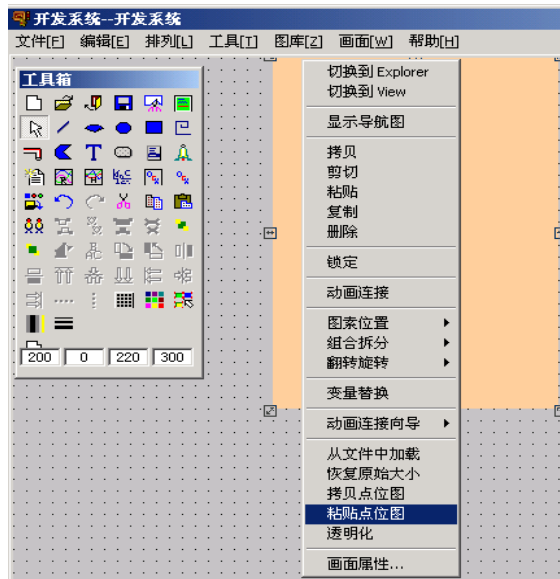


图 4.61 粘贴点位图

选择“粘贴点位图”命令即可将事先画好的图粘贴过来。如图 4.62 所示。

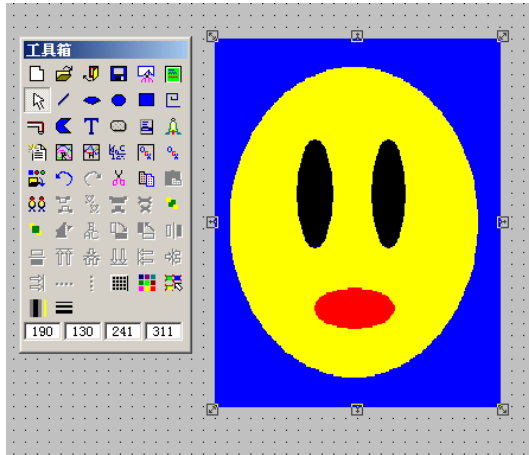


图 4.62 点位图画面

点位图透明化（视情况需要，可选操作）：

- ✧ **方法 1:** 在粘贴之前选中点位图对象，然后执行“编辑\点位图透明”或是单击右键执行“透明化”，在调色板中指定要透明化的颜色（本示例选择蓝色作为要透明化的颜色），最后粘贴点位图。如图 4.63 所示。

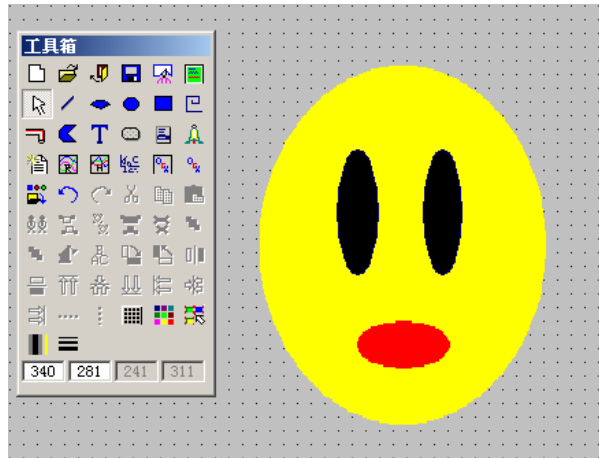


图 4.63 点位图透明 1

点位图中的蓝色透明成灰色，与背景色一样。

- ✧ **方法 2:** 首先执行点位图粘贴命令，然后执行“编辑\点位图透明”或是单击右键执行“透明化”，在调色板中指定要透明化的颜色（这次选择红色作为要透明化的颜色）。如图 4.64 所示。

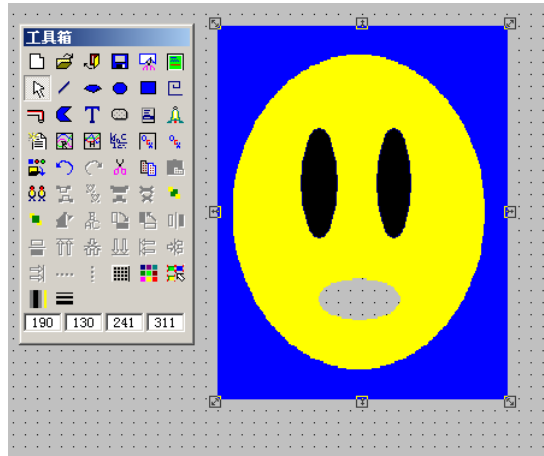


图 4.64 点位图透明 2

- ◇ **方法 3:** 可以使用调色板中的吸管吸取点位图的背景色作为窗口的背景色，实现透明化。如图 4.65 所示。

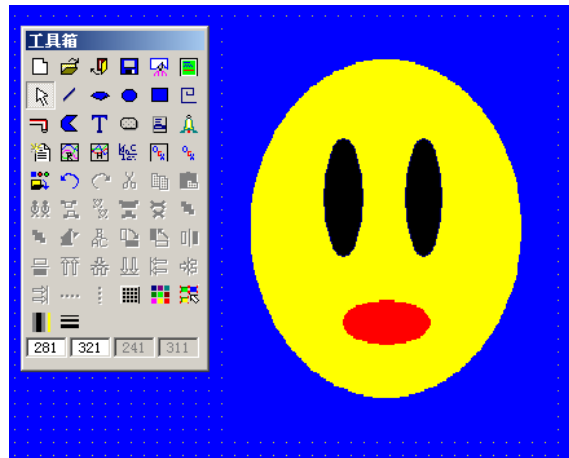


图 4.65 点位图透明 3

点位图原始大小：（视情况需要，可选操作）

- ◇ 粘贴完点位图后，执行“编辑\位图-原始大小”命令或是单击右键执行“恢复原始大小”，即可将点位图对象缩放成和图片本身一样的尺寸。如图 4.66 所示。

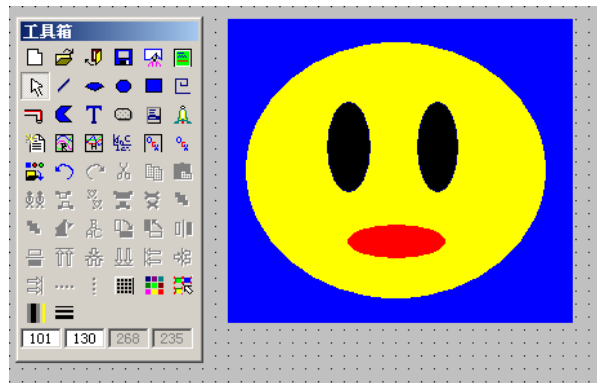



图 4.66 点位图原始大小

13、编辑\全选

此菜单命令使画面上所有图形对象都处于选中状态。

 注意：

在画面中选择图素的同时，按下<Ctrl>键可以任意多选中图素。

14、编辑\画面属性

此菜单命令用于对画面属性进行修改。单击“编辑\画面属性”菜单，则弹出“画面属性”对话框，如图 4.67 所示。该对话框与新画面对话框相同。

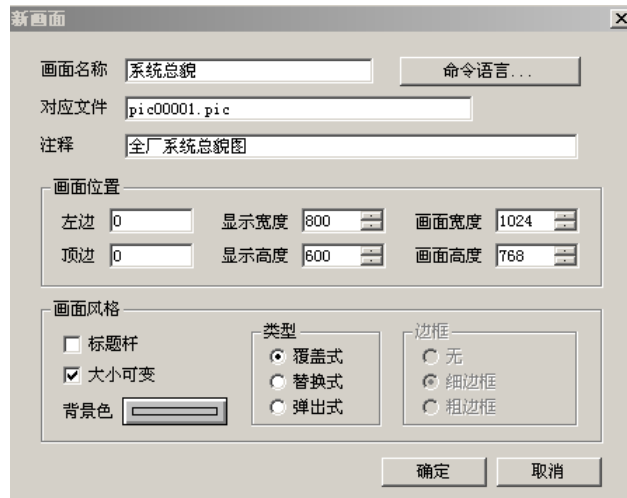


图 4.67 画面属性对话框

15、编辑\动画连接

此菜单命令用于弹出选中图形对象的动画连接对话框。在画面上选中图形对象后，单击“编辑\动画连接”菜单，弹出“动画连接”对话框。此命令的效果与双击图形对象相同。具体如何建立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

16、编辑\水平移动向导

此菜单命令用于使用可视化向导定义图素的水平移动的动画连接。在画面上选择图素，然后选择该命令，鼠标形状变为小“十”字型，选择图素水平移动的起始位置，单击鼠标左键，鼠标形状变为向左的箭头，表示当前定义的是运行时图素向左移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线。当向左移动到左边界后，单击鼠标左键，鼠标形状变为向右的箭头，表示当前定义的是运行时图素向右移动的距离，移动鼠标，箭

头随之移动，并画出一条移动轨迹线，当到达水平移动的右边界时，单击鼠标左键，弹出水平移动动画连接对话框。具体如何建立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

17、编辑\垂直移动向导

此菜单命令用于使用可视化向导定义图素的垂直移动的动画连接。在画面上选择图素，然后选择该命令，鼠标形状变为小“十”字型，选择图素垂直移动的起始位置，单击鼠标左键，鼠标形状变为向上的箭头，表示当前定义的是运行时图素向上移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线。当向上移动到上边界后，单击鼠标左键，鼠标形状变为向下的箭头，表示当前定义的是运行时图素向下移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线，当到达垂直移动的下边界时，单击鼠标左键，弹出垂直移动动画连接对话框。具体如何建立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

18、编辑\滑动杆水平输入向导

此菜单命令用于使用可视化向导定义图素的水平滑动杆输入的动画连接。在画面上选择图素，然后选择该命令，鼠标形状变为小“十”字型，选择图素水平移动的起始位置，单击鼠标左键，鼠标形状变为向左的箭头，表示当前定义的是运行时图素向左移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线。当向左移动到左边界后，单击鼠标左键，鼠标形状变为向右的箭头，表示当前定义的是运行时图素向右移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线，当到达水平移动的右边界时，单击鼠标左键，弹

出水平滑动杆输入动画连接对话框。具体如何建立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

19、编辑\滑动杆垂直输入向导

此菜单命令用于使用可视化向导定义图素的滑动杆垂直输入的动画连接。在画面上选择图素，然后选择该命令，鼠标形状变为小“十”字型，选择图素垂直移动的起始位置，单击鼠标左键，鼠标形状变为向上的箭头，表示当前定义的是运行时图素向上移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线。当向上移动到上边界后，单击鼠标左键，鼠标形状变为向下的箭头，表示当前定义的是运行时图素向下移动的距离，移动鼠标，箭头随之移动，并画出一条移动轨迹线，当到达垂直移动的下边界时，单击鼠标左键，弹出垂直滑动杆输入动画连接对话框。具体如何建立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

20、编辑\旋转连接向导

此菜单命令用于使用可视化向导定义图素的旋转的动画连接。在画面上选择图素，然后选择该命令，光标形状变为小“十”字型，在画面上相应位置单击鼠标左键，选择图素旋转时的围绕中心。随后鼠标形状变为逆时针方向的旋转箭头，表示现在定义的是图素逆时针旋转的起始位置和旋转角度。移动鼠标，环绕选定的中心，则一个图素形状的虚线框会随鼠标的移动而转动，确定逆时针旋转的起始位置后，单击鼠标左键，鼠标形状变为顺时针方向的旋转箭头，表示现在定义的是图素顺时针旋转的起始位置和旋转角度，方法同逆时针定义。选定好顺时针的位置后，单击鼠标弹出旋转动画连接对话框。具体如何建

立对象的动画连接请详细参见“第七章 图形画面和动画连接”。

21、编辑\变量替换

此菜单命令用于替换画面中引用的变量名，使该变量被替换为“数据词典”中已有的同类型的变量名。单击此选项出现“变量替换”对话框，如图 4.68 所示。



图 4.68 变量替换

- ◇ **原名称:** 输入想被替换的旧变量名，单击后面“?”按钮，弹出“选择变量名”画面，进行变量名选择；
- ◇ **替换为:** 输入新的变量名，单击后面“?”按钮，弹出“选择变量名”画面，进行变量名选择；
- ◇ **替换范围\选中的图素:** 只将当前画面中选中的图素的旧变量名替换为相应的新的变量名；
- ◇ **替换范围\当前画面:** 只将当前画面中的该旧变量名替换为相应的新的变量名；
- ◇ **替换方式:** 选择变量的替换方式。

具体变量替换方法请参见第五章 变量定义和管理。

 注意:

报表单元格和控件里引用的变量不能自动替换，需手工完成。

22、编辑\字符串替换

此菜单命令用于将画面中的文本文字、按钮上的文字进行替换。在画面上选中文本对象或按钮后，单击“编辑\字符串替换”菜单，弹出“字符串替换”对话框或“按钮属性”对话框，用以改变文本对象的内容或按钮的标识。如图 4.69 所示。

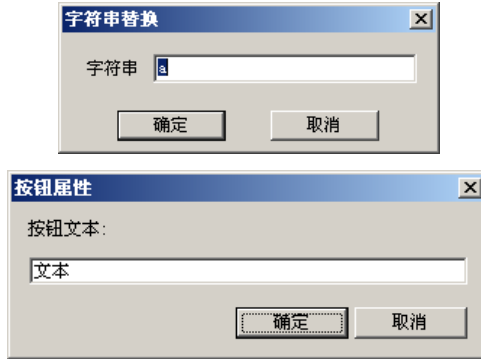


图 4.69 字符串替换、按钮属性对话框

23、编辑\插入控件

此菜单命令用于打开控件选择窗口，创建控件。单击“编辑\插入控件”菜单，弹出“创建控件”对话框，如图 4.70 所示。



图 4.70 创建控件

控件是组态王 7.0 的重要特色。它不同于 Active X 控件，是由组态王 7.0 开发的。组态王 7.0 已经提供了多种功能强大，方便实用的控件，而且正在开发更多种类的控件。在“创建控件”对话框中，您可以根据需要选择相应的控件插入到您的画面中。详细操作参见“第十四章 控件”。

24、编辑\插入通用控件

此菜单命令用于打开通用控件选择窗口，创建通用控件。单击“编辑\插入通用控件”菜单，弹出“插入控件”对话框，如图 4.71 所示。

组态王 7.0 支持 Active X 控件，“插入控件”对话框中的列表框中详细列出了本机上所有的 Active X 控件名称，用户可从中选择。Active X 控件包括标准的 Microsoft 的控件，包括用户自制的注册到 Windows 的 Active X 控件和专业厂家制作的 Active X 控件。详细操作参见“第十四章 控件”。

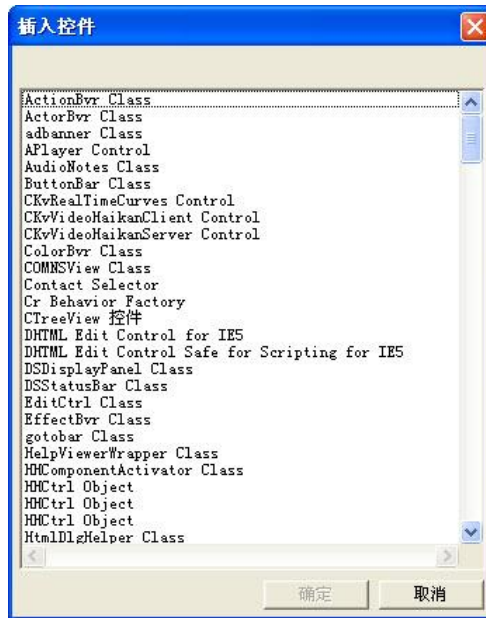


图 4.71 插入控件

4.2.1.3 排列菜单

排列菜单各命令用于调整画面中图形对象排列方式。用鼠标单击“排列”菜单，弹出下拉式菜单。如图 4.72 所示。

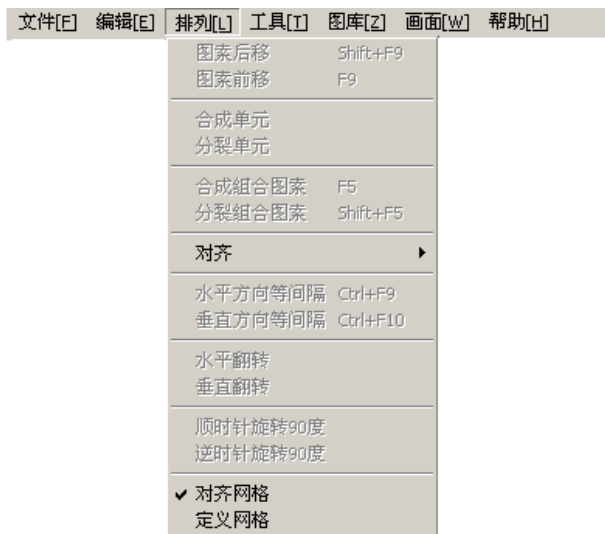


图 4.72 排列菜单

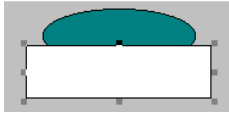
在使用这些命令之前，首先要选中需要调整排列方式的两个或两个以上的图形对象，再从“排列”菜单项的下拉式菜单中选择命令，执行相应的操作。

1、排列\图素后移

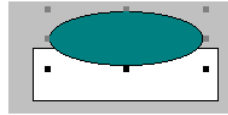
此菜单命令使一个或多个选中的图素对象移至所有其它与之相交的图素后面，作为背景。此图素后移操作命令正好是图素前移命令的相反过程，两者的使用方法完全相同。

 例如：

下图中的两个图素是相交的，矩形位于椭圆之前，如果要使矩形位于椭圆之后，则先选中矩形图素，再执行“排列\图素后移”命令。执行结果如下：



执行“图素后移”命令前的状态



执行“图素后移”命令后的状态

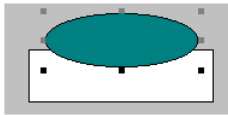
2、排列\图素前移

此菜单命令使一个或多个选中的图素对象移至所有其它与之相交的图素对象前面，作为前景。

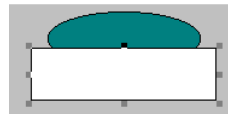


例如：

下图中的两个图素是相交的，矩形位于椭圆之后，如果要使矩形位于椭圆之前，则先选中矩形图素，再执行此“图素前移”命令。



执行“图素前移”命令前的状态



执行“图素前移”命令后的状态

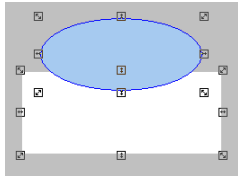
3、排列\合成单元

此菜单命令用于对所有图形元素或复杂对象进行合成，图形元素或复杂对象在合成前可以进行动画连接，合成后生成的新图形对象不能再进行动画连接。

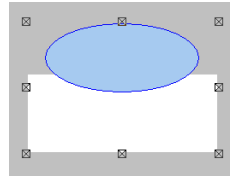


例如：

下图中有两个图素，矩形和椭圆，左图是执行合成单元命令之前同时选中两个图素的状态，右图是执行合成单元命令之后的状态。



执行“合成单元”命令前的状态



执行“合成单元”命令后的状态

4、排列\分裂单元

此菜单命令是“合成单元”命令的逆过程，把用合成单元命令形成的图形对象分解为合成前的单元，而且保持它们的原有属性不变。

5、排列\合成组合图素

此菜单命令将两个或多个选中的基本图素（没有任何动画连接）对象组合成一个整体，作为构成画面的复杂元素。按钮、趋势曲线、报警窗口、有连接的对象或另一个单元不能作为基本图素来合成复杂元素单元。合成后形成的新的图形对象可以进行动画连接。



例如：

如图 4.73 中有两个图形对象，要将这两个图素合成为一个复杂图素，先选中这两个图素，再使用“合成组合元素”命令。

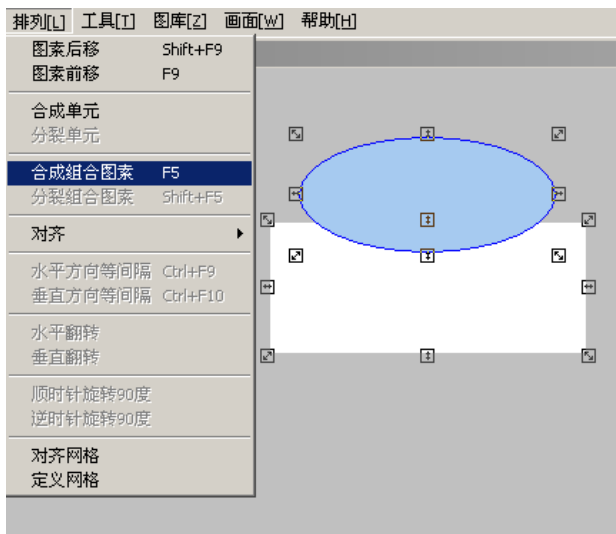


图 4.73 合成组合图素

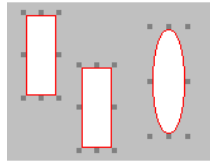
6、排列\分裂组合图素

此菜单命令将选中的单元分解成为原来合成组合图素前所用的两个或多个基本图素对象，此命令正好是合成组合图素的逆操作。执行分裂组合图素命令后原先组合图素中的动画连接会自动消失，恢复为组合前的图形对象没有任何动画连接的状态。

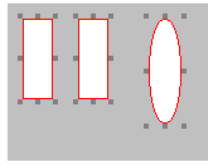
7、排列\对齐

◇ 排列\对齐\上对齐

此菜单命令使多个被选中对象的上边界与最上面的一个对象平齐。首先选中多个图形对象，然后单击“排列\对齐\上对齐”菜单。



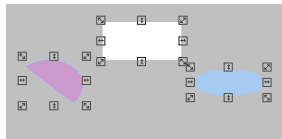
执行“上对齐”命令前



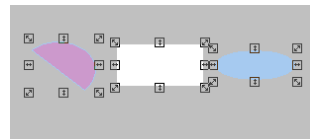
执行“上对齐”命令后

◇ 排列\对齐\水平对齐

此菜单命令使两个或多个选中对象的中心处于同一水平线上。首先选中多个图形对象，然后单击“排列\对齐\水平对齐”菜单。



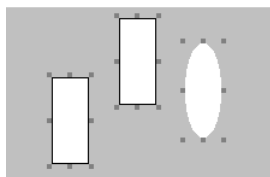
执行“水平对齐”命令前



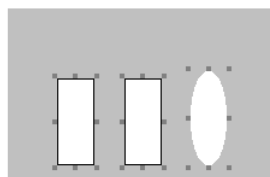
执行“水平对齐”命令后

◇ 排列\对齐\下对齐

此菜单命令使两个或多个选中对象的下边界与最下边的一个对象对齐。首先选中多个图形对象，然后单击“排列\对齐\下对齐”菜单。



执行“下对齐”命令前

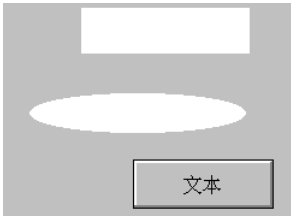


执行“下对齐”命令后

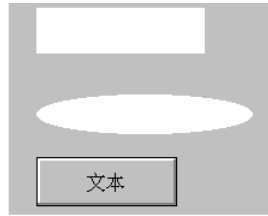
◇ 排列\对齐\左对齐

此菜单命令使两个或多个选中对象的左边界与最左的一个对象对齐。首先选中

多个图形对象，然后单击“排列\对齐\左对齐”菜单。



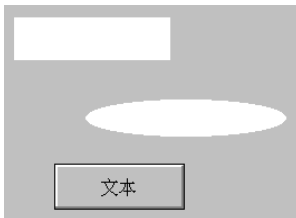
执行“左对齐”命令前



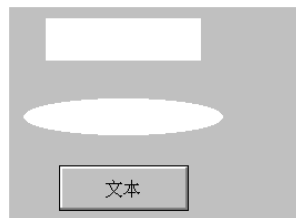
执行“左对齐”命令后

◇ 排列\对齐\垂直对齐

此菜单命令使两个或多个选中对象的中心在竖直方向对齐。首先选中多个图形对象，然后单击“排列\对齐\垂直对齐”菜单。



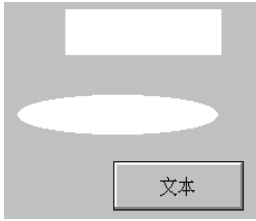
执行“垂直对齐”命令前



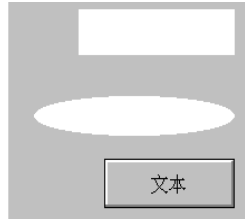
执行“垂直对齐”命令后

◇ 排列\对齐\右对齐

此菜单命令使两个或多个选中对象的右边界与最右的一个对象对齐。首先选中多个图形对象，然后单击“排列\对齐\右对齐”菜单。



执行“右对齐”命令前



执行“右对齐”命令后

8、排列\水平方向等间隔

此菜单命令使多个选中对象在水平方向上的间隔相等。首先选中多个图形对象，然后单击“排列\水平方向等间隔”菜单。



执行“水平方向等间隔”命令前



执行“水平方向等间隔”命令后


9、排列\垂直方向等间隔

此菜单命令使多个选中对象在垂直方向上的间隔相等。首先选中多个图形对象，然后单击“排列\垂直方向等间隔”菜单。



执行“垂直方向等间隔”命令前

执行“垂直方向等间隔”命令后

 注意:

水平方向等间隔与垂直方向等间隔仅限于单一图素，对组合图素不起作用。

10、排列\水平翻转

把被选中的图素水平翻转，也可以翻转多个图素合成的组合图素。翻转的轴线是包围图素或组合图素的矩形框的垂直对称轴。不能同时翻转多个图素对象。

11、排列\垂直翻转

把被选中的图素水平翻转，也可以翻转多个图素合成的组合图素。翻转的轴线是包围图素或组合图素的矩形框的水平对称轴。不能同时翻转多个图素对象。

12、排列\顺时针旋转 90 度

把被选中的单个图素以图素中心为圆心顺时针旋转 90 度，也可以旋转多个图素合成的组合图素，但是不能同时旋转多个图素对象。

13、排列\逆时针旋转 90 度

把被选中的单个图素以图素中心为圆心逆时针旋转 90 度，也可以旋转多个图素合成的组合图素，但是不能同时旋转多个图素对象。

14、排列\对齐网格

此菜单命令用于显示\隐藏画面上的网格，并且决定画面上图形对象的边界是否与栅格对齐。对齐网格后，图形对象的移动也将以栅格为距离单位。

15、排列\定义网格

此菜单命令定义网格是否显示，网格的大小以及是否需要对齐网格。单击“排列\定义网格”，弹出如图 4.74 所示对话框。

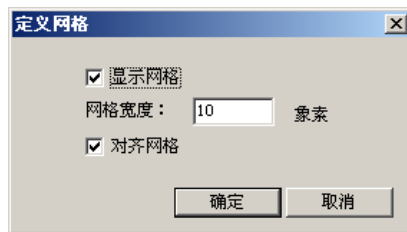


图 4.74 定义网格

选中“显示网格”时（选择框内出现“✓”号），画面背景上显示网格；选中“对齐网格”后，各图形对象的边界与栅格对齐，图形对象的移动也将以栅格为距离单位。

4.2.1.4 工具菜单

工具菜单各命令用于激活绘制图素的状态，图素包括线、填充形状（封闭图形）和文本三类简单对象和按钮、趋势曲线、报警窗口等特殊复杂图素。每种对象都有影响其外观的属性，如线颜色、填充颜色、字体颜色等，可在绘制时定义。如果选中工具菜单中的某一命令，同时在“工具”菜单命令前面出现“✓”号。用鼠标单击“工具”菜单，弹出下拉式菜单。如图 4.75 所示。

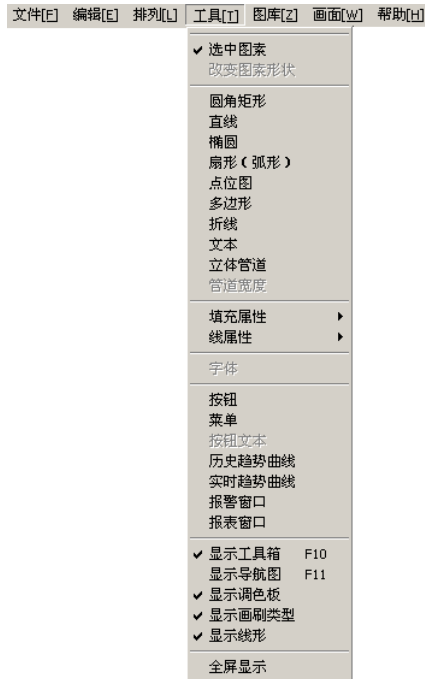


图 4.75 工具菜单

1、工具\选中图素

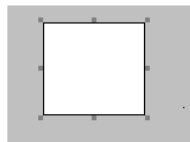
此菜单命令用于图形对象的选择、拖动和重定尺寸。这是鼠标的缺省工作方式，又是其它绘图工具完成操作后的自动返回方式。

选中对象有两种方法，操作如下：

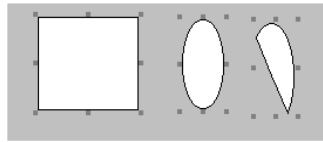
- ◇ 在对象所在区域单击鼠标左键可选中单个对象；此时按住<Ctrl>键，同时鼠标左键选中其它单个对象，可选中多个对象。
- ◇ 把鼠标置于能包围所有想选中对象的矩形的左顶点、右顶点、左下角点、右下

角点，按下鼠标的左键，拖曳鼠标出现一个虚线矩形框，使这个虚线矩形框能包围所有想要选中的对象，然后释放鼠标的左键，则虚框中的所有对象被选中。

选中的图形对象周围有 8 个灰色的小矩形，分别位于包围此对象的矩形框的四个顶点和四条边的中点。如果选中多个对象，则每个对象的周围都有八个小矩形。



选中一个图形对象



选中多个图形对象

当鼠标移动到被选中的对象上时，将变成十字形，此时按下鼠标，可拖动所有被选中的对象，到达新的位置时松开鼠标左键。当鼠标指向被选中对象周围的 8 个小矩形时，将变成双向箭头形，此时拖曳鼠标可改变对象的大小。

2、工具\改变图素形状

此菜单命令用于改变圆角矩形的圆角弧的半径、扇形或弧形的角度、多边形、直线或折线的各顶点的相对位置。用鼠标选中对象后，对象各顶点以小方框表示，其中较大的方框是拖动的焦点，用鼠标切换、拖动此焦点，可改变图素的形状。也可用键盘的 TAB 键切换焦点，或用光标键拖动这些焦点。操作方法如下：

- ◇ 用鼠标单击“改变图素形状”菜单命令，此时鼠标光标变为 V 形箭头，同时在菜单“工具\改变图素形状”命令前面出现“✓”号。
- ◇ 用鼠标单击图形对象，则对象顶点以小方框表示，共有八个方框，其中较大的方框

是拖动的焦点，选中的图形对象如下图所示。



选中一个图形对象

- ◇ 用鼠标切换、拖动焦点，可改变图素的形状。也可用键盘的 TAB 键切换焦点，或用光标键拖动这些焦点，直到工程人员满意为止。

3、工具\圆角矩形

此菜单命令用于绘制矩形或圆角矩形。单击“工具\圆角矩形”菜单，此时鼠标光标变为“十”字形，操作方法如下：

- ◇ 首先将鼠标光标置于一个起始位置，此位置就是矩形的左上角。
- ◇ 按下鼠标的左键并拖曳鼠标，牵拉出矩形的另一个对角顶点即可。在牵拉矩形的过程中矩形大小是以虚线框表示的。若需要画圆角矩形还需要选用“工具\改变图素形状”菜单方可完成。如下图所示。



4、工具\直线

此菜单命令用于绘制直线。单击“工具\直线”菜单，然后用鼠标牵拉出直线的两点

即可。操作方法如下：

- ◇ 首先将鼠标光标置于一个起始位置，此位置就是直线的起点。
- ◇ 按下鼠标左键并拖曳鼠标到达新位置，然后松开左键。直线在两点之间画出。

5、工具\椭圆

此菜单命令用于绘制椭圆（圆）。单击“工具\椭圆”菜单，然后用鼠标牵拉出矩形的两对角顶点即可画出与之相切的椭圆（圆）。操作方法如下：

- ◇ 先将鼠标光标置于一个起始位置，此位置就是矩形的左上角。注：此矩形是不可见的，所画的椭圆（圆）与此矩形内切。
- ◇ 按下鼠标的左键并拖曳鼠标，牵拉出矩形的另一个对角顶点即可。在牵拉矩形的过程中椭圆（圆）的大小是以虚线表示的。

6、工具\扇形（弧形）


此菜单命令用于绘制扇形（弧形），单击“工具\扇形（弧形）”菜单。操作方法如下：

- ◇ 首先将鼠标光标置于一个起始位置，此位置就是扇形（弧形）的左上角。
- ◇ 按下鼠标左键并拖曳鼠标，牵拉出矩形的两对角顶点即可画出与之相切的扇形。
- ◇ 若要画弧形还需选择菜单“工具\填充属性”，然后选择第二种填充方式。
- ◇ 选中“改变图素形状”菜单命令可改变扇形两边的夹角。

7、工具\点位图

此菜单命令用于绘制点位图对象。单击“工具\点位图”命令，此时鼠标光标变为“十”字形，操作方法如下：

- ◇ 首先将鼠标光标置于一个起始位置，此位置就是点位图矩形的左上角。
 - ◇ 按下鼠标的左键并拖曳鼠标，牵拉出点位图矩形的另一个对角顶点即可。在牵拉点位图矩形的过程中点位图的大小是以虚线表示的。
 - ◇ 使用绘图工具（如 WINDOWS 的画笔）画出需要的点位图，再将此点位图拷贝到 WINDOWS 的剪切板上，最后利用“组态王 7.0”的“编辑\粘贴点位图”命令将此点位图粘贴到点位图矩形内。
-

 注意：

用“点位图”菜单命令画出的点位图矩形和用“圆角矩形”菜单命令画出的矩形是不同的（尽管在外形上相同），区别方法如下：选中点位图矩形后，“编辑\粘贴点位图”命令项由灰色（禁止使用）变成亮色（允许使用）；选中矩形后，“编辑\粘贴点位图”命令项则保留为灰色（禁止使用）。

8、工具\多边形

此菜单命令用于绘制多边形，单击“工具\多边形”菜单，此时鼠标光标变为“十”字形，操作方法如下：

- ◇ 将鼠标光标置于一个起始位置，此位置就是多边形的起始点。
- ◇ 单击鼠标的左键并拖曳鼠标，牵拉出多边形的一条边，单击鼠标左键，则此条边固定下来。
- ◇ 依此方法单击鼠标左键确定多边形的各顶点，则可确定多边形的各条边。
- ◇ 最后双击鼠标左键完成多边形最后一个顶点的输入。

此命令除了画出多边形的每条边外，还要将多边形包围的区域填充颜色。

9、工具\折线

此菜单命令用于绘制折线，单击“工具\折线”菜单，此时鼠标光标变为“十”字形，操作方法如下：

- ◇ 将鼠标光标置于一个起始位置，此位置就是折线的起始点。
- ◇ 按下（或单击）鼠标的左键并拖曳鼠标，牵拉出折线的一条边，单击鼠标左键，则此条边固定下来。
- ◇ 依此方法单击鼠标左键确定折线的各顶点，则可确定折线的各条边。
- ◇ 最后双击左键完成折线最后一个顶点的输入。

此命令用于画出多条折线，对多条折线所包围的区域不进行颜色填充。

10、工具\文本

此菜单命令用于输入文字字符，单击“工具\文本”命令，此时鼠标光标变为“I”字形，输入文本的方法如下：

- ◇ 先将 I 形鼠标光标置于一个要输入文本的起始位置，单击鼠标左键，此位置就是要输入文本的起始位置。
- ◇ 用键盘输入文本字符串，单击鼠标左键结束文本输入。

若要改变字体及字体大小，还需选用“工具\字体”命令或是工具箱中的“改变字体”按钮，可以选择 Windows 系统支持的任一种字体；改变文本对象的颜色需要用调色板工具上的“文本颜色”按钮；改变文字的方向需要使用右键菜单

的“字符串竖排”或“字符串横排”项提供的功能。

11、工具\立体管道

此菜单命令用于在画面上放置立体管道图形，单击“工具\立体管道”菜单，此时鼠标光标变为“十”字形，操作方法如下：

- ◇ 将鼠标光标置于一个起始位置，此位置就是立体管道的起始点。
- ◇ 单击鼠标的左键并拖曳鼠标，牵拉出折线的一条边，单击鼠标左键，则此条边固定下来。
- ◇ 依此方法单击鼠标左键确定折线的各顶点，则可确定折线的各条边。
- ◇ 最后双击左键，则折线变为立体管道。

12、工具\管道宽度

此菜单命令用于修改画面上选中的立体管道的宽度。先选中要修改的立体管道，此时菜单命令“工具\管道宽度”由灰变亮，单击“工具\管道宽度”菜单，弹出“管道宽度”对话框，如图 4.76 所示。



图 4.76 管道宽度

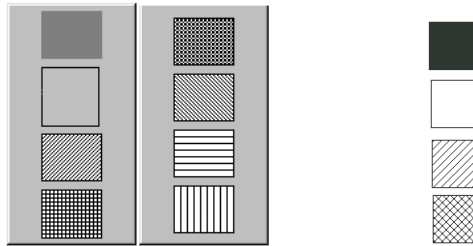
在对话框中设置管道宽度，管道内壁颜色及管道内液体流线的流动效果。只有设置了管道动画连接的“流动”属性，才能在运行系统中显示流动效果。

13、工具\填充属性

填充属性是指以何种画线方式来充满指定的封闭区域。系统提供了 8 种填充属性。如果要改变一个封闭图形的填充属性，请先选择这个封闭图形对象，然后选择菜单命令“工具\填充属性”，从中选择一种即可。其中：

- ◇ 第一种填充属性是实心填充，图形对象除边线以外的部分全部以填充色显示。
- ◇ 第二种填充属性是透明模式，在此方式下填充色不起作用。以此种方式绘制的图形对象是完全透明的，只保留边线部分。比如，矩形对象选择此模式时，将只有边框出现；圆弧对象选择此模式时，只保留弧线。

其他几种填充属性将以相应的图案填充选中的图形对象。



8 种填充属性

分别以属性 1、2、3、5 填充的矩形

14、工具\线属性

线属性包括线型和线宽，系统提供了 6 种线型、5 种线宽。如果要改变一条直线或折线的线属性，先选中这个图形对象，然后单击“工具\线属性”菜单，从中选择一种即可。



15、工具\字体

此菜单命令用于改变字体的缺省设置。操作方法如下：

- ✧ 输入文本字符串后，选中此对象，单击“工具\字体”菜单，弹出“字体”对话框，如图 4.77 所示。



图 4.77 字体

- ◇ 分别从“字体”选择框，“字形”选择框，“大小”选择框中确定工程人员需要的字体、字形、大小，再单击“确定”按钮，文本对象中的字体更改完成。

16、工具\按钮

此菜单命令用于绘制按钮。单击“工具\按钮”菜单，此时鼠标光标变为“十”字形。操作方法如下：

- ◇ 将鼠标光标置于一个起始位置，此位置就是矩形按钮的左上角。
- ◇ 按下鼠标的左键并拖曳鼠标，牵拉出矩形按钮的另一个对角顶点即可。在牵拉按钮矩形的过程中其大小是以虚线矩形框表示的，松开鼠标左键则按钮出现并固定。按钮支持“标准”，“椭圆形”，“菱形”三种类型，同时具有“透明”，“浮动”，“位图”风格。具体操作如下：

- 设置按钮类型：

选中按钮，在“按钮”上单击右键，选择“按钮类型”中的其中一种类型，如下图所示4.78所示：系统默认按钮类型为矩形。

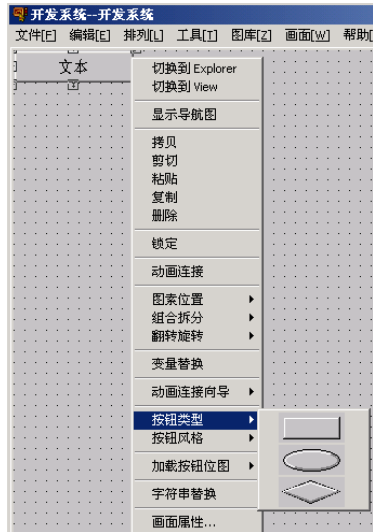


图4.78 设置按钮类型

- 设置按钮风格：

在“按钮”上单击右键，选择“按钮风格”，如图4.79所示：



图4.79 设置按钮风格

- 透明：按钮透明化，使按钮的颜色与开发系统窗口的颜色保持一致；
- 浮动：浮动只有在运行时体现。运行时按钮不显示出来，只有当鼠标移动到按钮位置时按钮才会显示出来；
- 位图：只有选择此项后，加载按钮位图命令才有效。

三种按钮风格可以同时选中，或是任意选择。

● 加载按钮位图：

选中按钮，在“按钮”上单击右键，选中加载按钮位图，如下图4.80所示。

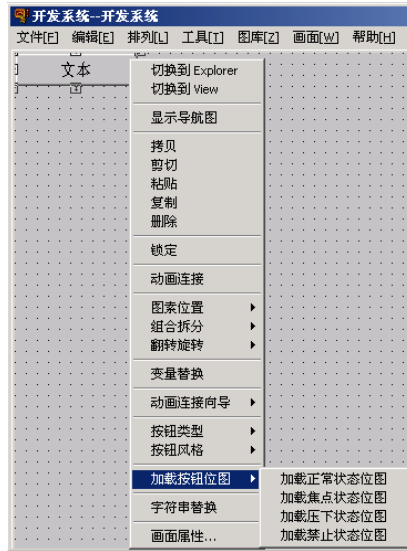


图4.80 加载按钮位图

- 加载正常状态位图：指运行时正常状态下的图形
- 加载焦点状态位图：指按钮获得焦点时显示的图形，即运行时当鼠标移动到按钮位置时显示出来的图形
- 加载压下状态位图：指运行时鼠标按下按钮时显示的图形
- 加载禁止状态位图：指运行时没有获得操作此按钮权限时显示的图形



例如：按钮加载位图

1. 从工具箱中选中“按钮”，在画面上绘制一个按钮。
2. 定义运行时执行按钮操作的命令语言：双击画面中的“按钮”，弹出“动画连接”画面，单击“命令语言连接”——“弹起时”按钮。在“命令语言”窗

口输入：“ShowPicture(“历史趋势曲线”);”命令（“历史趋势曲线”是定义的存在画面）。点击“确认”完成命令语言定义。

3. 右键单击画面中的“按钮”，从快捷菜单中选择“按钮风格”中的“位图”。
4. 右键单击画面中的“按钮”，从快捷菜单中选择“加载按钮位图”中的“加载正常状态位图”，弹出“图像文件”选择窗口，从中选择要加载的图片。
5. 依照上面步骤 4 所示，分别加载“加载焦点状态位图”、“加载压下状态位图”、“加载禁止状态位图”。

保存画面，切换到运行系统，就可以看到正常状态、焦点状态、按钮压下状态和禁止状态下按钮位图的显示情况。



注意：

按钮只有定义动画连接（如按钮命令语言）后，按钮风格和按钮位图才会在系统运行时有用。

17、工具\菜单

此菜单命令允许用户将经常要调用的功能做成菜单形式，方便用户管理，并且对该菜单可以设置权限，提高系统操作的安全性。单击“工具\菜单”菜单，鼠标光标变为“十”字形，操作方法如下：

- ✧ 首先将鼠标光标置于一个起始位置，此位置就是矩形菜单按钮的左上角。
- ✧ 按下鼠标的左键并拖曳鼠标，牵拉出菜单按钮的另一个对角顶点即可。在牵拉矩形菜单按钮的过程中其大小是以虚线矩形框表示的。松开鼠标左键则菜单出

现并固定，如图 4.81 所示。



图 4.81 绘制出的自定义菜单

- ◇ 菜单定义。绘制出菜单后，更重要的是对菜单进行功能定义，即定义菜单下的各功能项及其功能。双击绘制出的菜单按钮或者在菜单按钮上单击右键，选择“动画连接”，将弹出“菜单定义”对话框，如图 4.82 所示。



图 4.82 菜单定义对话框

- **菜单文本**：定义主菜单的名称，用户可以输入任何文本，包括空格，字符长度不能超过 31 个字符。
- **菜单项**：定义各个子菜单的名称。菜单项定义为树形结构，用户可以将各个功能做成下拉菜单的形式，运行时，通过点击该下拉菜单完成用户需要的功能。自定义菜单支持到二级菜单。每级菜单最多可定义 255 个项或子项，两级菜单名都可输入任何文本，包括空格，字符长度不能超过 31 个字符。两级菜单定义方法如下。
 - **一级菜单**：用鼠标点击“菜单项”下的编辑框，出现快捷菜单命令，如图 4.83 所示。



图 4.83 定义一级菜单

选择“新建项”命令，菜单项内出现输入子菜单名称状态，即可新建第一级子菜单。当输入完一项时，按下回车键或是单击鼠标左键即可完成新建项输入。或是直接使用快捷键<Ctrl>+N，也可出现输入第一级子菜单名称。如图 4.84 所示，建立一个“画面”菜单按钮，新建“打印”一级菜单项。



图 4.84 菜单

- **二级菜单：**用鼠标选中想要新建子菜单的一级菜单，单击右键，出现快捷菜单命令，如图 4.85 所示。



图 4.85 定义二级菜单

选择“新建子项”命令，菜单项内出现输入子菜单名称状态，即可新建第二级

子菜单。当输入完一项时，按下回车键或是单击鼠标左键即可完成新建项输入。或是直接使用快捷键<Ctrl>+U，也可出现输入第二级子菜单名称。如图 4.86 所示，在“打印”菜单项中建立两个二级菜单：“打印实时数据报表”和“打印历史数据报表”。



图 4.86 菜单

还可以对已经建立好的各个菜单进行修改。选中想要修改的菜单，单击鼠标左键弹出快捷菜单，其中有“编辑(E)”和“删除(D)”两个命令。除此之外，组态王 7.0 还提供了快捷键对菜单进行修改，如下：

- <Ctrl>+N: 新建菜单项
- <Ctrl>+U: 新建子菜单项
- <Ctrl>+E: 编辑当前选中的菜单(子)项
- <Ctrl>+D: 删除当前选中的菜单(子)项

如图 4.87 所示，定义一个自定义菜单，有两个一级菜单，分别为“打印”、“页面设置”，其中“打印”菜单中又有两个二级菜单，分别为“打印实时数据报表”、“打印历史数据报表”。



图 4.87 菜单

- **命令语言：**自定义菜单就是允许用户在运行时点击菜单各项执行已定义的功能。点击“命令语言”按钮可以调出“命令语言”界面，在编辑区书写命令语言来完成菜单各项要执行的功能。该命令实际是执行一个系统函数 `void OnMenuClick(LONG MenuIndex, LONG ChildMenuIndex)`；函数的参数为：
MenuIndex：第一级菜单项的索引号；ChildMenuIndex：第二级菜单项的索引号。
当没有第二级菜单项时，在命令语言中条件应为 `ChildMenuIndex==-1`。在命令语言编辑区中按照工程需要对 MenuIndex 和 ChildMenuIndex 的不同值定义不同的功能。MenuIndex 和 ChildMenuIndex 都是从等于 0 开始，MenuIndex==0 表示：

一级菜单中的第一个菜单; ChildMenuIndex==0 表示: 所属一级菜单中的第一个二级菜单。如图 4.88 所示。

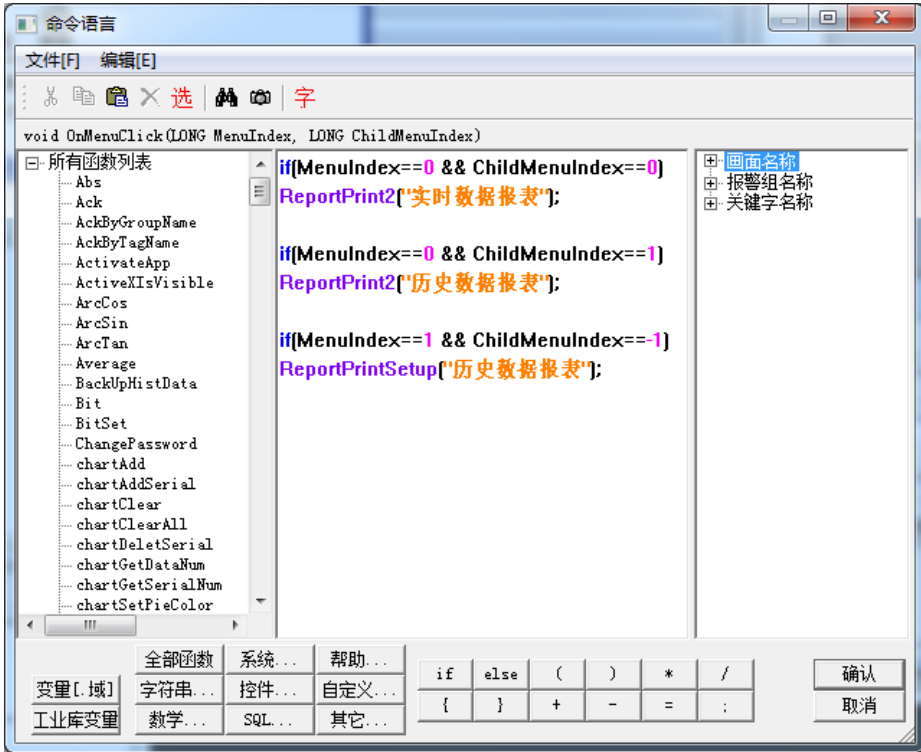


图 4.88 通过命令语言定义菜单各项的功能

该命令语言的含义为:

- 1) 当点击下拉菜单的第一项“打印”时, 继续弹出下一级子菜单, 点击子菜单第一项“打印实时数据报表”时: 执行函数 ReportPrint2(“打印实时数据报表”), 即打印实时数据报表。

2) 当点击“打印”子菜单第二项“打印历史数据报表”时：执行函数 ReportPrint2(“打印历史数据报表”), 即打印历史数据报表。

3) 当点击下拉菜单的第二项“页面设置”时, 执行函数 ReportPageSetup(“历史数据报表”), 即设置历史数据报表页面。

- **安全性:** 定义菜单按钮运行时的权限, 即没有授权的用户不可以操作该菜单按钮, 不能执行菜单各项功能。
- **权限:** 在权限文本框中输入菜单按钮的操作优先级, 范围为 1~999。
- **安全区:** 单击右侧的按钮, 弹出“选择安全区”画面, 选择该菜单按钮的操作安全区。安全区只允许选择, 不允许直接输入, 防止输入错误。

18、工具\按钮文本

此菜单命令用于修改按钮上的文本显示。只有选中按钮对象时, “按钮文本”菜单命令才由灰色(禁止使用)变成亮色(允许使用), 表示此菜单命令有效, 操作方法如下:

- 单击“工具\按钮文本”菜单, 弹出如图 4.89 对话框:



图 4.89 修改按钮文本

- 在对话框的“按钮文本”编辑框中输入文本字符串, 然后单击“确定”按钮。

 注意:


在组态王 7.0 中, 按钮和所有基本图素都支持按钮提示信息文本, 该文本的定义在动画连接对话框中定义。

19、工具\历史趋势曲线

此菜单命令用于绘制历史趋势曲线。历史趋势曲线可以把历史数据直观地显示在一张有格式的坐标图上。一个历史趋势曲线对象可同时为八个数据变量绘图, 每个画面中可绘制数目不限的历史趋势曲线对象。操作方法如下:

- 单击“工具\历史趋势曲线”菜单, 此时鼠标光标变为“十”字形, 同时此菜单左边出现“✓”号。
- 将鼠标光标放于一个起始位置, 此位置就是历史趋势曲线矩形区域的左上角。
- 再用鼠标牵拉出一个矩形, 历史趋势曲线就在这个矩形中绘出。以后选中此对象还可以移动或改变大小。

在生成历史趋势曲线后, 双击对象可弹出“历史趋势曲线”对话框, 以定义历史趋势曲线的主要属性。详见“第八章 趋势曲线和其他曲线”。

 注意:

这里的历史曲线为创建个性化的历史曲线, 与图库中提供的系统预置的图库历史曲线不同。

20、工具\实时趋势曲线

此菜单命令用于绘制实时趋势曲线。实时趋势曲线可以把数据的变化情况实时地显示在一张有格式的坐标图上，每个实时趋势曲线对象可同时为四个数据变量绘图，每个画面可绘制数目不限的实时趋势曲线对象。操作方法如下：

- 单击“工具\实时趋势曲线”菜单，此时鼠标光标变为“十”字形，同时此菜单左边出现“✓”号。
- 将鼠标光标放于一起始位置，此位置就是实时曲线矩形区域的左上角。
- 再用鼠标牵拉出一个矩形，实时趋势曲线将在此矩形中绘出。以后选中此对象还可以移动或改变大小。

在生成实时趋势曲线后，双击对象可弹出“实时趋势曲线”对话框，以定义实时趋势曲线的主要属性。详见“第八章 趋势曲线和其他曲线”。

21、工具\报警窗口

此菜单命令用于创建报警窗口。操作方法如下：

- 单击“工具\报警窗口”菜单，此时鼠标光标变为“十”字形，同时此菜单左边出现“✓”号。
- 将鼠标光标放于一个起始位置，此位置就是报警窗口矩形区域的左上角。
- 再用鼠标牵拉出一个矩形，这个虚线框矩形决定了报警窗口对象的位置和大小。以后选中此对象还可以移动或改变大小。

在生成报警窗口后，双击对象可弹出“报警窗口配置属性页”对话框，以定义报警窗口的主要属性。详见“第九章 报警和事件系统”。

22、工具\报表窗口

此菜单命令用于创建报表窗口。操作方法如下：

- 单击“工具\报表窗口”菜单，此时鼠标光标变为“十”字形，同时此菜单左边出现“✓”号。
- 光标变为小“十”字型。在画面上插入报表的位置按下鼠标左键并拖动，当画出的矩形框满足所需大小时，松开鼠标键。报表便创建成功。具体操作详见“第十七章 报表系统”。

23、工具\显示工具箱

此菜单命令用于浮动的图形工具箱在可见或不可见之间切换，工具箱缺省是可见的，可见时菜单选项左边有“✓”号。单击“工具\显示工具箱”菜单，浮动的图形工具箱在画面上消失，同时菜单选项左边“✓”号也消失。再次单击该命令工具箱又变为可见。

24、工具\显示导航图

此菜单命令用于浮动的导航图在可见或不可见之间切换，导航图缺省是不可见的，不可见时菜单选项左边没有“✓”号显示。此菜单命令只有在画面宽度（高度）大于显示窗口宽度（高度）时才有效，即在定义了大画面功能时才会有效。

单击“工具\显示导航图”菜单，浮动的导航图在画面上显示，同时菜单选项左边有“✓”号。如图 4.90 所示。再次单击该命令工具箱又变为不可见。

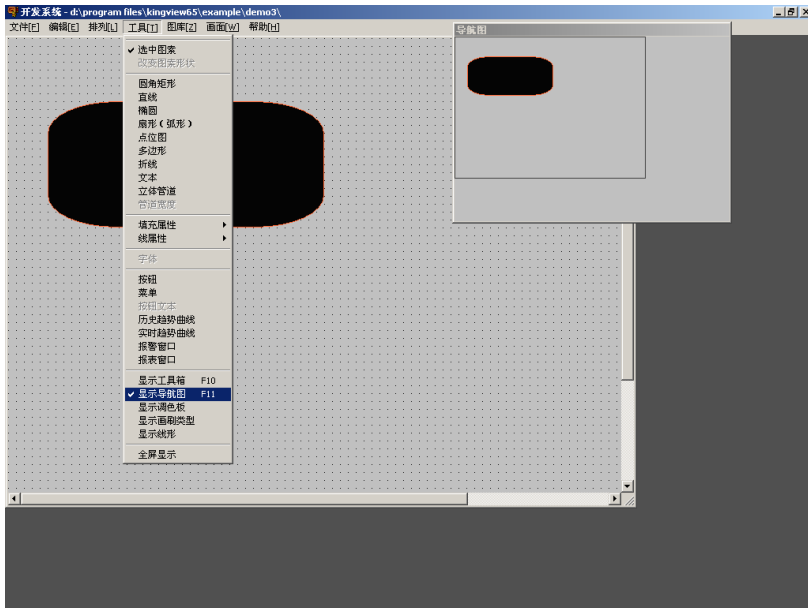


图 4.90 大画面导航图

导航图中始终显示当前编辑的画面，显示器显示的画面在整个画面中的位置在导航图中为一个标志矩形。画面中的图素在导航图中为缩小的图素。但是报警窗口、报表、组态王 7.0 控件、标准 Active X 控件不是真正缩小的图素，而只是一个标识符。

导航图的大小是固定的，当画面实际大小的长宽比例与导航图比例不一致时，靠左或上为有效区域。用鼠标左键单击导航图蓝色标题栏，同时拖动导航图，可以把导航图放在屏幕的任意位置上。

画面和导航图之间可进行互动操作：

- (1) 导航图显示时，导航图内容为当前编辑的画面。
- (2) 可用 Ctrl+Tab 键在当前打开的画面中切换，但已定义未打开的画面不会对该操作

响应，如果导航图处于显示状态，导航图内容也随画面改变而改变。

- (3) 当编辑画面内容时（绘制、修改、删除、拖动图素等），如果导航图在显示状态，画面内容也相应会改变。
- (4) 当画面滚动时，导航图中标志画面显示内容的标志矩形也随之移动。
- (5) 当在导航图中鼠标单击指定的位置时，可将当前编辑画面滚动到以导航图中单击处坐标为中心的位置上。导航图中标志当前显示位置的标志矩形也随之移动，但大小不变。
- (6) 当在导航图中单击标志矩形内部位置时，按下鼠标左键并拖动鼠标到指定位置，放开鼠标后当前编辑画面自动滚动到相应位置。
- (7) 当画面没有滚动条时，显示导航图操作将不起作用。
- (8) 当导航图在显示状态时，下列操作会使导航图自动隐藏：
 - 在画面属性中调整了画面大小，使它等于显示窗口大小。
 - 当用鼠标拖动鼠标改变显示窗口大小，使显示窗口大到和画面大小一样或大于设置的画面大小时。

大画面中对图素进行了一些处理，方便了画面操作：

- 绘制大图素时，如超出画面显示部分，画面会自动滚动，但图素大小受画面大小的限制。
- 当移动图素时，被移动的图素位置超出画面显示部分时，画面会自动移动，但移动范围受画面大小的限制。
- 当用鼠标选择图素时，选择的区域超过画面显示部分时，画面会自动滚动，但

选择区域受画面大小的限制。

- 当进行上述操作时，如导航图处于显示状态，则在导航图中也响应相应操作。
- 上述“图素”是一个广义的提法，指普通图素、组合图素、图库、组态王 7.0 控件、标准 Active X 控件。

25、工具\显示调色板

此菜单命令用于浮动的图形调色板在可见或不可见之间切换，调色板缺省是可见的，可见时菜单选项左边有“✓”号。单击“工具\显示调色板”菜单，浮动的调色板在画面上消失，同时菜单选项左边“✓”号也消失。再次单击该命令调色板又变为可见。

26、工具\显示画刷类型

此菜单命令用于浮动的画刷类型在可见或不可见之间切换，画刷类型缺省是不可见的，不可见时菜单选项左边没有“✓”号显示。单击“工具\显示画刷类型”菜单，浮动的画刷类型在画面上显示，同时菜单选项左边有“✓”号。再次单击该命令画刷类型又变为不可见。

27、工具\显示线形

此菜单命令用于浮动的线形类型在可见或不可见之间切换，线形类型缺省是不可见的，不可见时菜单选项左边没有“✓”号显示。单击“工具\显示线形”菜单，浮动的线形类型在画面上显示，同时菜单选项左边有“✓”号。再次单击该命令线形类型又变为不可见。

 注意：

工具的选择和使用请参见“第七章 图形画面和动画连接”。

28、工具\全屏显示

此菜单命令的功能与工具箱中的“全屏显示”按钮的功能相同，用于将画面开发环境整屏显示，单击“工具\全屏显示”菜单，则画面开发系统的标题栏和菜单条消失，再用鼠标左键单击工具箱中的“全屏显示”按钮，则画面开发系统的标题栏和菜单条重新出现。

4.2.1.5 图库菜单

图库菜单用于打开图库、调出图库内容、创建新图库精灵、转化图素等操作。用鼠标单击“图库”菜单，弹出下拉式菜单。如图 4.91 所示。

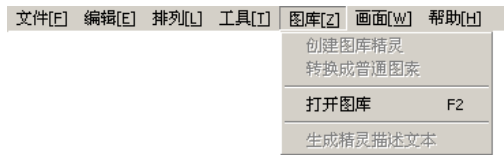


图 4.91 图库菜单

1、图库\创建图库精灵

此菜单命令用于把图素、复杂图素、单元或它们的任意组合转化为图库精灵。在画面上选中所有需要转换成图库精灵的图形对象，然后选用此命令。在弹出的对话框中输入图库精灵的名称。详细操作请参见“第十三章 图库”中“如何创建图库精灵”。

2、图库\转换成普通图素

此菜单命令的功能与“创建图库精灵”相反，用于把画面上的图库精灵分解为组成精灵的各个图形对象。详细操作请参见“第十三章 图库”中“如何将图库精灵转化为普通图素”。

3、图库\打开图库

此菜单命令用于打开图库管理器，从而可以在画面上加载各种图库精灵。单击“图库\打开图库”菜单，弹出“图库管理器”窗口，如图 4.92 所示。



图 4.92 图库管理器

从图库管理器中选择所需的图库精灵，用鼠标左键双击该图库精灵，此时图库管理器窗口从画面上消失，显示为开发系统画面窗口，此时鼠标变为“|—”形状，将鼠标移动到想要放置图库精灵的位置，单击鼠标左键，将图库精灵放置到指定位置上。

图库精灵中大部分都有连接向导或是精灵外观设置，可将精灵和数据词典中的变量

联系起来，但是也有一些精灵没有动画连接，只能作为普通图片使用。将图库精灵加载到画面上之后，双击精灵可弹出连接向导，每种精灵有各自的连接向导，一般是将组态王 7.0 的变量连接到精灵中，还有对精灵外观的设置。向导简单易用。

4、图库\生成精灵描述文本

此菜单命令用于对画面中选中的要制作图库精灵的图素生成 C 程序段的描述文本文件。该段描述文本将有助于用户用编程的方式来自制组态王 7.0 图库精灵。详细操作请参见“第十三章 图库”中“关于图库开发包”。

4.2.1.6 画面菜单

在画面菜单下方列出已经打开的画面名称，选取其中的一项可激活相应的画面，使之显示在屏幕上。当前画面的左边有“√”号。如图 4.93 所示。

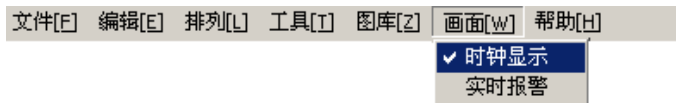


图 4.93 画面菜单

4.2.1.7 帮助菜单

此菜单命令用于查看组态王 7.0 帮助文件。如图 4.94 所示。



图 4.94 帮助菜单

1、产品帮助

选择“帮助”菜单的“产品帮助”项，打开组态王 7.0 联机帮助。

2、驱动帮助

选择“帮助”菜单的“驱动帮助”项，打开组态王 7.0 驱动帮助。

4.2.2 工具箱

4.2.2.1 简单图素对象

组态王 7.0 开发系统中的图形对象又称图素，“组态王 7.0”系统提供了矩形（圆角矩形）、直线、折线、椭圆（圆）、扇形（弧形）、点位图、多边形（多边线）、立体管道、文本等简单图素对象。利用这些简单图素对象可以构造复杂的图形画面。

4.2.2.2 复杂图素对象

组态王 7.0 画面制作系统除了可以用上面的简单图素对象组成复杂的图素以外，系统还提供了按钮、实时（历史）趋势曲线窗口、报警窗口、报表窗口等特殊的复杂图素对象。这些特殊的复杂图素把设计人员从重复的图形编程中解放出来，使他们能更专注于对象的控制。对于这些特殊的复杂图形对象以后章节中还要分别重点讲解使用方法。

4.2.3 画面

4.2.3.1 如何创建一个画面

使用工程管理器新建一个组态王 7.0 工程后，进入组态王 7.0 工程浏览器，新建组态王 7.0 画面。新建画面的方法有三种。

◇ **第一种：**在“系统”标签页的“画面”选项下新建画面。

单击工程浏览器左边“工程目录显示区”中“画面”项，右面“目录内容显示区”中显示“新建”图标，鼠标双击该图标，弹出“新画面”对话框，如图 4.34 所示。或者右键单击“新建”画面图标，弹出下图 4.35 的快捷菜单，选择“新建 画面”命令，弹出“新画面”对话框：



图 4.34 新画面

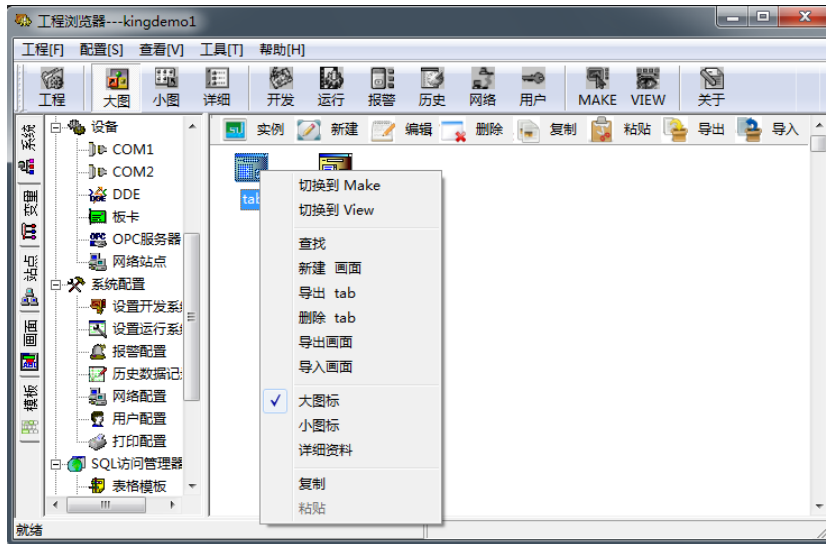


图 4.35 “新建 画面”快捷菜单

◇ **第二种：**在“画面”标签页下的画面组新建画面。

画面组建立之后，就可以在画面组中新建画面了。在画面组中新建画面有两种方式：一种是选择某一画面组，在右侧的画面视图区中选择“新建…”，单击鼠标右键，在弹出的右键菜单中选择“新建 画面”，弹出新建画面的对话框，输入画面名称，确定，新建画面完成。

另一种是选择某一画面组，在右侧的画面视图区中选择任一画面，单击鼠标右键，在弹出的右键菜单中选择“新建 画面”。在子画面组和“画面”文件夹下也可以新建画面，方法同上。

在“画面”标签页中建立的画面，均可在“系统”标签页的画面显示区中显示。选择

“画面”标签页中的“画面”文件夹，在右侧的画面视图区中看到的画面是未进行分组管理的画面。在“系统”标签页中建立的画面，也在“画面”文件夹的右侧画面视图区中显示。在画面组下建立了画面的组态王 7.0 界面如下图 4.39。

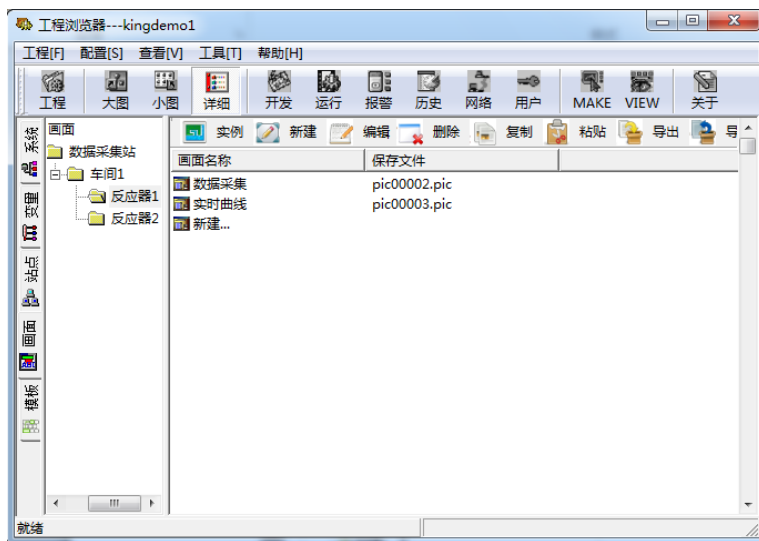


图 4.39 画面组下建立了画面的组态王 7.0 界面

- ◇ **第三种：**单击工具条“MAKE”按钮或右键单击工程浏览器空白处从显示的快捷菜单中选择“切换到 Make”命令，进入组态王 7.0 “开发系统”。选择“文件\新画面”菜单命令，弹出“新画面”对话框。如图 4.34 所示。

4.2.3.2 画面属性

画面属性参见新建画面时的介绍。

4.2.3.3 对画面的操作

1、复制粘贴画面

◇ 画面组中画面的复制和粘贴

在“画面”标签页中选择要复制的画面（可以是一个或多个画面），单击鼠标右键，在弹出的右键菜单中选择“复制”，选择目标画面组，在右侧视图区任意位置单击鼠标右键，在弹出的右键菜单中选择“粘贴”，则画面被复制到了目标画面组中。粘贴生成的新画面名称是系统自动生成的。

◇ 系统标签下画面的复制和粘贴

在“系统”标签页下画面对应的内容显示区选择要复制的画面（可以是一个或多个画面），单击鼠标右键，在弹出的右键菜单中选择“复制”，同样在内容显示区的任意位置单击鼠标右键，在弹出的右键菜单中选择“粘贴”。粘贴生成的新画面名称是系统自动生成的。

2、查找画面

在工程浏览器的工具栏中，提供了“大图”，“小图”，“详细”三种控制画面在目录内容显示区中显示方式的工具。无论以哪种方式显示，画面在目录内容显示区中都是按照画面名称顺序进行排列的。可以从排好顺序的画面中直接查找所需的画面。

另外，组态王 7.0 还提供了查找画面的工具。选择工程浏览器左边“工程目录显示区”中“画面”项，在右面“目录内容显示区”中空白处，单击右键显示快捷菜单，如图 4.35

所示，选择“查找”命令，弹出查找对话框，如图 4.36 所示。输入所要查找的画面的名称，选择查找方式——模糊查找、精确查找，然后选择是否查找到后直接编辑，选择完成后点击“确定”按钮，开始查找画面。如果查找到的话，鼠标将光标将高亮显示所查找到的第一个符合条件的画面。如果没有找到，系统会提示在系统中没有找到画面。

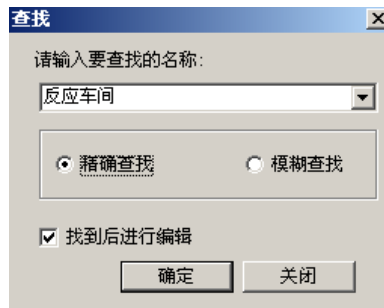


图 4.36 查找画面

各选项的含义为：

- ◇ **找到后进行编辑**：如果查找到符合条件的画面后打开画面，进入画面开发系统。
- ◇ **精确查找**：按照提供的画面名称字符全字符匹配查找。
- ◇ **模糊查找**：按照提供的字符在画面名称中含有该字符的画面，如果查找到，将在画面列表中高亮显示该画面名称。如果选择了编辑，则直接打开画面。

3、打开画面

打开画面的方式有以下几种：

- ◇ 在画面开发系统中选择“文件—》打开 (Ctrl+O)”菜单项，选择要打开的画面。
- ◇ 在工程浏览器系统或画面标签下的内容显示区，选中要打开的画面，双击打开画面。

4、关闭画面

◇ 在画面开发系统中选择“文件—》关闭”菜单项,选择要关闭的画面。

5、存入画面

◇ 在画面开发系统中选择“文件—》存入 (Ctrl+S)”菜单项,选择要保存的画面。

6、全部存画面

◇ 在画面开发系统中选择“文件—》全部存”菜单项,全部保存画面。

7、删除画面

◇ 在画面标签下画面组或者系统标签下画面对应的内容显示区选择某画面,单击鼠标右键,在弹出的右键菜单中选择“删除××”(××为画面名称),则画面组中的画面被删除。

◇ 在画面开发系统中选择“文件—》删除”菜单项,删除画面。

4.2.4 组态王 7.0 开发系统菜单命令汇总及键盘快捷键

文件菜单 ALT+F

新画面	Ctrl+N
打开	Ctrl+O
存入	Ctrl+S
删除	Ctrl+D
退出	Alt+X

编辑菜单 ALT+E

剪切	Ctrl+X
拷贝	Ctrl+C
粘贴	Ctrl+V
删除	Del
全选	F3
画面属性	Ctrl+W

排列菜单 ALT+L

图素后移	Shift+F9
图素前移	F9
合成组合图素	F5
分裂组合图素	Shift+F5
上对齐	Ctrl+F3
水平对齐	Ctrl+F7
下对齐	Ctrl+F4
左对齐	Ctrl+F5
垂直对齐	Ctrl+F8
右对齐	Ctrl+F6
水平方向等间隔	Ctrl+F9
垂直方向等间隔	Ctrl+F10

工具菜单 ATL+T

显示工具箱	F10
显示导航图	F11

图库 ALT+Z

打开图库	F2
------	----

画面 ALT+W

画面菜单	ALT+W
------	-------

帮助 ALT+H

产品帮助	F1
------	----

第五章 变量定义和管理

- ☞ 介绍组态王数据库的相关知识
- ☞ 教您如何定义和使用数据词典
- ☞ 介绍结构变量的使用
- ☞ 介绍变量组的使用

数据库是“组态王”最核心的部分。在组态王运行时，工业现场的生产状况要以动画的形式反映在屏幕上，同时工程人员在计算机前发布的指令也要迅速送达生产现场，所有这一切都是以实时数据库为中介环节，数据库是联系上位机和下位机的桥梁。

在数据库中存放的是变量的当前值，变量包括系统变量和用户定义的变量。变量的集合形象地称为“数据词典”，数据词典记录了所有用户可使用的数据变量的详细信息。

5.1 变量的类型

组态王系统中定义的变量与一般程序设计语言，比如 BASIC、PASCAL、C 语言，定义的变量有很大的不同，既能满足程序设计的一般需要，又考虑到工控软件的特殊需要。

5.1.1 基本变量类型

变量的基本类型共有两类：内存变量、I/O 变量。

◇ I/O 变量

是指可与外部数据采集程序直接进行数据交换的变量，如下位机数据采集设备（如 PLC、仪表等）或其它应用程序（如 DDE、OPC 服务器等）。这种数据交换是双向的、动态的，就是说：在“组态王”系统运行过程中，每当 I/O 变量的值改变时，该值就会自动写入下位机或其它应用程序；每当下位机或应用程序中的值改变时，“组态王”系统中的变量值也会自动更新。所以，那些从下位机采集来的数据、发送给下位机的指令，比如“反应罐液位”、“电源开关”等变量，都需要设置成“I/O 变量”。

◇ 内存变量

是指那些不需要和其它应用程序交换数据、也不需要从下位机得到数据、只在“组态王”内需要的变量，比如计算过程的中间变量，就可以设置成“内存变量”。

5.1.2 变量的数据类型

组态王中变量的数据类型与一般程序设计语言中的变量比较类似，主要有以下几种：

● 实型变量

类似一般程序设计语言中的浮点型变量，用于表示浮点（float）型数据，取值范围 $-3.40E+38 \sim +3.40E+38$ ，有效值 7 位。

● 离散变量

类似一般程序设计语言中的布尔（BOOL）变量，只有 0，1 两种取值，用于表示一些

开关量。

- **字符串型变量**

类似一般程序设计语言中的字符串变量，可用于记录一些有特定含义的字符串，如名称、密码等，该类型变量可以进行比较运算和赋值运算。字符串长度最大值为 128 个字符。

- **整数变量**

类似一般程序设计语言中的有符号长整数型变量，用于表示带符号的整型数据，取值范围（-2147483648）~2147483647。

- **结构变量**

当组态王工程中定义了结构变量时（关于结构变量的定义详见结构变量一节），在变量类型的下拉列表框中会自动列出已定义的结构变量，一个结构变量作为一种变量类型，结构变量下可包含多个成员，每一个成员就是一个基本变量，成员类型可以为：内存离散、内存整型、内存实型、内存字符串、I0 离散、I0 整型、I0 实型、I0 字符串。



注意：

结构变量的成员的变量类型必须在定义结构变量的成员时先定义（详见结构变量一节），包括离散型、整型、实型、字符串型或已定义的结构变量。在变量定义的界面上只能选择该变量是内存型还是 I0 型。

5.1.3 特殊变量类型

特殊变量类型有报警窗口变量、历史趋势曲线变量、系统预设变量三种。这几种特殊类型的变量正是体现了“组态王”系统面向工控软件、自动生成人机接口的特色。

● 报警窗口变量

这是工程人员在制作画面时通过定义报警窗口生成的,在报警窗口定义对话框中有一选项为:“报警窗口名”,工程人员在此处键入的内容即为报警窗口变量。此变量在数据词典中是找不到的,是组态王内部定义的特殊变量。可用命令语言编制程序来设置或改变报警窗口的一些特性,如改变报警组名或优先级,在窗口内上下翻页等。

● 历史趋势曲线变量

这是工程人员在制作画面时通过定义历史趋势曲线时生成的,在历史趋势曲线定义对话框中有一选项为:“历史趋势曲线名”,工程人员在此处键入的内容即为历史趋势曲线变量(区分大小写)。此变量在数据词典中是找不到的,是组态王内部定义的特殊变量。工程人员可用命令语言编制程序来设置或改变历史趋势曲线的一些特性,如改变历史趋势曲线的起始时间或显示的时间长度等。

● 系统预设变量

预设变量中有8个时间变量是系统已经在数据库中定义的,用户可以直接使用:

- **\$Year**: 返回系统当前日期的年份。
- **\$Month**: 返回1到12之间的整数,表示当前日期的月份。
- **\$Day**: 返回1到31之间的整数,表示当前日期的日。
- **\$Hour**: 返回0到23之间的整数,表示当前时间的时。
- **\$Minute**: 返回0到59之间的整数,表示当前时间的分。

- **\$Second**: 返回0到59之间的整数，表示当前时间的秒。
- **\$Data**: 返回系统当前日期字符串。
- **\$Time**: 返回系统当前时间字符串。

以上变量由系统自动更新，工程人员只能读取时间变量，而不能改变它们的值。

- **\$UserName**: 在程序运行时记录当前登录的用户的名字。
- **\$AccessLevel**: 在程序运行时记录当前登录的用户的访问权限。

以上两个变量的用法参见第九章“访问权限和口令的设置”。

- **\$StartHistoryRecord**: 表明历史记录是否启动。（1=启动；0=未启动）

工程人员在开发程序时，可通过按钮弹起命令预先设置该变量为1，在程序运行时可由用户控制，按下按钮启动历史记录。

- **\$StartAERecord**: 表明报警记录是否启动。（1=启动；0=未启动）

工程人员在开发程序时，可通过按钮弹起命令预先设置该变量为1，在程序运行时可由工程人员控制，按下按钮启动报警记录。

- **\$NewAlarm**: 每当报警发生时，“\$新报警”被系统自动设置为1。由工程人员负责把该值恢复到0。工程人员在开发程序时，可通过数据变化命令语言设置，当报警发生时，产生声音报警（用PlaySound（）函数），在程序运行时可由工程人员控制，听到报警后，将该变量置0，确认报警。如图示：

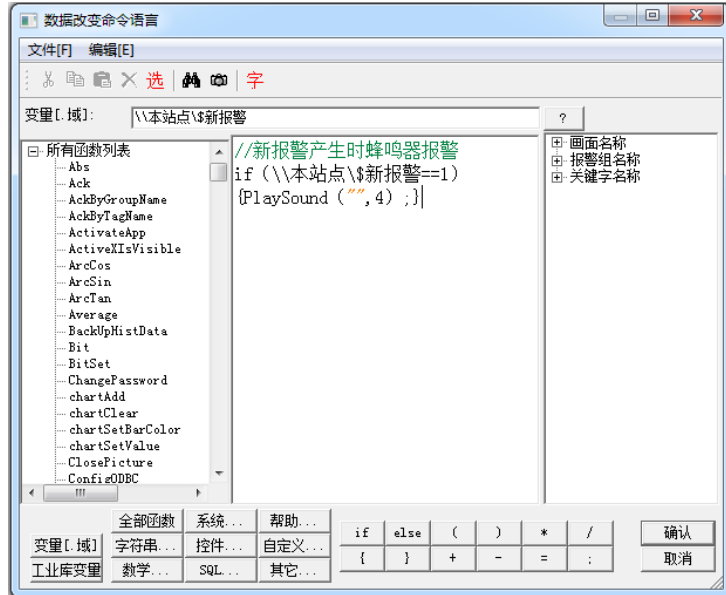


图5.1 系统变量的引用

- **\$StartSystemScripts:** 表明后台命令是否启动。(1=启动; 0=未启动)
工程人员在开发程序时, 可通过按钮弹起命令预先设置该变量为1, 在程序运行时可由工程人员控制, 按下按钮启动后台命令。
- **\$RedundantStatus:** 表明双机热备中主从计算机所处的状态, 整型 (1=主机工作正常; 2=主机工作不正常; -1=从机工作正常; -2=从机工作不正常; 0=无双机热备)。主、从机初始工作状态是由组态王中的网络配置决定的。该变量的值只能由主机进行修改, 从机只能进行监视, 不能修改该变量的值。
- **\$Millisecond:** 返回当前系统的毫秒数。(本版本中暂不支持)

- **\$NetStatus:** 用户通过引用网络上计算机的\$网络状态的变量得到网络通讯的状态。显示的数据是从0到5的数据，0代表人为的将网络中断，1到4代表网络在通过可能存在的4块网卡中的某一块进行通讯。5代表通讯故障。当此数字为1到5时用户只能将此数字改为0，中断网络通讯，其它的数字，变量不接受。但此数字为0时，用户任意输入数据，寄存器的数值将变成5，网络通讯进入尝试恢复的状态。（本版本中暂不支持）

5.2 基本变量的定义

内存离散、内存实型、内存长整数、内存字符串、I/O 离散、I/O 实型、I/O 长整数、I/O 字符串，这八种基本类型的变量是通过“变量属性”对话框定义的。

具体做法是，在工程浏览器左边的目录树中选择“数据词典”项，右侧的内容显示区会显示当前工程中所定义的变量。双击“新建”图标，弹出“定义变量”属性对话框。“组态王 7.0”的变量属性由基本属性、报警配置、记录配置三个属性页组成。采用这种卡片式管理方式，用户只要用鼠标单击卡片顶部的属性标签，则该属性卡片有效，用户可以定义相应的属性。“变量属性”对话框如下图 5.2 所示。



图 5.2 变量基本属性


“变量属性”对话框的基本属性卡片中的各项用来定义变量的基本特征，各项意义解释如下：

5.2.1 基本属性定义

- **变量名：**

唯一标识一个应用程序中数据变量的名字，同一应用程序中的数据变量不能重名，数据变量名区分大小写，最长不能超过 31 个字符。用鼠标单击编辑框的任何位置进入编辑状态，工程人员此时可以输入变量名字，变量名可以是汉字或英文名字，第一个字符不能是数字。例如，温度、压力、液位、var1 等均可以作为变量名。变量

的名称最多为 31 个字符。

 组态王变量命名规则：

变量命名时不能与组态王中现有的变量名、函数名、关键字、构件名称等重复；命名的首字符只能为字符，不能为数字等非法字符，名称中间不允许有空格、算术符号等非法字符存在。名称长度不能超过 31 个字符。

- **变量类型：**

在对话框中只能定义八种基本类型中的一种，用鼠标单击变量类型下拉列表框列出可供选择的数据类型。当定义有结构模板时，一个结构模板就是一种变量类型。

- **描述：**

用于输入对变量的描述信息。例如若想在报警窗口中显示某变量的描述信息，可在定义变量时，在描述编辑框中加入适当说明，并在报警窗口中加上描述项，则在运行系统的报警窗口中可见该变量的描述信息。（最长不超过 39 个字符）

- **变化灵敏度：**

数据类型为模拟量或整型时此项有效。只有当该数据变量的值变化幅度超过“变化灵敏度”时，“组态王”才更新与之相连接的画面显示（缺省为 0）。

- **最小值：**

指该变量值在数据库中的下限。

- **最大值：**

指该变量值在数据库中的上限。



注意：

组态王中最大的精度为 float 型，四个字节。定义最大值时注意不要超限。

- **最小原始值：**

变量为 I0 模拟变量时，驱动程序中输入原始模拟值的下限。（具体可参见组态王驱动在线帮助）

- **最大原始值：**

变量为 I0 模拟变量时，驱动程序中输入原始模拟值的上限。（具体可参见组态王驱动在线帮助）



以上四项是对 I0 模拟量进行工程值自动转换所需要的。组态王将采集到的数据按照这四项的对应关系自动转为工程值。

- **保存参数：**

在系统运行时，如果变量的域（可读可写型）值发生了变化，组态王运行系统退出时，系统自动保存该值。组态王运行系统再次启动后，变量的初始域值为上次系统运行退出时保存的值。

- **保存数值：**

系统运行时，如果变量的值发生了变化，组态王运行系统退出时，系统自动保存该值。组态王运行系统再次启动后，变量的初始值为上次系统运行退出时保存的值。

- **初始值：**

这项内容与所定义的变量类型有关，定义模拟量时出现编辑框可输入一个数值，定

义离散量时出现开或关两种选择。定义字符串变量时出现编辑框可输入字符串，它们规定软件开始运行时变量的初始值。

- **连接设备：**

只对 I/O 类型的变量起作用，工程人员只需从下拉式“连接设备”列表框中选择相应的设备即可。此列表框所列出的连接设备名是组态王设备管理中已安装的逻辑设备名。用户要想使用自己的 I/O 设备，首先单击“连接设备”按钮，则“变量属性”对话框自动变成小图标出现在屏幕左下角，同时弹出“设备配置向导”对话框，工程人员根据安装向导完成相应设备的安装，当关闭“设备配置向导”对话框时，“变量属性”对话框又自动弹出；工程人员也可以直接从设备管理中定义自己的逻辑设备名。



注意：

如果连接设备选为 Windows 的 DDE 服务程序，则“连接设备”选项下的选项名为“项目名”；当连接设备选为 PLC 等，则“连接设备”选项下的选项名为“寄存器”；如果连接设备选为板卡等，则“连接设备”选项下的选项名为“通道”。

- **项目名：**

连接设备为 DDE 设备时，DDE 会话中的项目名，可参考 Windows 的 DDE 交换协议资料。

- **寄存器：**

指定要与组态王定义的变量进行连接通讯的寄存器变量名，该寄存器与工程人员指定的连接设备有关。

- **转换方式:**

规定 I/O 模拟量输入原始值到数据库使用值的转换方式。有线性转化、开方转换、和非线性表、累计等转换方式。关于转换的具体概念和方法，请参见本章“IO 变量的转换方式”一节。

- **数据类型:**

只对 I/O 类型的变量起作用，定义变量对应的寄存器的数据类型，共有 9 种数据类型供用户使用，这 9 种数据类型分别是：

- **BIT:** 1 位；范围是：0 或 1
- **BYTE:** 8 位，1 个字节；范围是：0---255
- **SHORT:** 2 个字节；范围是：-32768---32767
- **USHORT:** 16 位，2 个字节；范围是：0---65535
- **BCD:** 16 位，2 个字节；范围是：0---9999
- **LONG:** 32 位，4 个字节；范围是：-2147483648---2147483647
- **LONGBCD:** 32 位，4 个字节；范围是：0---4294967295
- **FLOAT:** 32 位，4 个字节；范围是：-3.40E+38~+3.40E+38，有效位 7 位
- **STRING:** 128 个字符长度

各寄存器的数据类型请参见组态王的驱动帮助中相关设备的帮助。

- **采集频率:**

用于定义数据变量的采样频率。与组态王的基准频率设置有关。

- **读写属性:**

定义数据变量的读写属性，工程人员可根据需要定义变量为“只读”属性、“只写”属性、“读写”属性。

- **只读：**


对于只进行采集而不需要人为手动修改其值，并输出到下位设备的变量一般定义属性为只读；

- **只写：**

对于只需要进行输出而不需要读回的变量一般定义属性为只写；

- **读写：**

对于需要进行输出控制又需要读回的变量一般定义属性为读写。

 **注意：**

当采集频率为 0 时，只要组态王上的变量值发生变化时，就会进行写操作；
当采集频率不为 0 时，会按照采集频率周期性的输出值到设备

- **允许 DDE 访问：**

组态王内置的驱动程序与外围设备进行数据交换，为了方便工程人员用其它程序对该变量进行访问，可通过选中“允许 DDE 访问”，这样组态王就作为 DDE 服务器，可与 DDE 客户程序进行数据交换，具体操作见“第十八章 组态王与其它应用程序的动态数据交换（DDE）”。

5.2.2 报警定义

参见本手册“第九章报警和事件系统”。

5.2.3 记录和安全定义

参见本手册“第二十五章组态王历史库”。

单击“确定”按钮，则工程人员定义的变量有效时保存新建的变量名到数据库的数据词典中。若变量名不合法，会弹出提示对话框提醒工程人员修改变量名。单击“取消”按钮，则工程人员定义的变量无效，并返回“数据词典”界面。

5.3 结构变量

在工程实际中，往往一个被控对象有很多参数，而这样的被控对象很多，而且都具有相同的参数。如一个储料罐，可能有压力、液位、温度、上下限硬报警等参数，而这样的储料罐可能在同一工程中有很多。如果用户对每一个对象的每一个参数都在组态王中定义一个变量，有可能会造成使用时查找变量不方便，定义变量所耗费的时间很长，而且大多数定义的都是有重复属性的变量。如果将这些参数作为一个对象变量的属性，在使用时直接定义对象变量，就会减少大量的工作，提高效率。为此，组态王引入了结构变量的概念。

5.3.1 什么是结构变量

为方便用户快速、成批定义变量，组态王支持结构数据类型，使用结构数据类型定义结构变量。结构变量是指利用定义的结构模板在组态王中定义变量，该结构模板包含若干个成员，当定义的变量的类型为该结构模板类型时，该模板下所有的成员都成为组态王的基本变量。一个结构模板下最多可以定义 64 个成员。结构变量中结构模板允许两层嵌套，即在定义了多个结构模板后，在一个结构模板的成员数据类型中可嵌套其它结构模板数据

类型。

5.3.2 结构变量的定义

要使用结构变量，首先需要定义结构模板和结构成员及属性。在组态王工程浏览器中选择数据库下的结构变量，如图 5.3 所示。

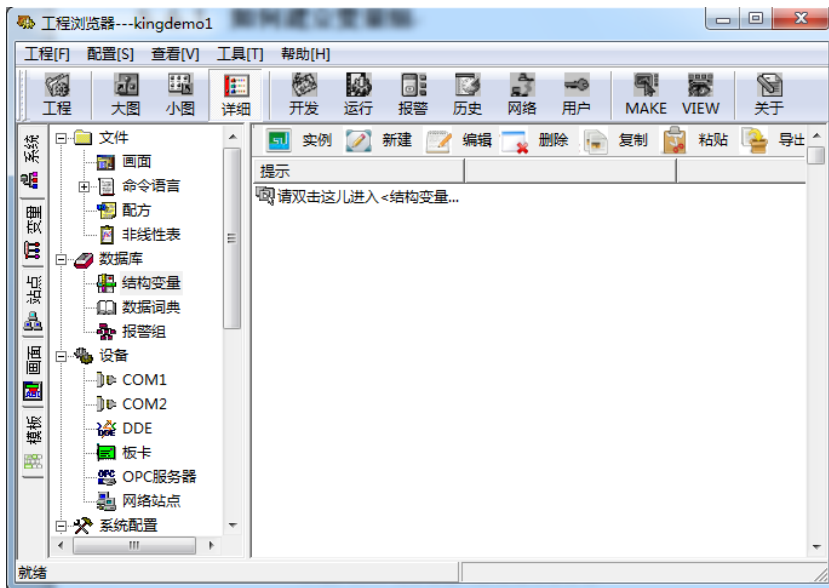


图 5.3 选择定义结构变量

双击右侧的提示图标，进入结构变量定义对话框，如下图 5.4 所示。

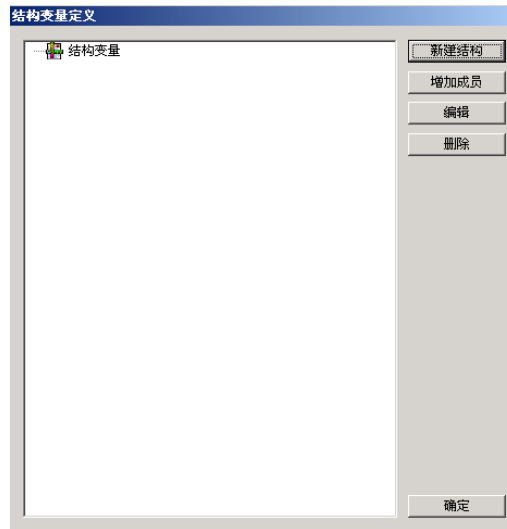


图 5.4 结构变量定义对话框

在结构变量定义对话框中有“新建结构”、“增加成员”、“修改”、“删除”几个功能。如一个储料罐具有压力、温度、物位、上限报警、下限报警等几个参数，下面以此为例来说明组态王中结构变量的定义和使用过程。

- **新建结构：**增加新的结构。单击<新建结构>按钮，弹出结构变量名输入对话框，如下图 5.5 所示。输入结构变量名称，单击<确定>按钮，在结构变量树状目录中显示出用户定义的结构模板。

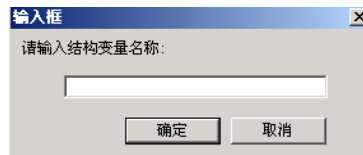


图 5.5 结构名输入对话框

⚠ 注意：结构模板的名称和成员的名称首字母不能为数字，中间不能包含空格。

命名要符合组态王变量命名规则。

如在结构名称输入对话框中输入“储料罐”，单击确定按钮，关闭对话框，则在结构变量定义界面上增加了一个新的结构，如下图 5.6 所示。

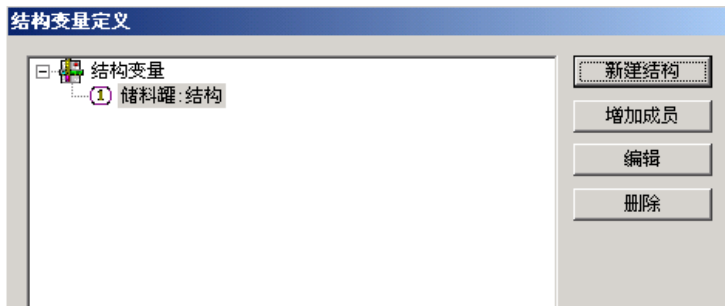


图 5.6 新增加的结构

按照上述方法，可以建立多个结构。

- **增加成员：**选中一个结构模板，如图 5.6 所示，单击<增加成员>按钮，弹出新建结构成员对话框，如图 5.7 所示。

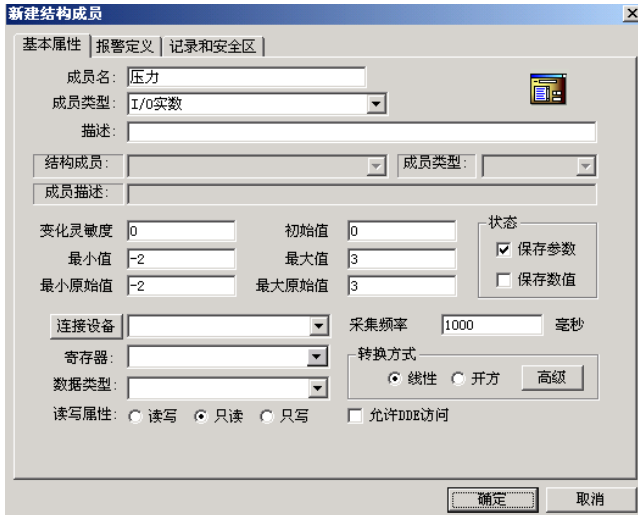


图 5.7 结构成员定义对话框

该对话框与组态王基本变量定义属性对话框相同，用户在这里可以直接定义结构成员的各种属性，如基本数值属性、IO 属性、报警属性、记录属性等。在成员名编辑框中输入成员名称。然后单击成员类型列表框，选择该成员的数据类型。另外，如果用户定义了其它结构模板，并且其它结构模板下定义了结构成员，那么，其它结构模板的名称也会出现在数据类型中，用户可以选择结构模板作为数据类型，将其嵌入当前结构模板中，如下图 5.8 所示。

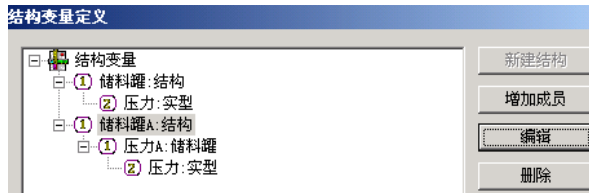


图 5.8 结构变量定义

所有属性定义与基本变量属性定义相同，这里不再细述。定义完毕后，单击<确定>按钮，关闭“结构变量定义”对话框。

按照上述方法，可以将其它成员加入到成员列表中来。定义完成后，如图 5.9 所示。如果此时确定完成，单击对话框上的“确定”按钮，关闭对话框。

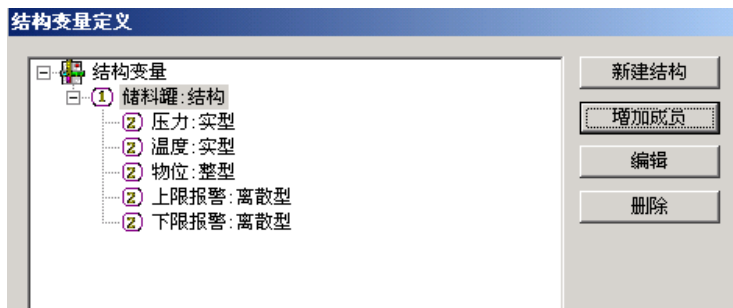
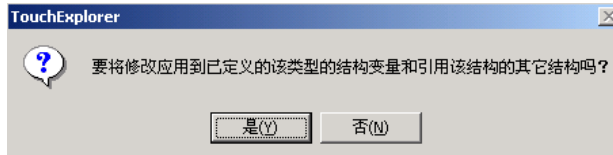


图 5.9 结构成员定义完成

- **编辑：**可以编辑结构模板和结构模板成员

编辑结构模板：选中一结构模板，单击<编辑>按钮，可以编辑结构模板名称。

编辑结构模板成员：选中一个结构成员，单击<编辑>按钮，可以编辑其成员名称、成员类型。修改成员的属性并确认后，系统会出现如下图所示的提示框，提示是否将当前成员属性的修改应用到已定义的结构变量和引用该结构的结构中，如果确认修改全部相关属性，选择“是”；如果只修改当前的成员的属性，选择“否”。




定义完成后，单击<确定>按钮，关闭对话框。

⚠ 注意：在下列情况下，结构模板或成员不能被编辑。

1. 要编辑的结构模板被其他模板成员引用，且引用结构模板的模板定义了结构变量。
2. 要编辑的结构模板已定义结构变量，且该结构变量已被使用（画面，命令语言等）
3. 要编辑的结构模板成员对应的结构模板被其他结构模板成员引用，而且引用结构模板的结构模板定义了结构变量。
4. 要编辑的结构模板成员对应的结构模板已定义了结构变量且结构变量已被使用（画面，命令语言等）。
5. 具有嵌套结构的结构模板成员。如上图中储料罐 A 下的压力。
6. 当结构模板成员类型为其他结构模板时，结构模板成员的类型不能编辑。如上图中储料罐 A 下的压力 A。

-
- **删除结构模板或成员：**选择一个结构模板或成员，单击<删除>按钮。

 注意：在下列情况下，结构模板或成员不能被删除。

1. 要删除的结构模板被其他模板成员引用
2. 要删除的结构模板已定义结构变量
3. 要删除的结构模板成员对应的结构模板被其他结构模板成员引用
4. 要删除的结构模板成员对应的模板已定义结构变量

5.3.3 结构变量的使用

- 定义结构变量类型的变量

如果定义了结构变量和成员，在数据词典中定义变量选择变量类型时，下拉列表框中除了基本的八种类型外，还会出现所有结构模板名称，一种结构模板就是一种变量类型。在组态王的工程浏览器中，单击数据库中的变量词典，单击右侧的新建，弹出变量属性对话框，如下图 5.10 所示。

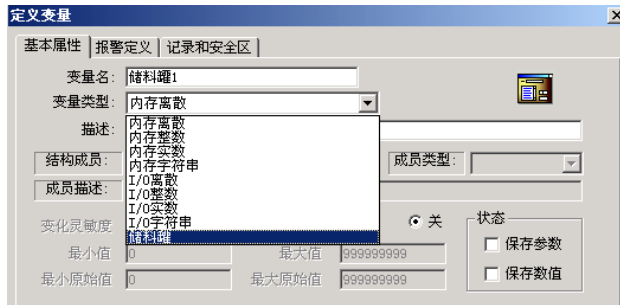


图 5.10 选择结构变量类型

在变量名中输入对象名称（或基本变量名称），在“变量类型”列表中选择刚才定

义的“储料罐”数据类型。选择完后如图 5.11 所示。

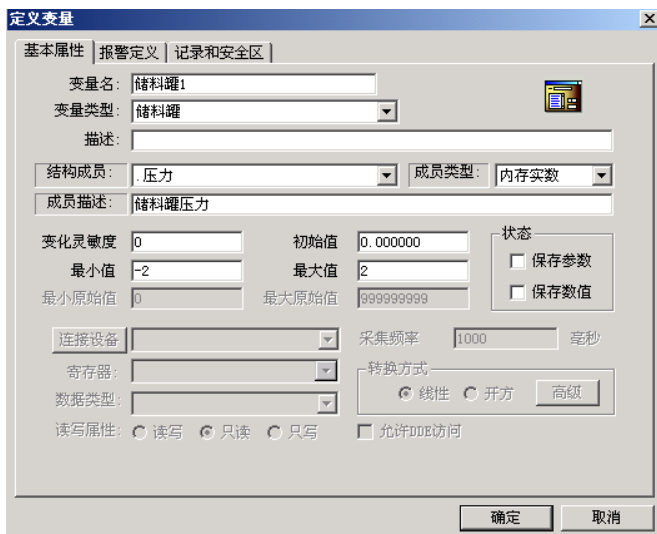


图 5.11 结构变量

在结构成员中选择该模板结构中的某一个成员，在成员类型中选择该成员的变量类型（因为其数据类型在定义结构变量时已经定义过，所以在此处只是选择内存型、I/O 型）。其余各项定义与定义组态王普通变量一致。定义完毕后，单击<确定>完成。这样，在数据词典里定义一个变量，利用结构变量，这一个变量代表很多个变量（因为一个结构中有着很多个成员）。数据词典列表中显示的结构变量的 ID 号为其最后一个成员的 ID 号，每个成员都会被自动分配一个 ID 号。

● 在工程中使用结构变量

变量表达式的格式为组态王变量属性对话框中的：变量名. 结构成员名称。在变量浏览器中选择变量如图 5.12 所示。

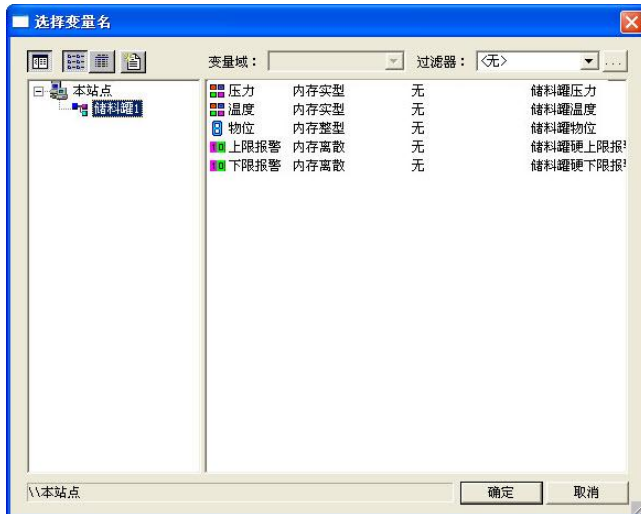


图 5.12 结构变量在变量浏览器中

在站点名称目录“本站点”下选择结构变量名称“储料罐 1”，则右边变量列表中显示所有成员变量。选择到动画连接中如图 5.13 所示。



图 5.13 结构类型变量的使用

还可以定义嵌套的结构变量，首先在结构变量定义对话框中新建一个结构，或选中已有的结构，选择“增加成员”，在定义成员的变量类型时，选择已经定义的结构，如下图 5.14 所示。

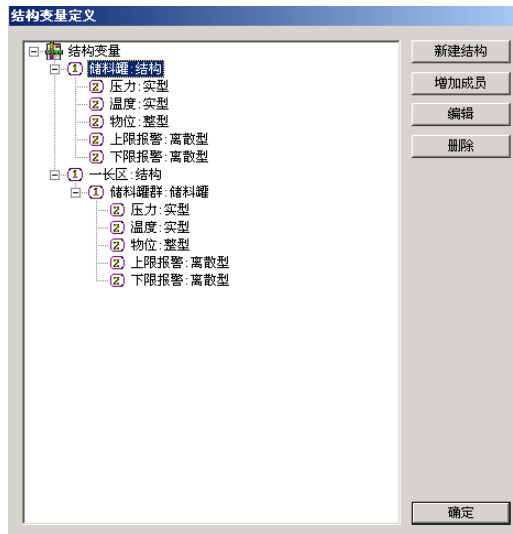


图 5.14 定义嵌套结构

新建的结构变量的数据类型选择已经定义的结构模板“储料罐”，如下图 5.15 所示。

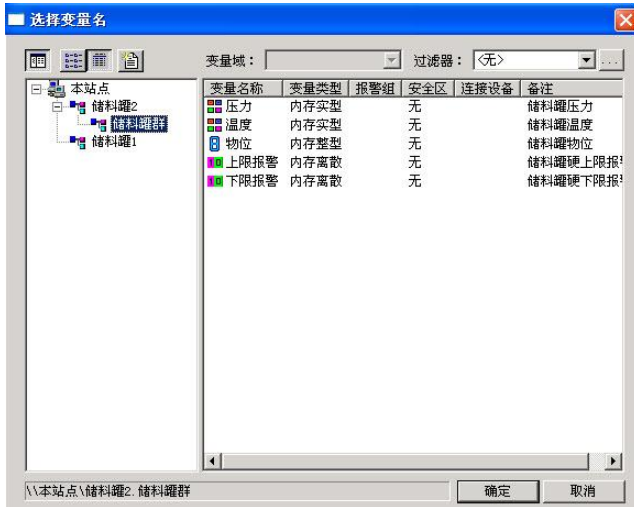


图 5.15 选择嵌套结构变量

对于嵌套的结构类型变量格式为：变量.结构模板.成员。如图 5.16 所示。

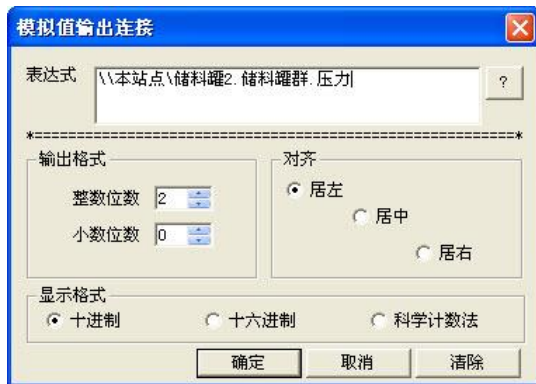


图 5.16 嵌套结构变量

5.4 变量的属性——变量域

变量的属性也是为满足工控软件的需求而引入的重要概念。它反映了变量的参数状态、报警状态,历史数据记录状态,比如实型变量“反应罐温度”,可以具有“高报警限”、“低报警限”等属性,当实际温度高于“高报警限”或低于“低报警限”时,就会在报警窗口内显示报警,而且它们大多是开放的,工程人员可在定义变量时,设置它的部分属性。也可以用命令语言编制程序来读取或设置变量的属性,比如在情况发生变化时,重新设置“反应罐温度”的“高、低报警限”。需要注意的是,有的属性可以被读取或设置,称为“可读可写”型;有的属性只能被读取不能被设置,称为“只读”型;有的属性只能被设置而不能读取,称为“只写”型。从而大大提高了组态的功能。

5.4.1 什么是变量的域

变量的属性用专门术语称为“变量的域”。对每个变量域的引用就是把变量名和域名用“.”号(西文输入状态下的句号)连接起来即可,类似于高级语言(C++)中的“结构”,比如变量“反应罐温度”的报警组名(Group)域,写成“反应罐温度.Group”。

5.4.2 变量有哪些域

变量的域越多,表明属性越丰富。不同类型的变量具有不同的域,详解如下:

5.4.2.1 变量的基本属性域

变量的基本属性域包括 Name, Comment, 质量戳相关域, 时间戳相关域。

- **Name:** 表示变量的名称, 字符型, 只读。
- **Comment:** 表示变量的描述内容, 字符型, 可读可写。

- **质量戳相关域：**变量的质量戳表示变量的数据质量好坏。质量戳相关域包括：

Quality：表示变量质量戳的值，整型，只读。

QualityString：表示变量质量戳字符串，字符串型，只读

- **时间戳相关域：**变量的时间戳表示变量数据的采集时间。时间戳相关域包括：

TimeYear：表示变量时间戳年的值，整型，只读

TimeMonth：表示变量时间戳月的值，整型，只读

TimeDate：表示变量时间戳日的值，整型，只读

TimeHour：表示变量时间戳小时的值，整型，只读

TimeMinture：表示变量时间戳分的值，整型，只读

TimeSecond：表示变量时间戳秒的值，整型，只读

TimeMsec：表示变量时间戳毫秒的值，整型，只读

TimeZone：表示变量时间戳时区的值，整型，只读

TimeDateString：表示变量时间戳日期的字符串，字符串型，只读

TimeTimeString：表示变量时间戳时间的字符串，字符串型，只读

组态王的变量（除报警窗和历史曲线变量外）均具有以上 14 种域。另外，I0 整型变量和 I0 实型变量还有下面四个域：

- **MaxEU：**最大值，模拟型，可读可写。
- **MinEU：**最小值，模拟型，可读可写。
- **MaxRAW：**表示变量的最大原始值，模拟型，可读可写。
- **MinRAW：**表示变量的最小原始值，模拟型，可读可写。

内存整型，内存实型变量也有 MaxEU 和 MinEU 域。



例如：

定义变量

基本属性 | 报警定义 | 记录和安全区

变量名: Test

变量类型: I/O整数

描述: This is a test

结构成员: 成员类型:

成员描述:

变化灵敏度: 0 初始值: 0

最小值: 0 最大值: 100

最小原始值: 0 最大原始值: 100

状态

保存参数

保存数值

图5.17 变量域值的定义

如上图 5.17 所示 test.Name 的值是“test”，test.Comment 的值是“This is a test”，test.MaxRaw 的值是 100，test.MinRaw 的值是 0。

5.4.2.2 变量的报警域

◇ 离散变量的报警域包括

- **Ack** 表示变量报警是否被应答，离散型，只读。
- **Alarm** 表示变量是否有报警，离散型，只读。
- **AlarmEnable** 表示变量的报警使能状态，离散型，可读可写
- **DataChanged** 表示变量的变化状态，当变量值变化时，该域置1，用户可以手动赋值为0，离散型，可读可写。
- **DataUpDate** 表示变量的状态变化，离散型，只读。默认状态为false。当从设备上采集上数据，填充实时库时（不管数据是否变化），该值置为true，需要手动复位。

- **Group** 表示变量所属的报警组ID，模拟型，可读可写。
- **Priority** 表示变量的报警优先级，模拟型，可读可写。
- **ExtendFields** 表示报警变量的扩展域1，字符串型，可读可写
tring1
- **ExtendFields** 表示报警变量的扩展域 2，字符串型，可读可写
tring2



注意：

变量的 Group 域（报警组 ID）的值只能通过命令语言来修改。

◇ 整型、实型变量除包括上面与离散变量相同的报警域外，还包括以下几种域：

- **HiHiLimit** 高高报警限，模拟型，可读可写。
- **HiHiStatus** 高高报警状态，离散型，只读。
- **HiLimit** 高报警限，模拟型，可读可写。
- **HiStatus** 高报警状态，离散型，只读。
- **LoLimit** 低报警限，模拟型，可读可写。
- **LoStatus** 低报警状态，离散型，只读。
- **LoLoLimit** 低低报警限，模拟型，可读可写。
- **LoLoStatus** 低低报警状态，离散型，只读。
- **F_ALARMDELAY** 越限或偏差报警延时，模拟值，可读可写。
TIME
- **MajorDevPct** 大偏差报警限，模拟型，可读可写。
- **MajorDevStat** 大偏差报警状态，离散型，只读。

us

- **MinorDevPct** 小偏差报警限，模拟型，可读可写。
- **MinorDevStat** 小偏差报警状态，离散型，只读。

us

- **DevTarget** 偏差报警限的目标值，模拟型，可读可写。
- **RocPct** 变化率报警限，模拟型，可读可写。
- **RocStatus** 变化率报警状态，离散型，只读。



例如：

在数据库中有一个 I/O 离散变量“断电保护”，如果要把它的报警优先级增加一级，则可用命令语句：

断电保护.Priority=断电保护.Priority-1 （数字越小，优先级越高）



例如：

在数据库中定义一个 I/O 实型变量“反应罐温度”，如果要显示“反应罐温度”的小偏差报警状态，则可用下述表达式：

反应罐温度.MinorDevStatus

5.4.2.3 变量的历史记录起停控制域

- **RecLogEnable:**

变量的历史记录起停控制域，表示变量的历史记录状态，可读写：0 表示该变量停止记录历史数据，1 表示该变量记录历史数据，默认为记录历史数据。字符串变量没有

RecLogEnable 域。



例如：

在数据库中定义一个I/O实型变量“反应罐温度”，如果要停止该变量的历史数据记录，可用命令语言设置：反应罐温度. RecLogEnable=0

5.4.2.4 报警窗口的域

和报警窗相关的域有：

- **Group:**

表示报警窗口显示的变量的报警组名，组变量，只写。

- **Priority:**

表示报警窗口显示的变量的报警优先级，模拟型，可读可写。



例如：

定义一个报警组名“第一车间”，为了让报警窗口变量“化工厂报警窗口”显示第一车间的报警，可用命令语言设置：

化工厂报警窗口. Group=第一车间



例如：

为了让报警窗口变量“化工厂报警窗口”显示报警优先级大于等于 10 的报警，可用命令语言设置：

化工厂报警窗口. Priority=10

5.4.2.5 历史趋势曲线的域

和历史趋势曲线的域包括：

ChartLength	历史趋势曲线的时间长度，长整型，可读可写，单位为秒。
ChartStart	历史趋势曲线的起始时间，长整型，可读可写，单位为秒。
ValueStart	历史趋势曲线的纵轴起始值，模拟型，可读可写。
ValueSIze	历史趋势曲线的纵轴量程，模拟型，可读可写。
ValueEnd	历史趋势曲线的纵轴结束值，模拟型，可读可写。
ScooterPosLeft	左指示器的位置，模拟型，可读可写。
ScooterPosRigh	右指示器的位置，模拟型，可读可写。
t	
Pen1 到 Pen8	历史趋势曲线显示的变量，变量 ID 号，可读可写，用于改变绘出曲线所用的变量。

历史趋势曲线的域的使用详见“第八章 趋势曲线和其它曲线”。

5.4.3 变量域的使用

在 5.4.2 节中已经介绍了一些变量的域的使用。变量的域可以用来在画面上显示，也可以在命令语言中使用。引用变量域时，可以直接手动输入，也可以通过变量浏览器来选择，如图 5.18 所示，在变量浏览器中选择相关变量，单击“变量域”列表框，弹出当前选择变量的所有域的列表，在列表框上挪动鼠标箭头，系统会自动显示一个提示文本，显示当前位置的变量域的数据类型。

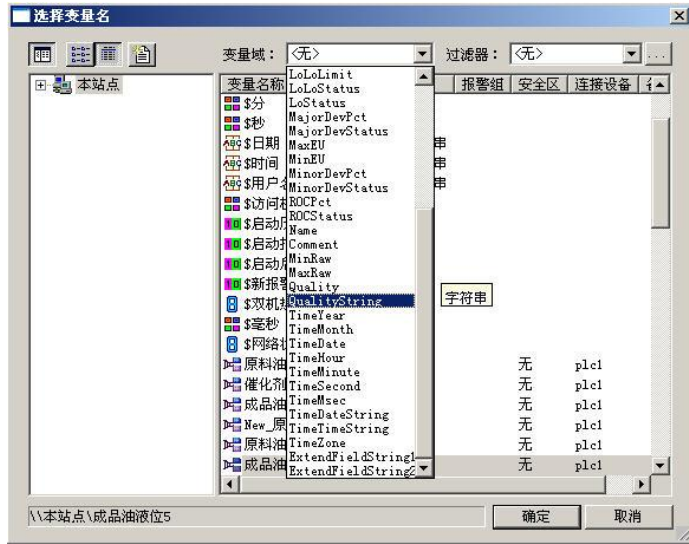


图5.18 选择变量域

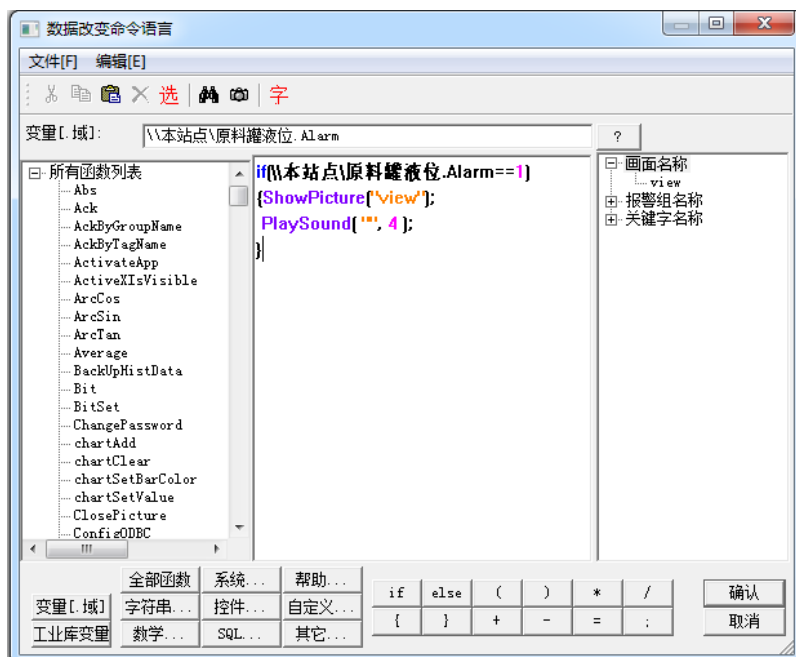


图 5.19 使用变量域

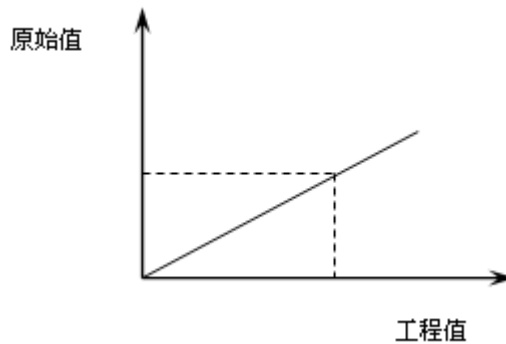
5.5 IO 变量的转换方式

对于 IO 变量——IO 模拟变量，在现场实际中，可能要根据输入要求的不同要将其按照不同的方式进行转换。比如一般的信号与工程值都是线性对应的，可以选择线性转换；有些需要进行累计计算，则选择累计转换。

组态王为用户提供了线性、开方、非线性表、直接累计、差值累计等多种转换方式。

5.5.1 线性转换方式

用原始值和数据库使用值的线性插值进行转换。如下图所示，线性转换是将设备中的值与工程值按照固定的比例系数进行转换。如图 5.20 所示，在变量基本属性定义对话框的“最大值”、“最小值”编辑框中输入变量工程值的范围，在“最大原始值”、“最小原始值”编辑框中输入设备中转换后的数字量值的范围(可以参考组态王驱动帮助中的介绍)，则系统运行时，按照指定的量程范围进行转换，得到当前实际的工程值。线性转换方式是最直接也是最简单的一种 IO 转换方式。



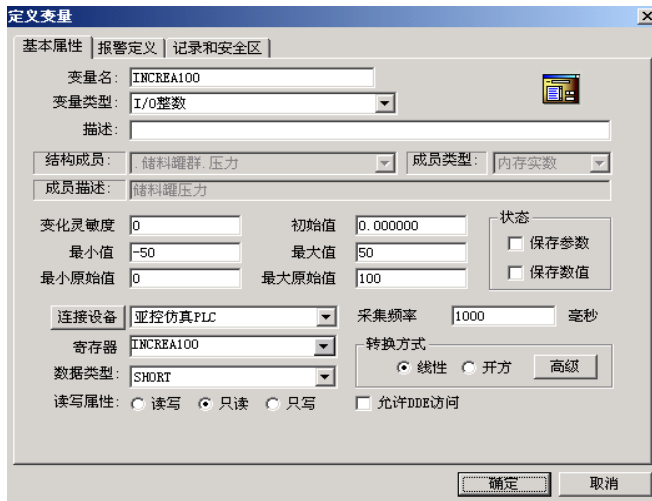


图 5.20 定义线性转换



例1:

与PLC电阻器连接的流量传感器在空流时产生0值，在满流时产生9999值。如果输入如下的数值：

最小原始值=0 最小值=0

最大原始值=9999 最大值=100

其转换比例= $(100-0)/(9999-0)=0.01$

则：如果原始值为 5000 时，内部使用的值为 $5000*0.01=50$ 。



例2:

与PLC电阻器连接的流量传感器在空流时产生6400值，在300GPM时产生32000值。应当输入下列数值：

最小原始值=6400

最小值=0

最大原始值=32000

最大值=300

其转换比例= $(300-0)/(32000-6400)=3/256$,

则：如果原始值为 19200 时，内部使用的值为 $(19200-6400)*3/256=150$ ；

原始值为 6400 时，内部使用的值为 0；原始值小于 6400 时，内部使用的值为 0。

5.5.2 开方转换方式

用原始值的平方根进行转换。即转换时将采集到的原始值进行开方运算，得到的值为实际工程值，该值的范围在变量基本属性定义的“最大值”、“最小值”范围内。如图 5.21 所示。

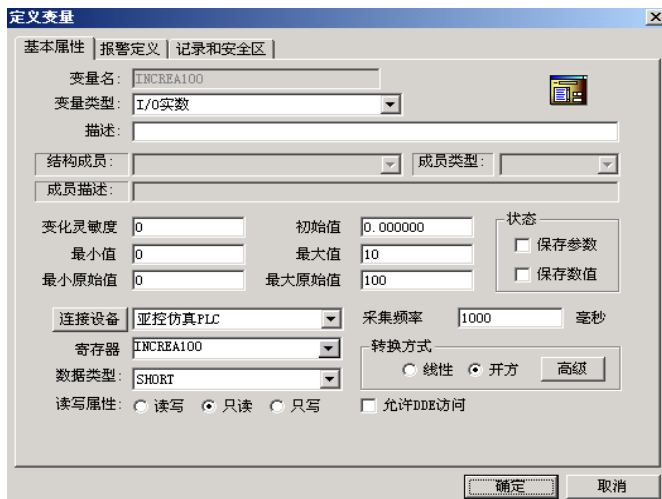
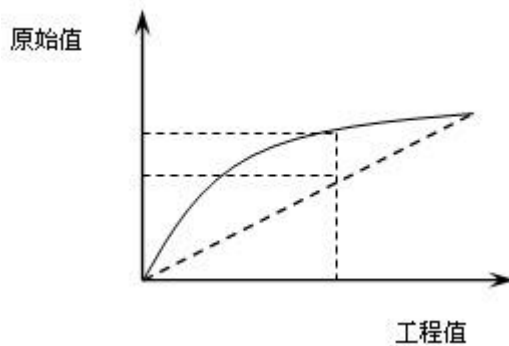


图 5.21 定义开方转换

5.5.3 非线性表转换方式

在实际应用中，采集到的信号与工程值不成线性比例关系，而是一个非线性的曲线关系。如果按照线性比例计算，则得到的工程值误差将会很大。如下图所示。对一些模拟量的采集，如热电阻、热电偶等的信号为非线性信号，如果采用一般的分段线性化的方法进行转换，不但要做大量的程序运算，而且还会存在很大的误差，达不到要求。



为了帮助用户得到更精确的数据，组态王中提供了非线性表。原始值和工程值可以是正比或反比的非线性关系，原始值和工程值可以是负数。

5.5.3.1 非线性表的定义

在组态王中引入了通用查表的方式，进行数据的非线性转换。用户可以输入数据转换标准表，组态王将采集到的数据的设备原始值和变量原始值进行了线性对应后（此处“设备原始值”是指从设备采集到的原始数据；“变量原始值”是指经过组态王的最大、最小值和最大、最小原始值转换后的值，包括开方和线性，“变量原始值”以下通称“原始值”），将通过查表得到工程值，在组态王运行系统中显示工程值或利用工程值建立动画

连接。非线性表是用户先定义好的原始值和工程值一一对应的表格，当转换后的原始值在非线性表中找不到对应的项时，将按照指定的公式进行计算，公式将在后面介绍。非线性查表转换的定义分为两个步骤：

- ◇ 变量将按照变量定义画面中的最大值、最小值、最大原始值和最小原始值进行线性转换，即将从设备采集到的原始数据经过与组态王的初步转换。其具体含义见本节举例“I/O 实型变量的转换方式和转换比例”。
- ◇ 将上述转换的结果按照非线性表进行查表转换，得到变量的工程值，用于在运行时显示、存储数据、进行动画连接等。关于非线性查表转换方式的具体使用如下：
 - **建立非线性表：**在工程浏览器的目录显示区中，选中大纲项“文件”下的成员“非线性表”，双击“新建……”图标，弹出“分段线性化定义”对话框，如图 5.22 所示。



图 5.22 分段线性化定义对话框

表格共三列，第一列为序号，增加点时系统自动生成。第二列是原始值，该值是指从设备采集到的原始数据经过与组态王变量定义界面上的最小值、最大值、最小原始值、最大原始值转换后的值。第三列为该原始值应该对应的工程值。

- **非线性表名称：**在此编辑框内输入非线性表名称，非线性表名称唯一，表名可以为数字或字符。
- **增加点：**增加原始值与工程值对应的关系点数。单击该按钮后，在“线性化分段定义”显示框中将增加一行，序号自动增加，值为空白或上一行的值。用户根据数据对应关系，在表格框中写入值，即对应关系。例如，对

于非线性表“线性转换表”，用户建立 10 组对应关系，如图 5.23 所示。



图 5.23 定义非线性表

- **删除点：**删除表格中不需要的线性对应关系。选中表格中需要删除行中的任意一格，单击该按钮就可删除。
- **对变量进行线性转换定义：**在数据词典中选择需要查表转换的 I/O 变量，双击该变量名称后，弹出“变量属性”对话框。在“变量定义”界面上，点击“转换方式”下的“高级”按钮，弹出“数据转换”对话框，如图 5.24 所示。默认选项为“无”。当用户需要对采集的数据进行线性转换时，请选中“查表”一项。其右边的下拉列表框和“+”按钮变为有效。

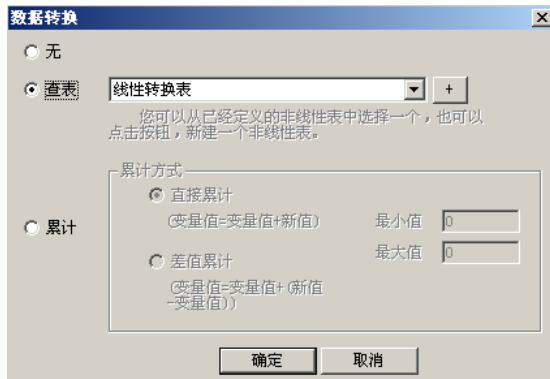


图 5.24 变量定义数据转换对话框

单击下拉列表框右边的箭头，系统会自动列出已经建好的所有非线性表，从中选取即可。如果还未建立合适的非线性表，可以单击“+”按钮，弹出“分段线性化定义”对话框，如图 5.24 所示，用户根据需要建立非线性表，使用方法见（1）。

运行时，变量的显示和建立动画连接都将是查表转换后的工程值。查非线性表的计算公式为：

$$\left((\text{后工程值} - \text{前工程值}) * (\text{当前原始值} - \text{前原始值}) / (\text{后原始值} - \text{前原始值}) \right) + \text{前工程值}$$

当前原始值：当前变量的变量原始值。

后工程值：当前原始值在表格中原始值项所处的位置的后一项数值对应关系中的工程值。

前工程值：当前原始值在表格中原始值项所处的位置的前一项数值对应关系中

的工程值。

后原始值：当前原始值在表格中原始值项所处的位置的下一原始值。

前原始值：当前原始值在表格中原始值项所处的位置的前一原始值。



例如：

在建立的非线性列表中，数据对应关系为：



序号	原始值	工程值
1	4	8
2	6	14

那么当原始值为 5 时，其工程值的计算为：

工程值= ((14-8) * (5-4) / (6-4)) +8，即为 11。

在画面中显示的该变量值为 11。

5.5.3.2 非线性表的导入、导出

当非线性表比较庞大，分段比较多时，在组态王中直接进行定义就显得很困难。为此，组态王为用户提供了非线性表的导入、导出功能，可以将非线性表导出为.csv 格式的文件；也可将用户编辑的符合格式要求的.csv 格式的文件导入到当前的非线性表中来。这样方便了用户的操作。

打开已经定义的非线性表，单击“导出……”按钮，弹出“保存为”对话框，选择保存路径及保存名称，单击“保存”按钮，可以将非线性表的内容保存到文件中。如下图 5.25 所示。



图 5.25 导出非线性表

导出后的文件内容如下图 5.26 所示。



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - 线性转换表". The menu bar includes "文件(F)", "编辑(E)", "视图(V)", "插入(I)", "格式(O)", "工具(T)", "数据(D)", "窗口(W)", and "帮助(H)". The spreadsheet contains the following data:

	A	B	C	D	E
1	序号	原始值	工程值		
2	1	-30	-40		
3	2	-10	-5		
4	3	0	18		
5	4	20	40		
6	5	40	52		
7	6	60	65		
8	7	80	78		
9	8	100	93		
10	9	120	105		
11	10	140	120		
12					
13					
14					

图 5.26 导出的非线性表内容

用户也可以按照图 5.26 所示的文件格式制作非线性表，然后导入到工程中来。

对于非线性表的导入有两个途径：从其它工程导入和从.csv 格式的文件导入，如图 5.27 所示，单击“分段线性化定义”对话框上的“导入”按钮，弹出“导入非线性表”对话框，该对话框分为两个部分，上部分为当前工程管理器中的工程列表，选择非线性表所在的工程，在“非线性表”的列表框中会列出该工程中含有的非线性表名称。选择所需的表名称，单击“导入”按钮，可以将非线性表导入到当前工程里来。

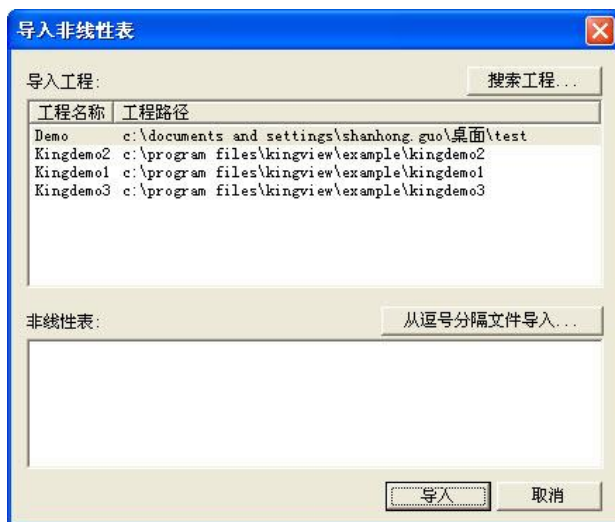


图 5.27 导入非线性表

另外也可以选择文件导入。单击“从逗号分隔文件导入”按钮，弹出文件选择对话框，选择要导入的文件即可。

总之，非线性表的导入、导出功能方便了用户对非线性表的重复利用和快速编辑。提高了工作效率。

5.5.4 累计转换方式

累计是在工程中经常用到的一种工作方式，经常用在流量、电量等计算方面。组态王的变量可以定义为自动进行数据的累计。组态王提供两种累计算法：直接累计和差值累计。累计计算时间与变量采集频率相同，对于两种累计方式均需定义累计后的值的最大最小值范围，如图 5.28 所示。

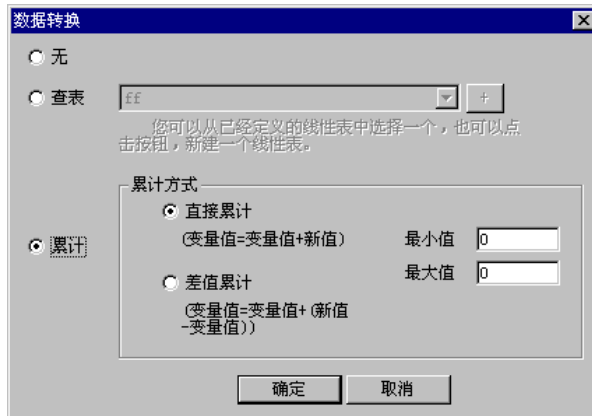


图 5.28 数据转换的累计功能定义对话框

当累计后的变量的数值超过最大值时，变量的数值将恢复为该对话框中定义的最小值。

◇ **直接累计：**

从设备采集的数值，经过线性转换后直接与该变量的原数值相加。计算公式为：

变量值=变量值+采集的数值



例如：

管道流量 S 计算，采集频率为 1000ms，5 秒钟之内采集的数据经过线性转换后工程值依次为 S1=100、S2=200、S3=100、S4=50、S5=200，那么 5 秒钟内直接累计流量结果为：

$$S=S1+S2+S3+S4+S5, \text{ 即为 } 650。$$

◇ **差值累计：**

变量在每次进行累计时，将变量实际采集到的数值与上次采集的数值求差值，对其

差值进行累计计算。当本次采集的数值小于上次数值时，即差值为负时，将通过变量定义的画面中的最大值和最小值进行转化。

差值累计计算公式为：

显示值=显示旧值+（采集新值-采集旧值）（公式一）

当变量新值小于变量旧值时，公式为：

显示值=显示旧值+（采集新值-采集旧值）+（变量最大值-变量最小值）（公式二）

变量最大值是在变量属性定义画面最大最小值中定义的变量最大值。



例如：

条件如上例，变量定义画面中定义的变量初始值为 0，最大值为 300。那么 5 秒钟之内的差值累计流量计算为：

$$\text{第 1 次: } S(1)=S(0)+ (100-0)=100$$

（采用公式一）

$$\text{第 2 次: } S(2)=S(1)+ (200-100)=200$$

（采用公式一）

$$\text{第 3 次: } S(3)=S(2)+ (100-200)+(300-0)=400 \quad \text{（采用公式二）}$$

$$\text{第 4 次: } S(4)=S(3)+ (50-100)+(300-0)=650 \quad \text{（采用公式二）}$$

$$\text{第 5 次: } S(5)=S(4)+ (200-50)=800 \quad \text{（采用公式一）}$$

即 5 秒钟之内的差值累计流量为 800。

5.6 变量管理工具一 —— 管理变量

当工程中拥有大量的变量时，会给开发者查找变量带来一定的困难，为此组态王提供了变量分组管理的方式。即按照开发者的意图将变量放到不同的组中，这样在修改和选择变量时，只需到相应的分组中去寻找即可，缩小了查找范围，节省了时间。并且它对变量的整体使用没有任何影响。

5.6.1 如何建立变量组

在组态王工程浏览器框架窗口上放置有四个标签：“系统”、“变量”、“站点”和“画面”。选择“变量”标签，左侧视窗中显示“变量组”。单击“变量组”，右侧视窗将显示工程中所有变量，如下图 5.29 所示

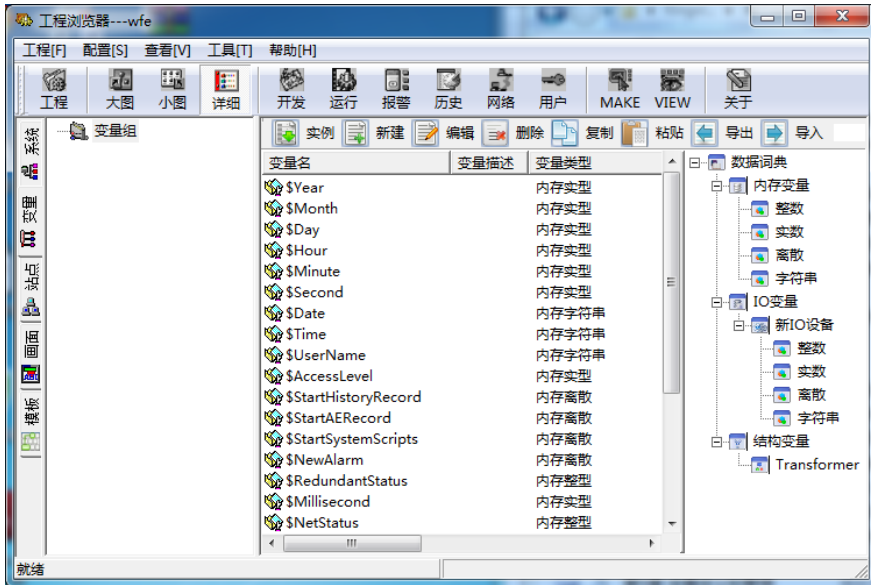


图 5.29 变量组

在“变量组”目录上单击鼠标右键，弹出快捷菜单，选择“建立变量组”。如图 5.30 所示。

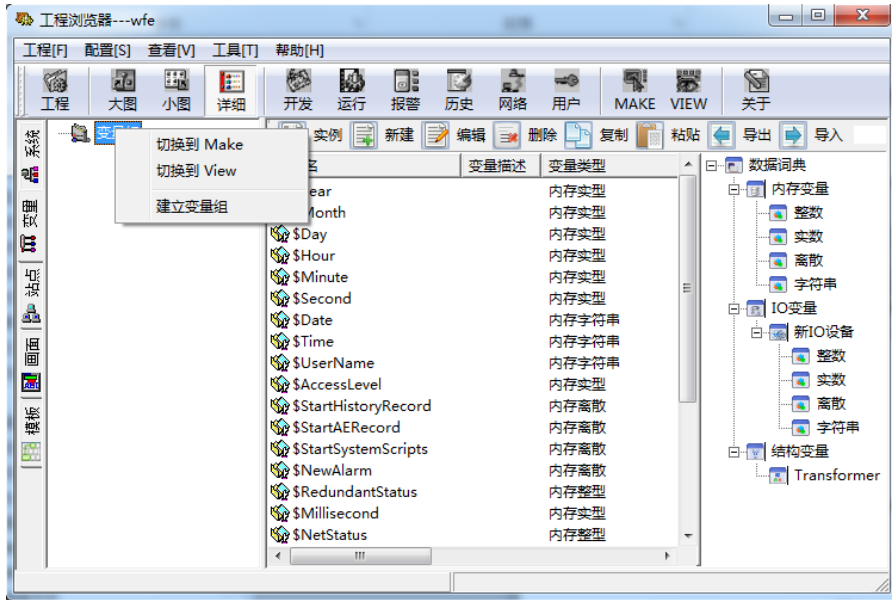


图 5.30 建立变量组

在编辑框中输入变量组的名称，如下图 5.31 所示。

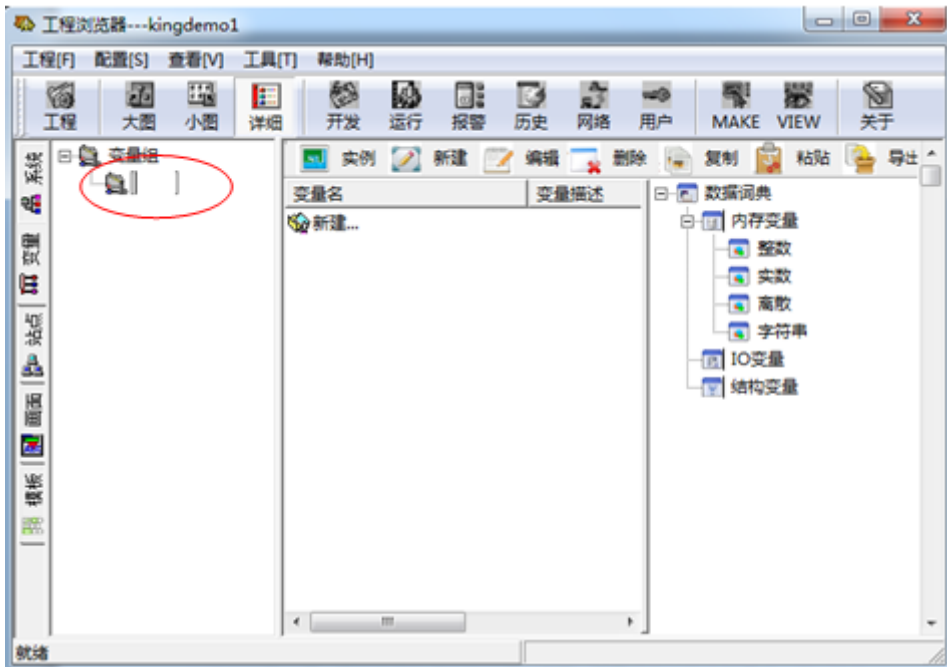


图 5.31 命名变量组

如果按照默认项，系统自动生成名称并添加序号。变量组定义的名称是唯一的，而且要符合组态王变量命名规则。如下图 5.32 所示。

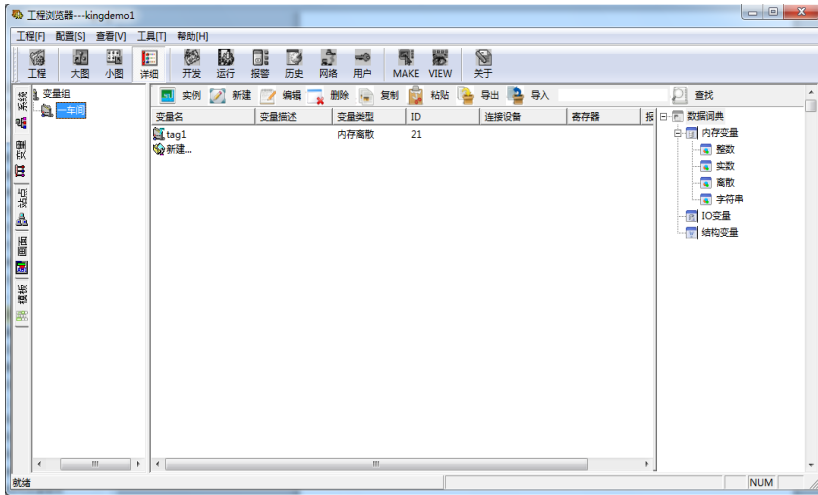


图 5.32 建立完成的变量组

变量组建立完成后，可以在变量组下直接新建变量，在该变量组下建立的变量属于该变量组。变量组中建立的变量可以在系统中的变量词典中全部看到。在变量组下，还可以再建立子变量组，如图 5.33 所示。属于子变量组的变量同样属于上级变量组。

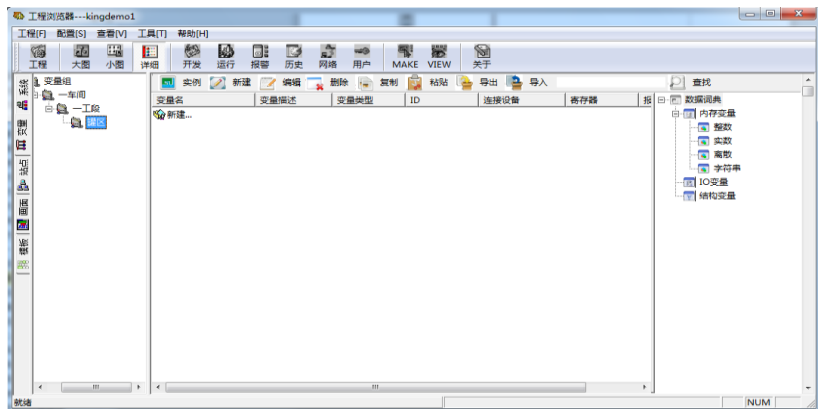



图 5.33 建立子变量组

选择建立的变量组，单击鼠标右键，在弹出的快捷菜单中选择“编辑变量组”，可以修改变量组的名称。

 注意：

根变量组名称“变量组”是不允许修改和删除的。

5.6.2 如何在变量组中增加变量

变量组建立完成后，就可以在里面增加变量了。增加变量可以直接新建，如图 5.33 所示，双击“新建……”图标直接新建变量。也可以从已定义的变量，包括其它变量组中移动到当前变量组来。下面主要介绍后者。

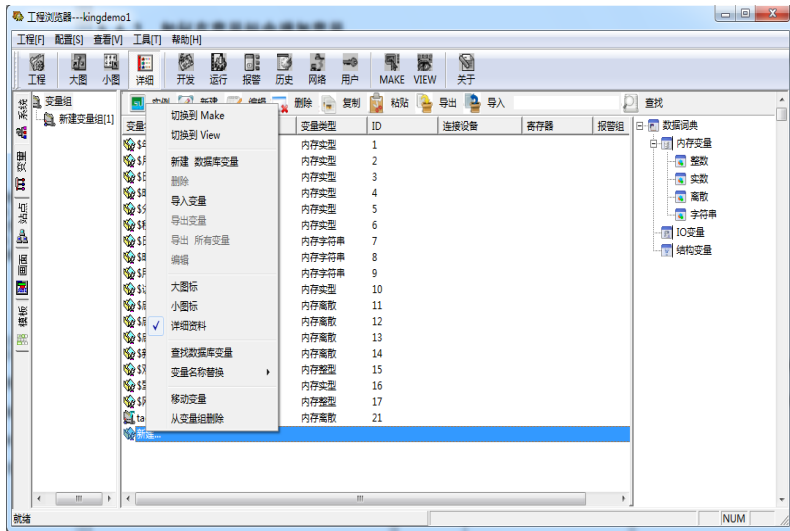


图 5.34 选择要移动的变量

如图 5.34 所示，在某个变量组中选择要移动的变量，单击鼠标右键，在弹出的快捷

菜单中选择“移动变量”，然后选择目标变量组，在右侧的内容区域中，单击鼠标右键，在弹出的快捷菜单中选择“放入变量组”，则被选择的变量就被移动到目标变量组中。在系统变量词典中，属于变量组的变量图标与其它图标不相同。

在变量分组完成后，使用时，只需在变量浏览器中选择相应的变量组目录即可，如图 5.35 所示。变量的引用不受变量组的影响，所以变量可以被放置到任何一个变量组下。

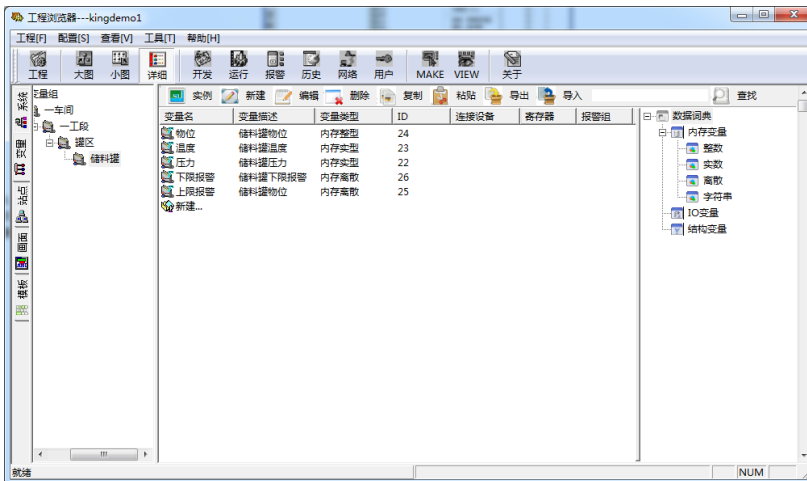


图 5.35 通过变量组选择变量

5.6.3 变量组内变量排序

在某变量组内的变量可以按不同方式进行排序显示，除不能使用按“变量描述”进行排序显示以外，可以按“变量名称”、“变量类型”、“ID”、“连接设备”、“寄存器”、“报警组”进行排序显示。

5.6.4 如何在变量组中删除变量

如果不需要在变量组中保留某个变量时，可以选择从变量组中删除该变量，也可以选择将该变量移动到其它变量组中。从变量组中删除的变量将不属于任何一个变量组，但变量仍然存在于数据词典中。

进入该变量组目录，选中该变量，单击鼠标右键，在弹出的快捷菜单中选择“从变量组删除”，如图 5.36 所示，则该变量将从当前变量组中消失。如果选择“移动变量”，可以将该变量移动到其它变量组。

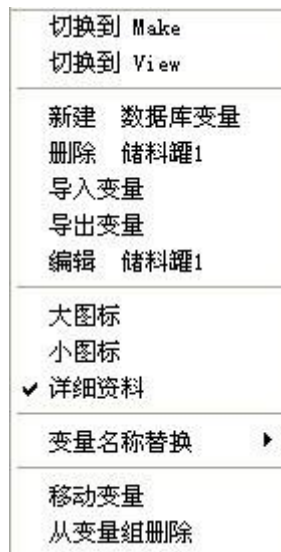


图 5.36 移动变量

5.6.5 如何删除变量组

当不再需要变量组时，可以将其删除，删除变量组前，首先要保证变量组下没有任何变量存在，另外也要先将子变量组删除。

在要删除的变量组上单击鼠标右键，然后在快捷菜单上选择“删除变量组”，系统提示删除确认信息，如果确认，当前变量组将被永久删除。

5.7 变量管理工具二——导入导出、更新替换、获取变量使用情况

组态王中提供了很多变量管理和使用的工具和方法等。如数据词典导入、导出；变量的更新、替换；获得变量的使用情况等。

5.7.1 数据词典导入导出到 Excel 中

5.7.1.1 数据词典导出到 Excel

为了使用户更方便地使用、查看、定义或打印组态王的变量，组态王提供了数据词典的导入导出功能。组态王的变量被导出到 Excel 格式的文件中，用户可以在 Excel 文件中查看、修改变量的一些属性，或直接在該文件中新建变量并定义其属性，然后导入到工程中。数据词典导出功能在工程管理器中。

打开工程管理器，关闭组态王开发和运行系统，在工程管理器的工程列表中选择要导出数据词典的工程。点击工程管理器工具条上的“DB 导出”按钮，或选择菜单“工具\数

据词典导出”命令。执行该命令后，系统弹出文件选择对话框，如图 5.37 所示：



图 5.37 输入数据词典导出的文件名

选择保存导出的数据词典文件的路径，并输入保存的文件名称，单击“保存”，工程管理器的状态栏上会出现当前进程的提示和进度条显示。导出后的文件如图 5.38 所示。



图 5.38 数据词典导出的 Excel 文件

数据词典的导出规则：

数据词典导出后的 Excel 文件共有四页：说明页、模板页、结构变量页和基本变量页。

◇ 说明页：

为数据词典导入、导出的使用说明。用户在导入、导出数据词典时，应参照该说明进行。该页的内容不可修改。

◇ 模板页：

模板页为工程中定义的所有结构模板的信息，其具体格式举例如下：

[模板]	模板 ID	模板名称	模板使用记数	注释
	1	离散结构	1	
		[成员]	成员 ID	成员名称 成员类型
			1	离散成员 1 离散型
			2	离散成员 2 离散型
			
	模板 ID	模板名称	模板使用记数	注释
	2	整型结构	1	
		[成员]	成员 ID	成员名称 成员类型
			1	整型成员 1 整型
			2	整型成员 2 整型
			

- **模板：**模板 ID 为在结构模板中定义的模板的序号；
- **模板名称：**为结构模板中定义的模板的名称；

- **模板记数:** 为在定义结构变量时应用该模板的次数。
 - **成员:** 成员 ID 是在结构模板中定义的成员的序号;
 - **成员名称:** 为结构模板中定义的该结构的成员的名称;
 - **成员类型:** 为结构模板中定义的该结构成员的数据类型。
-

 **注意:**

用户在新建结构模板时, 必须严格按照以上格式建立, 否则在导入数据词典时, 会产生错误, 系统将拒绝导入。结构模板嵌套最多为两级。

◇ **结构变量页:**

为用户定义的结构变量的信息。如用户定义了一个结构变量名称为: 离散结构变量, 则该页中的信息为:

[结构变量] 变量 ID	变量名称	变量类型	变量使用记数	注释
1	离散结构变量	离散结构	0	
[成员]	成员名称	成员基本变量 ID		
	离散结构变量. 离散成	41		离散结构变
	员 1			量 1
	离散结构变量. 离散成	42		离散结构变
	员 2			量 2
	离散结构变量. 成员 3	43		离散结构变
				量 3

- **变量 ID:** 该结构变量在定义的结构变量列表中的序号。

- **变量名称：**结构变量名称，即定义基本变量为结构变量类型时基本变量的变量名称。
 - **变量类型：**定义基本变量为结构变量类型时选择的结构模板名称。
 - **变量使用计数：**该结构变量类型的基本变量在组态王中被引用的次数。
 - **成员名称：**该结构变量类型的基本变量的每个成员在组态王中的名称。
 - **成员基本变量 ID：**结构变量类型的基本变量的每个成员在基本变量中的 ID 号。
 - **注释：**对于该结构变量的注释和基本变量的每个成员的注释。
-



注意：

用户在新建结构变量时，必须严格按照以上格式建立，否则在导入数据词典时，会产生错误，系统将拒绝导入。

变量的 ID 号不能重复，而且基于同一个结构模板的结构变量的 ID 号是连续的。

- ◇ **基本变量页：**将组态王的基本变量按照变量类型的不同分别列出。导出的每个变量的内容为变量名称、ID 号；变量是否记录数据、是否记录参数等选项的情况；变量各个域的值及其描述文本等。

建立基本变量时各有效属性值必须符合以下规则：

- **模板成员类型有效值：**内存/I/O 整型、内存/I/O 实型、内存/I/O 离散型、内存/I/O 字符串型、报警窗口、历史趋势曲线、其它模板类型名称。
- **变量名（名称）：**由字母、数字、汉字组成，但不能以数字开头。有效长度小于 32。

- **是否保存值：**有效值为是或否。
- **是否保存参数：**有效值为是或否。
- **报警组：**组态王中定义的报警组的名称。
- **优先级：**1-999。
- **报警文本**(低低、低、高、高高、关、开、关到开、开到关)：文本有效长度小于 16。
- **变化率单位：**有效值为时、分、秒。
- **扩展域 1、扩展域 2：**文本有效长度小于 16。
- **历史记录方式：**有效值为不记录、变化记录、定时记录。
- **是否生成操作事件：**有效值为是或否。
- **安全区：**组态王中用户配置中定义的关于安全区的名称（选其一）。
- **注释：**文本有效长度小于 40。
- **是否产生报警：**有效值为是或否。
- **报警类型：**有效值为开、关或变位。
- **设备名称：**为组态王中定义的设备名称，有效长度小于 32。
- **读写属性：**有效值为读写、只读或只写。
- **是否允许 DDE 访问：**有效值为是或否。
- **转换方式：**有效值为线性或开方。
- **高级转换方式：**查表、直接累计或差值累计。
- **线性表名称：**查表方式下，线性表的名称。

- **最小累计值：**使用累计转换方式时，定义累计值的最小值。
- **最大累计值：**使用累计转换方式时，定义累计值的最大值。

.....

其余各项与组态王变量定义相同。

5.7.1.2 从 Excel 中导入数据词典

数据词典的导入是将 Excel 中定义好的数据或将由组态王工程导出的数据词典导入到组态王工程中。打开工程管理器，关闭组态王开发和运行系统，在工程管理器的工程列表中选择要导入数据词典的工程。点击工程管理器工具条上的“DB 导入”按钮，或选择菜单“工具\数据词典导入”命令。执行该命令，首先弹出“导入数据词典”提示信息框，如下图 5.39 所示，提示用户在导入数据词典之前是否备份工程。

 注意：

在这里，为了防止用户工程在导入过程中出现错误，建议用户对工程进行备份。

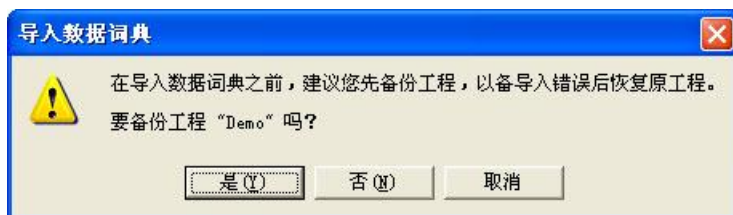


图 5.39 导入数据词典提示信息框

- ◇ 单击“是”按钮，进行工程备份
- ◇ 单击“取消”按钮，取消导入数据词典操作

◇ 单击“否”按钮，进行数据词典的导入。弹出文件选择对话框，如图 5.40 所示。



图 5.40 选择导入数据词典文件

数据词典既可以导入到原工程中，也可以导入到其它工程中。在导入数据词典时，系统自动根据 Excel 文件和被导入工程的数据词典进行比较，在比较完成后，系统弹出“变量导入校验报告”，如图 5.41 所示。

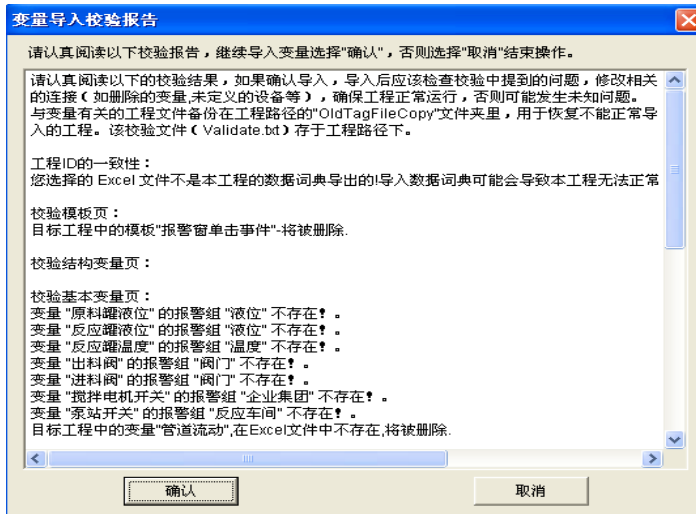


图 5.41 变量导入校验报告

在该校验报告中,显示了 Excel 文件中所有与当前被导入工程在数据词典中不同的地方,包括变量 ID、报警组名称,定义的设备、寄存器等。用户要认真核对这些校验结果,如果确定导入,点击校验报告对话框上的“确定”按钮,进行导入,否则点击“取消”。系统会自动备份当前工程的数据词典数据文件。

如果导入后出现错误的话,可以从工程目录下的“OldTagFileCopy”目录下将当前工程的原数据词典文件恢复到工程目录下。

在导入完成后,用户应该按照校验报告提示的信息对变量以及工程中关联变量的地方进行修改,然后再启动组态王运行系统,否则可能产生无法预料的异常情况出现。

注意:

请认真阅读校验结果,如果确认导入,导入后应该检查校验中提到的问题,修改相关的连接(如删除的变量,未定义的设备等),确保工程正常运行,否则可能发生未知问题。

与变量有关的工程文件备份在工程路径的“OldTagFileCopy”文件夹里,用于恢复不能正常导入的工程。该校验文件(Validate.txt)存于工程路径下。

5.7.2 数据词典导入导出到 Access 中

5.7.2.1 数据词典导出到 Access

打开工程浏览器,选中要导出的变量,如图 5.42 所示。

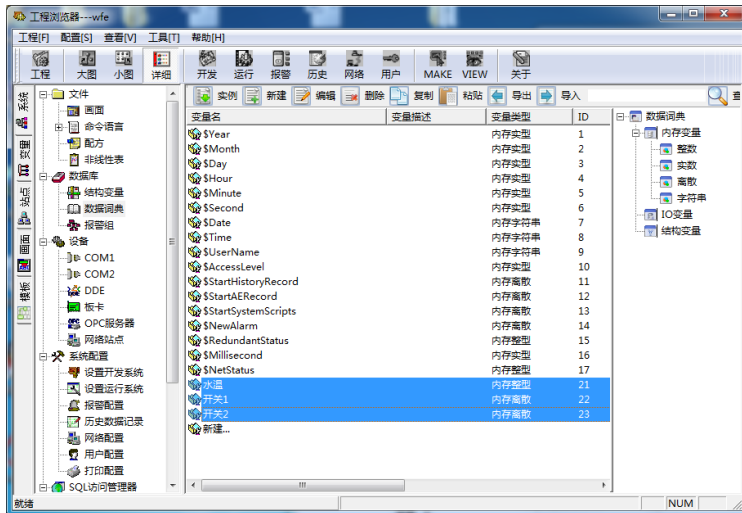


图 5.42 选中要导出的变量

然后单击右键，弹出窗口，选中“导出变量”项，如下图 5.43 所示。



图 5.43 导出变量

执行该命令后，系统弹出文件选择对话框，如下图 5.44 所示。

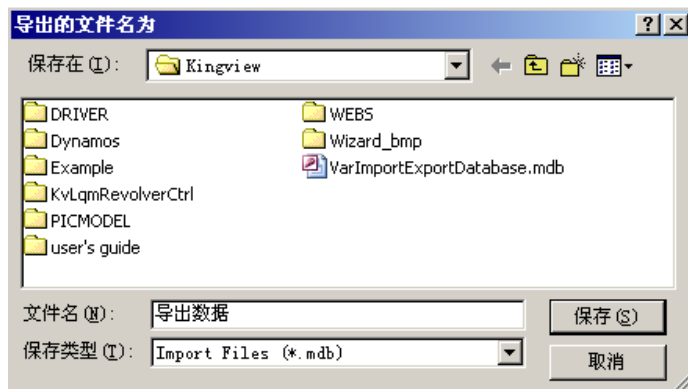


图 5.44 为导出的文件命名

选择保存导出的数据词典文件的路径，并输入保存的文件名称，如上图 5.44 所示，将导出的文件放在 kingview 文件夹下，并命名为：导出数据，单击“保存”，弹出导出变量信息的进度条，如下图 5.45 所示。



图 5.45 变量导出进度显示

在变量导出完毕后，弹出提示信息，如下图 5.46 所示。

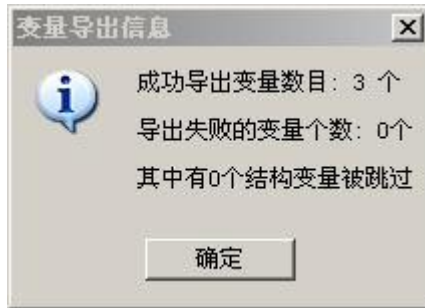


图 5.46 成功导出变量

这时，在指定的路径下面生成一个导出数据.mdb 文件。

打开生成的数据库表格，在表项中，显示可导出的变量类型，如下图 5.47 所示。

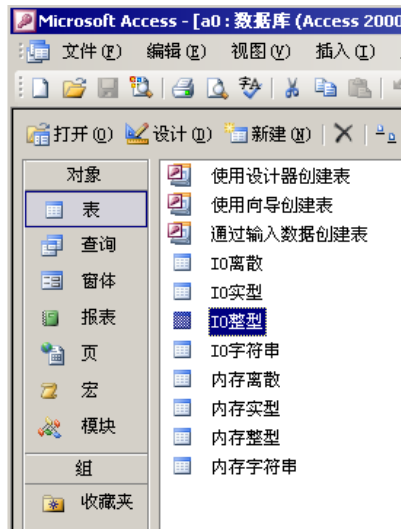


图 5.47 数据词典导出的 Access 文件

选中一个数据类型，打开设计，表格中的字段名对应组态王数据的各项名称，例如：

打开 IO 整型的设计页，如下图 5.48 所示。

字段名称	数据类型	
VarName	文本	变量名字
ChangeSensitivity	数字	变化灵敏度
InitialValue	数字	初始值
MinValue	数字	最小值
MaxValue	数字	最大值
MinOriginalValue	数字	最小原始值
MaxOriginalValue	数字	最大原始值
SaveData	是/否	是否保存参数
SaveParameter	是/否	是否保存参数
DeviceName	文本	设备名称
DeviceType	文本	设备类型
RegisterName	文本	寄存器名称
DataType	文本	数据类型
RWAttribute	文本	读写属性
CollectFrequency	数字	采集频率
CovertWay	文本	转换方式
AdvanceCovertWay	文本	高级转换方式
LinearTableName	文本	线性表名称
MinAccumulateValue	数字	最小累计值
MaxAccumulateValue	数字	最大累计值
DDEAccess	是/否	是否允许DDE访问
AlarmGroup	文本	报警组
Priority	数字	优先级
LoLoLimit	数字	低低限值
LoLoAlarmText	文本	低低报警文本

图 5.48 数据库字段名与组态王导出项的对应关系

组态王同时支持以变量组为单位的变量的导出。组态王工程浏览器窗口中的“变量”标签下，选中要导出变量的根变量组或变量组，在内容显示区单击鼠标右键，弹出如图所示窗口。

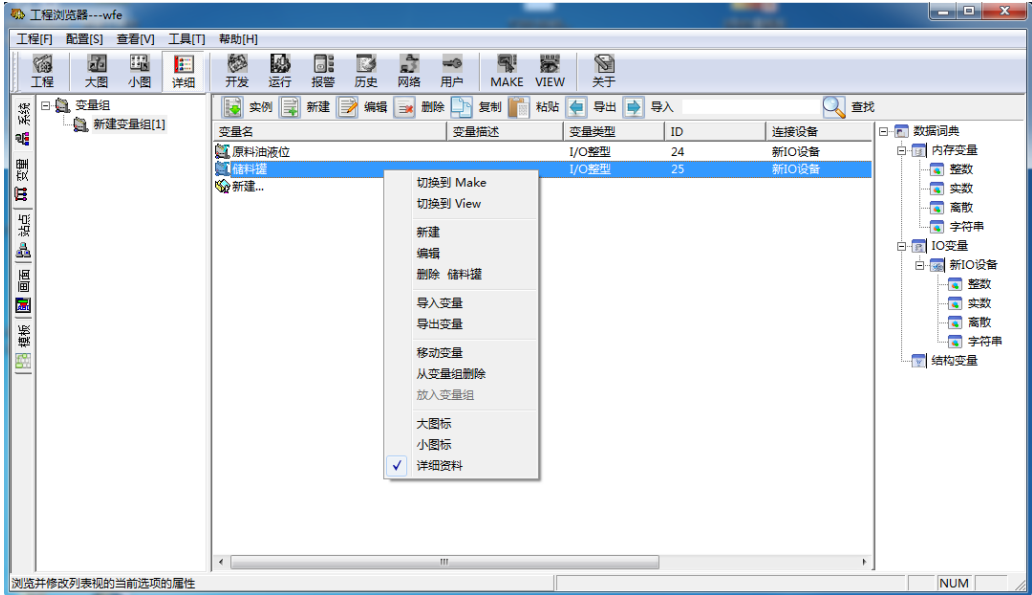


图 5.49 变量组中导出变量

具体操作请参考从数据词典中的导出。

5.7.2.2 从 Access 中导入数据词典

数据词典的导入是将由组态王工程导出的数据词典导入到组态王工程中。数据词典既可以导入到原工程中，也可以导入到其它工程中。下面就这两种方式详细说明。

5.7.3 数据词典导入导出到工程

5.7.3.1 将数据词典导入到原工程

◇ 变量导入方式 1：变量为单位的导入

打开工程管理器的数据词典，在变量区单击右键，弹出菜单，选中“导入变量”项，如图 5.50 所示。



图 5.50 变量导入

执行该命令后，系统弹出打开对话框，如下图 5.51 所示。

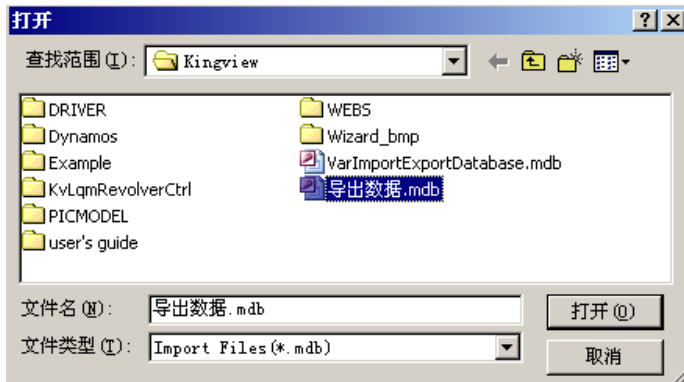


图 5.51 选择导入的数据库

选择要导入的数据库的文件，如上图所示，导入的数据库名称为“导出数据.mdb”。

单击“打开”，显示导入变量列表，如下图 5.52 所示。

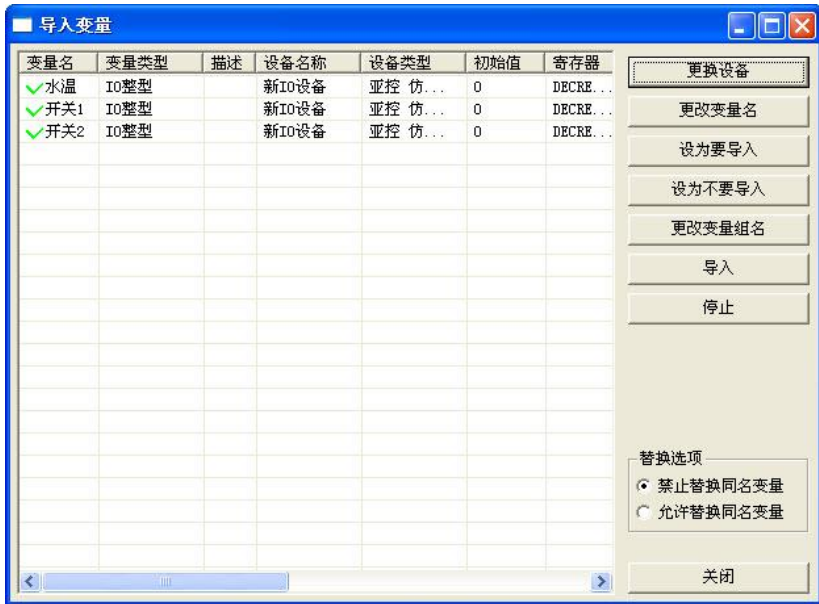


图 5.52 导入变量列表

通过该表格，可以对导入的变量进行变量名修改，变量关联设备的修改。

- **更换设备：**将导入变量关联的设备更换为工程中定义的设备。
- **修改变量名：**将导入的变量名进行修改。可单个修改变量名，也可以批量修改。

1) 单个修改变量名：选中一个变量，点击“修改变量名”，弹出如下对话框：



输入要修改的名称，确定后变量被更名。变量更名要符合变量的命名规则。

- 2) 批量修改变量名称：按住“shift”键，连续选择多个变量后，点击“修改变量名”，弹出如下对话框：



添加前：在原来变量名称前面添加字符，变量更名原则要符合变量的命名规则。

添加后：在原来变量名称后面添加字符，变量更名原则要符合变量的命名规则。

- **设为可导入：**将选中的变量设置为导入工程。
- **设为不可导入：**将选中的变量设置为不导入工程，并在变量名称前显示“**×**”。
- **修改变量组名：**选中变量，单击“修改变量组名”，弹出“修改变量组名”对话框，如图 5.53 所示。输入变量组名。在根变量组或数据字典中导入变量时，可以指定变量的所在组名，变量组信息为空的变量直接导入到根变量组中。



图 5.53 修改变量组名

- **导入：**将列表中的设置为可导入的变量导入到工程中。

- **替换选项**

禁止替换同名变量：导入变量与工程中的变量同名时，将不进行导入同名变量。

允许替换同名变量：导入的变量与工程中的变量同名时，用导入变量替换工程中的同名变量。

- **停止：**在导入过程中可以随时停止变量的导入操作。
- **关闭：**将导入变量对话框关闭。



例如：

将 5.7.2.1 章节中导出的变量，再导入到本工程中，图 5.59 中所示的变量“开关 1”更名为“开关 11”，将变量“开关 2”设置为不可导入，变量“水温”与工程中的变量名同名；选择“禁止替换同名变量”，如下图 5.54 所示。



图 5.54 修改导入变量列表

单击“导入”，弹出如下所示对话框：



图 5.55 导入变量完成

单击“确定”按钮，将“导入变量”对话框关闭，打开工程的数据词典，如下图 5.56 所示。

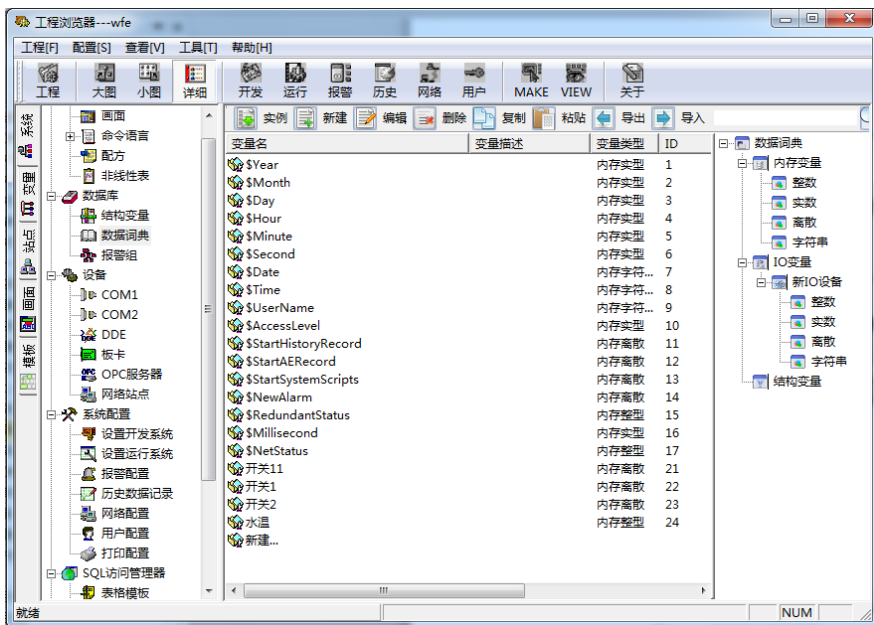
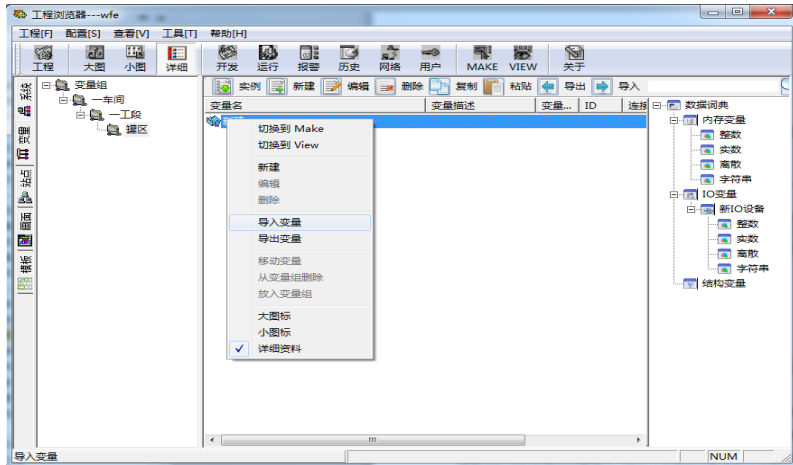


图 5.56 导入变量后的数据词典

◇ 变量导入方式 2：变量组为单位的变量导入

组态王同时支持以变量组为单位的变量的导入。组态王工程浏览器窗口中的“变量”标签下，选中要导入变量的根变量组或变量组，在内容显示区单击鼠标右键，弹出窗口，如下图 5.57 所示。



5.57 变量组导入变量

具体操作请参考从数据词典中的导入。在以变量组为单位的变量的导入中，导入变量之前可根据实际需要配置变量组名称，系统首先导入各变量，再根据各变量所作的配置自动移动到相应的变量组中，对于不存在的变量组自动创建后将变量移入。配置变量组信息有两种方法：

- 第一种方式是在 Access 数据库中直接输入。

打开导出的 Access 数据库文件，打开各类型变量对应的表格，通过配置对应表格中的“GroupPath”字段内容来配置变量组信息。



图 5.58 变量组信息配置一

- 第二种方式是执行导入时通过界面配置。

选择要导入的 Access 文件，进入下面的“导入变量”窗口。变量列表中的“变量组信息”一列显示为该变量所属的变量组路径。



图 5.59 变量组信息配置二

在变量列表中选择一个或多个变量，单击按钮“更改变量组名”，弹出如下窗口。




图 5.60 变量组信息配置三

输入要设置的变量组完整路径，单击“确定”完成配置。配置完所有变量后，单击按钮“导入”，将所有变量导入到数据字典中，并自动分配到各个变量组中。

5.7.3.2 将数据词典导入到其它工程

使用方法参照 5.7.3.1 章节。

 注意：

1、由于该版本从 Access 数据库中导入变量时没有校验，所以只适用于将导出的数据库导入到原来的工程，或者另一个工程，如果使用者自己编辑 Access 数据库在进行导入，可能会引起一些意外的错误，请慎用！

2、该功能不支持结构变量。

5.7.4 获取变量使用情况信息和删除变量

工程人员往往需要知道工程中变量的使用情况，如变量在哪些地方被引用了，有哪些变量没有被工程所使用等。组态王中都一一提供了相应的工具。

5.7.4.1 如何得到变量使用信息

变量在工程中可能被多处引用，组态王提供了“变量使用报告”功能，为用户准确提供变量引用情况。在选择工程浏览器中的菜单“工具\变量使用报告”，系统出现一个信息提示框，表明系统正在调入画面等，查找变量的引用。当查找完成后，弹出“变量使用报告”对话框，如图 5.61 所示。



图 5.61 变量使用报告

在“变量使用报告”中使用树状结构列出了所有的变量的使用和未使用情况，变量在哪个命令语言中被使用了，在画面中被引用的位置（坐标）。如果没有被引用，则在该变量的节点处没有含有子节点的标记。

在变量使用报告的状态栏中显示当前“可使用点数”，即使用的组态王加密锁点数；和“已使用点数”，即已经使用的变量数。用户据此可以判断还可以使用的点数。

点击“保存”按钮，弹出文件保存对话框，可以将变量使用报告保存为.csv 格式的文件。

点击“查找”按钮，弹出“查找”对话框，如下图 5.62 所示。



图 5.62 查找

查找变量有两个选项：“精确查找”、“模糊查找”，如果选择“精确查找”，则在查找的变量名编辑框中输入准确的变量名称，单击“确定”按钮开始查找。如果选择“模糊查找”，则在查找的变量名编辑框中输入变量名称的头若干个字符，单击“确定”按钮开始查找。确定后在变量使用报告的对话框中变量使用信息直接定位到这个变量，这样用户可以很方便的查找到某个变量的情况。（注意“找到后进行编辑”在此处不起作用）

5.7.4.2 变量使用更新

使用变量使用更新，可以在增加变量，修改变量引用后，保证变量使用情况的确切信息。

在工程浏览器中选择“工具\更新变量计数”菜单，系统出现更新信息提示条，自动更新变量使用情况。这样可以在查看变量使用报告时使引用情况更准确。

5.7.4.3 删除未用的变量

如果某些变量在工程中未被使用，可以将其删除。删除时，有两种方法：直接在变量词典中删除；使用组态王提供的未使用变量列表中删除。

如果用户确定数据词典中的某个变量未被使用，可以在变量词典中直接选中该变量，单击鼠标右键，在快捷菜单中选择“删除”，系统提示是否确定，选择确定后，将直接永久性删除变量。

如果用户不能确定，可以选择工程浏览器中的“工具\删除未用变量”菜单，系统会弹出“删除未用变量”对话框，如图 5.63 所示，在列表中列出当前工程中定义的未被使用的变量，点击选择要删除的变量，按住<Shift>键任意多选。单击“确定”按钮，将永久性删除选中的变量。如果确定要全部删除列表中的变量，直接单击“全选”按钮，所有列表中的变量将被选中，单击“确定”按钮删除。

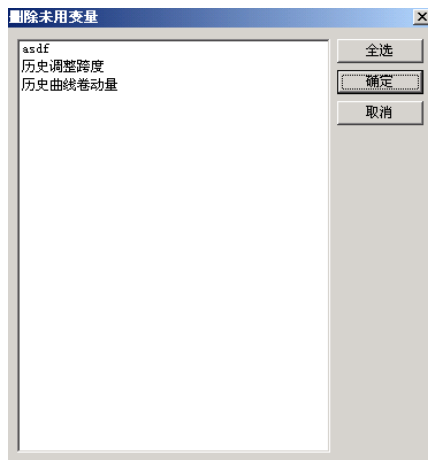



图 5.63 删除未用变量

 注意：

1、 为保险起见，在删除变量时建议用户不要直接从数据词典中删除变量，而是使用第二种方法删除。

2、 在删除未用变量之前，使用“更新变量计数”刷新系统中变量的使用情况。

5.7.5 变量属性修改和变量替换

5.7.5.1 查找数据库变量

进入工程浏览器中的数据词典，选择菜单“工具\查找数据库变量”，或直接在数据词典中单击鼠标右键，在快捷菜单中选择“查找数据库变量”，系统弹出“查找变量”对话框，如图 5.64 所示。查找变量有两个选项：“精确查找”、“模糊查找”，如果选择“精确查找”，则在查找的变量名编辑框中输入准确的变量名称，单击“确定”按钮开始查找。如果选择“模糊查找”，则在查找的变量名编辑框中输入变量名称的头若干个字符，单击“确定”按钮开始查找。如果选择了“找到后进行编辑”选项，则系统会在数据词典中从头开始查找与要查找的字符相匹配的变量，找到后，系统自动打开变量属性对话框。用户可以编辑变量属性。

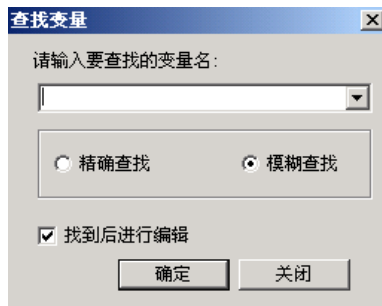


图 5.64 查找变量

5.7.5.2 多选修改变量属性

组态王提供了允许用户同时选择多个变量（不包括结构变量）修改变量属性的功能。如图 5.65 所示，在数据词典中选择要修改属性的变量，选择时，按住<Ctrl>键任意多选，按住<Shift>键选择相邻的多个变量。单击鼠标右键，在快捷菜单中选择“编辑 选种的变量”命令，系统弹出变量属性对话框，如图 5.60 所示。相同的变量属性是允许用户进行编辑的，而不同的属性选项则为灰色，不允许编辑。如果各变量的属性参数定义相同则在相应项中显示出来；如果相同的属性原来各变量定义的参数不相同，则系统在该项中不显示任何内容，如果用户不修改，则确认后，系统保留这些项原来的值，如果做了修改，则所有选中变量的对应属性都变成被修改的值。

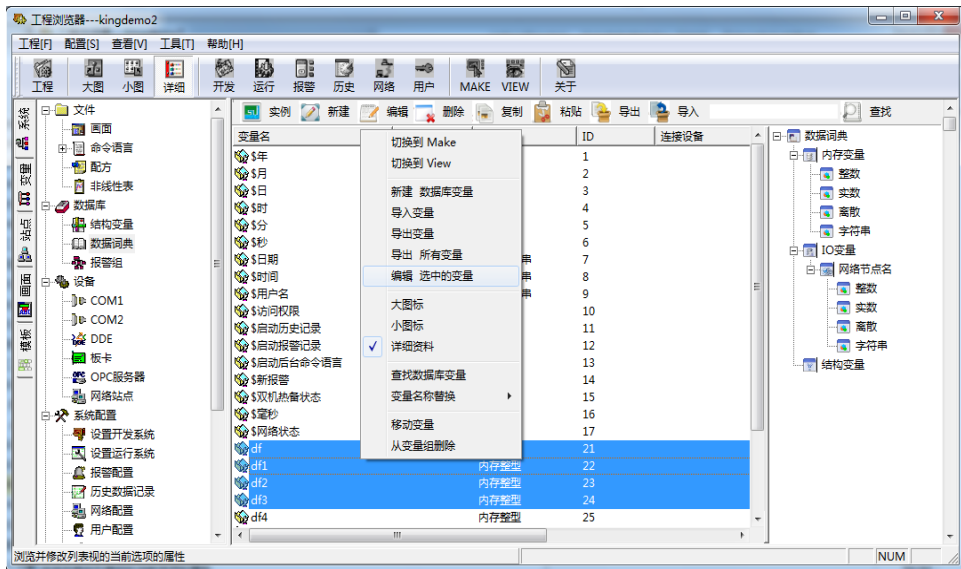


图 5.65 多选变量

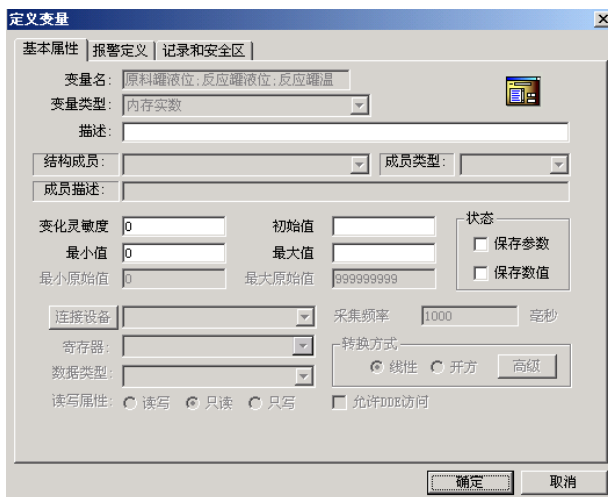


图 5.66 修改多选变量的属性

5.7.5.3 变量替换

组态王提供了变量替换的功能，不仅可以对本工程中的变量进行变量名称替换、变量使用替换，还可以对远程站点变量或非本工程变量进行站点名替换、标识符替换等。用户可以在工程中不同的范围内使用不同的替换方式进行变量替换。

组态王具有变量单个替换和批量替换的功能。

5.7.5.3.1 单个替换

◇ 替换范围

组态王变量替换按替换的范围大小可分为 4 种：

- 整个工程，包括所有画面和脚本等
- 选中的画面组中的所有画面和脚本

- 当前画面及脚本
- 当前画面中选中的图素

◇ 替换方式

变量替换有以下几种方式：变量更名、变量替换、站点名替换、标识符替换为变量和变量中字符串替换。下面以“整个工程”范围替换为例分别介绍各种替换方式，其它替换范围的替换方式含义相同。

1) 整个工程范围变量替换

整个工程变量替换包含全部 4 种替换方式。

- **变量更名（更改数据词典）**

在工程浏览器中选择菜单“工具/替换变量名称/单个替换”，或在数据词典中选择右键菜单“变量名称替换/单个替换”，弹出如下图 5.67 所示对话框。对已定义和使用的变量更换变量名称，同时系统也会自动修改数据词典中对应的变量名称。

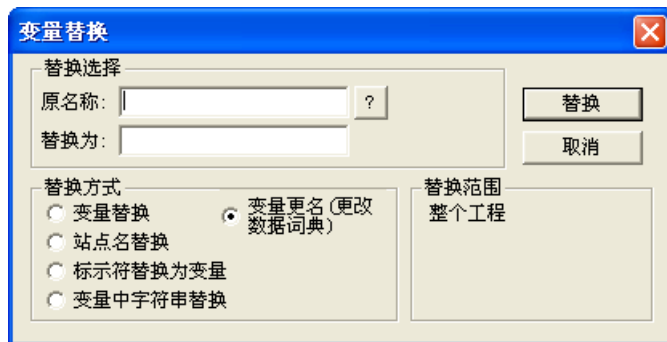


图 5.67 整个工程范围变量更名

其中“原变量”必须是工程中存在的变量，即已定义的变量；可以点击“？”按钮选择变量。在“替换为”编辑框中输入新的变量名称，该变量名称不能与数据词典中已有的变量名称、其它名称相同，要符合组态王变量命名规则。

输入要替换的内容后，点击“替换”按钮，将工程中指定的变量名替换为新的变量名，“原变量”不存在于工程中。

这种替换方式只在整个工程和画面组范围使用。

- **变量替换**

如下图 5.68 所示，在变量替换对话框“替换方式”中选择“变量替换”。这种替换方式为将工程选择范围内的画面和脚本等中已使用的“原变量”替换为指定的“替换为”中的变量。其中“原变量”和“替换为”所输入的变量必须在当前工程中存在，而且其数据类型必须相同。否则不能完成替换。

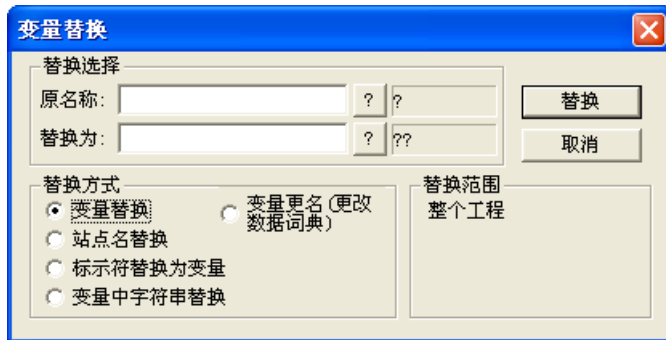


图 5.68 整个工程范围变量使用替换

- **站点名替换**

组态王网络功能中提供了远程变量引用功能。而如果用户在服务器和客户端要浏览相同的画面，可以把服务器上的画面直接导入到客户端的工程中，然后在客户端工程中使用“站点名替换”，将变量的站点名称自动修改为远程服务器的名称，可以大大减少工程人员的工作量。

如下图 5.69 所示，在“原名称”中输入本地工程要被替换的站点名，在“替换为”编辑框中输入目标站点名，然后选择“替换”按钮，系统自动完成站点名称替换。

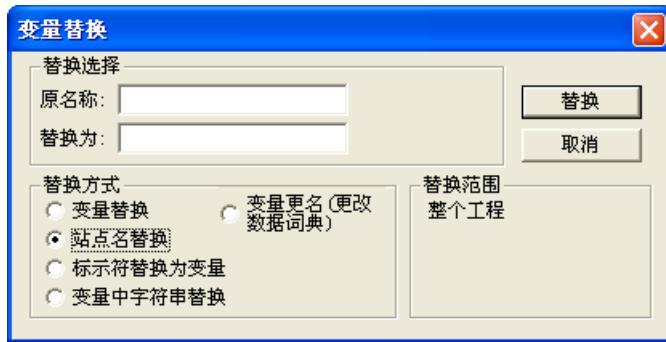


图 5.69 整个工程范围站点名称替换

- 标识符替换为变量

组态王提供了画面和脚本的导入、导出功能。在这些操作完成后，在目标工程中被导入的画面或脚本中可能会含有一些本工程不存在的变量，组态王将其作为一个“标示符”。如果不替换这些变量，在组态王运行时，画面对应点的数据无法刷新，脚本无法执行。所以可以使用“标示符替换为变量”功能来对“标示符”进行替换。

如下图 5.70 所示，在“原名称”编辑框中输入要被替换的“标示符”，在“替换为”编辑框中输入本地已定义的变量，或通过“？”按钮进行选择。输入完成后，选择“替换”按钮完成替换。

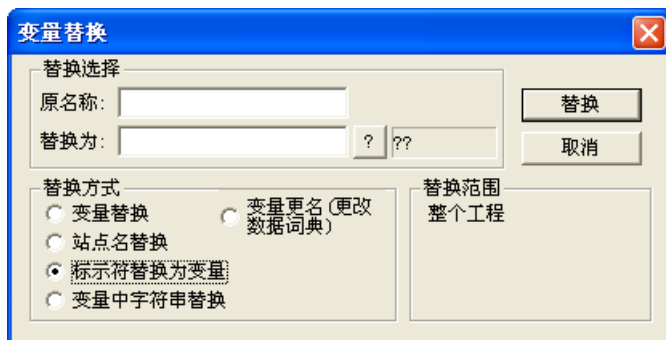


图 5.70 整个工程范围标示符替换

- 变量中字符串替换

在实际应用中常会遇到两个项目背景和监控内容基本相同的情况，此时如果可以把 A 项目中数据词典里的变量名称做系统的修改用到 B 项目中，如把变量中的 A 站点标识符改为 B 站点标识符，将会减少很大工作量。为此，组态王 6.55 以后版本中新增了替换变量中字符串的功能。通过此功能，输入要替换的 A 标识符和替换为的 B 标识符，即可把所有具有 A 标识符的变量中的 A 标识符替换为 B 标识符(该功能对远程站点的工程增加相似的新站点也很有用)。

变量中字符串替换就是把包含指定字符串的变量中的指定字符串替换为另一个字符串。该替换功能允许对本地变量，远程变量，当前画面和选中的图素进行替换操作。

在工程浏览器中，选择“工具/替换变量名称/单个替换”，弹出如图 5.71 所示的对话框。

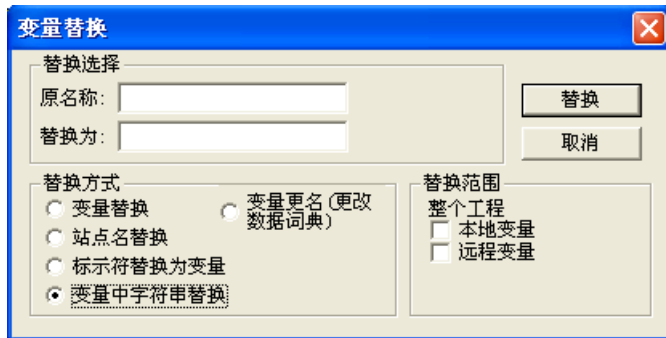


图 5.71 整个工程的变量中字符串替换

在“原名称”编辑框中输入要被替换的标示符，在“替换为”编辑框中输入要替换为的标示符，在替换方式中选择“变量中字符串替换”选项。

这里的替换范围是针对整个工程中的变量（包括所有画面和命令语言），并把整个工程中的变量分为两类：本地变量和远程变量。对于包含指定标示符的整个工程中的变量，可以选择替换整个工程中的本地变量，或者替换整个工程中的远程变量，或者是替换所有变量（包括远程和本地）。

该功能是组态王 6.55 以后版本的一个重要特性，如果能够巧妙地运用它，将会起到事半功倍的效果。

下面简单介绍其它范围替换变量的方法。

2) 对选中的画面组进行变量替换

点击工程浏览器的画面页签，选择要进行变量替换的画面组，点击右键，弹出浮动菜

单，如图 5.72 所示。

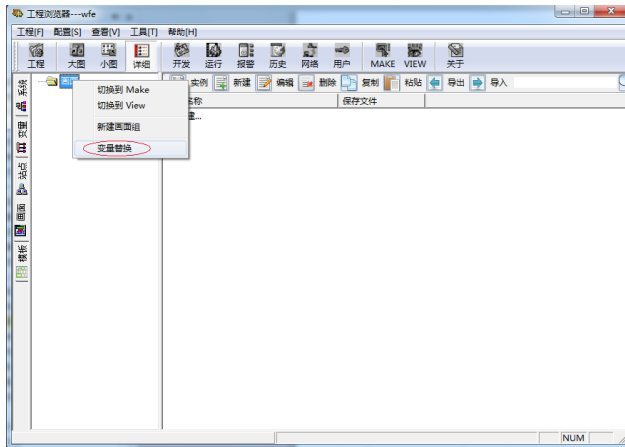


图 5.72 画面组选择变量替换

选择“变量替换”，弹出变量替换对话框，如图 5.73 所示。

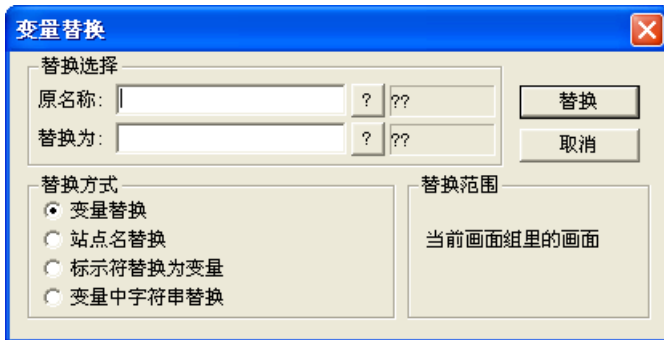


图 5.73 画面组变量替换窗口

其中的替换方式和使用方法与整个工程范围替换相同。其变量的替换范围为当前选择的画面组及其子画面组中的所有画面和脚本。

3) 对当前画面及脚本进行变量使用替换

在画面开发系统中，打开某个画面，不选择任何图素，选择菜单“编辑/变量替换”，弹出变量替换对话框，如图 5.74 所示。

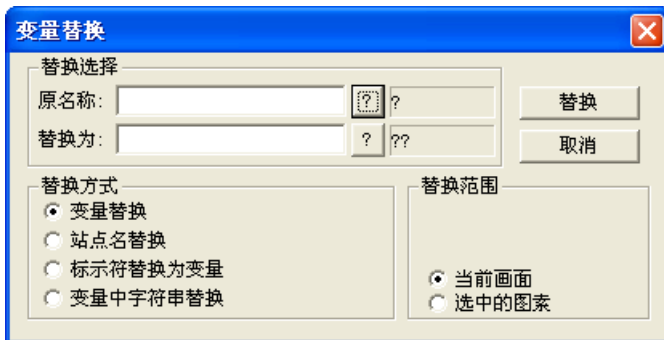


图 5.74 画面和图素变量替换

在对话框中的“替换范围”中选择“当前画面”。

其中的替换方式和使用方法与整个工程范围替换相同。其变量的替换范围为当前选择的画面中的所有图素和脚本。

4) 对当前画面选中的图素进行变量使用替换

在画面中选择要进行变量替换的图素，然后选择右键菜单“变量替换”，弹出变量替换对话框，如图 5.75 所示。

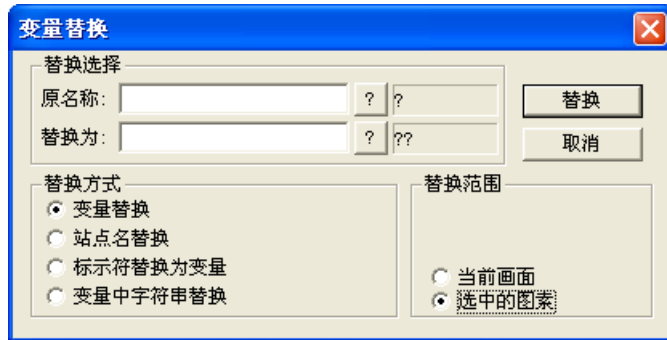


图 5.75 图素变量替换窗口

在对话框中的“替换范围”中选择“选中的图素”。

其中的替换方式和使用方法与整个工程范围替换相同。其变量的替换范围为当前选择的画面中的所有图素和脚本。

5.7.5.3.2 批量替换

1. 变量替换：实现将原变量替换为目标变量。宜用范围：内存变量、IO 变量、系统变量、结构变量。

变量更名：将原变量更名。包括名称替换、添加前缀和添加后缀三种。

2. 在工程浏览器中选择菜单“工具/替换变量名称/批量替换”，或在数据词典中选择右键菜单“变量名称替换/批量替换”，弹出如下图 5.76 所示对话框。对已定义和使用的变量更换变量名称，同时系统也会自动修改数据词典中对应的变量名称。

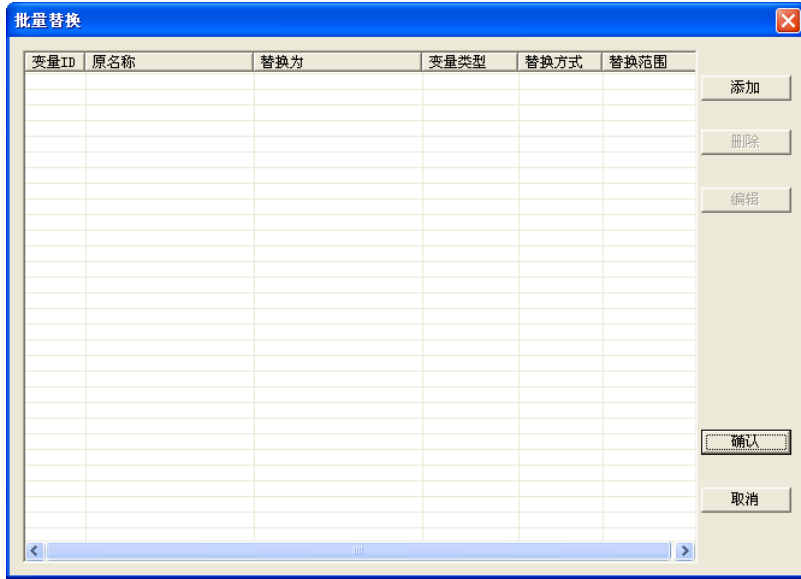


图 5.76 批量替换

◇ 加入变量

点击“添加”按钮，选择需要添加的变量，弹出“加入变量”对话框，如图 5.77 所示。单击“确定”，完成变量的加入。



图 5.77 加入变量

◇ 配置替换选项

选择某一项或几项变量，点击“编辑”按钮或在键盘上按下“Enter”键，打开“编辑选项”窗口。也可鼠标双击某一项，打开“编辑选项”对话框，如图 5.77 所示。如果选择的变量中含有系统变量或结构变量，则在变量编辑选项对话框中只能进行变量替换功能。如图 5.78。

单击“变量名”中的“替换为”显示框右边的变量列表键，弹出变量列表框，选择替换后的变量。



图 5.78 变量替换对话框

如果选择的变量中不含有系统变量和结构变量，则可以在变量编辑选项对话框中选择变量更名功能。

在“批量替换”栏中选择一个变量，则弹出图 5.79 的“编辑选项”框，选择替换方式中的“变量更名”，在“变量更名选项”中选择需要的方式，在新名称的显示栏进行操作，然后单击“确认”按钮，系统自动完成名称的替换。



图 5.79 变量更名对话框-名称替换

在“批量替换”栏中选择的变量多于一个时，只能对变量名进行“添加前缀”和“添加后缀”的操作，如图 5.80 所示。



图 5.80 变量更名对话框-增加前缀

各项变量配置后的信息可以在“批量替换”主对话框中看到，如图 5.81 所示。



图 5.81 批量替换结果

如果有一个变量没有编辑修改，在单击“确认”时，则弹出如下窗口：

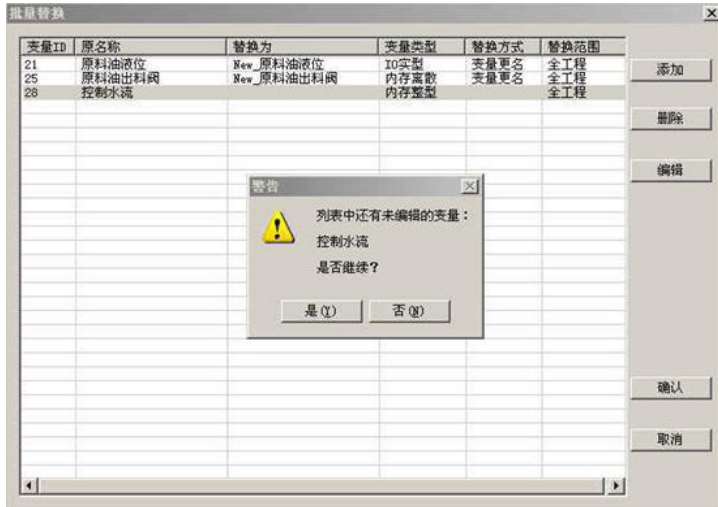


图 5.82 批量替换是否继续

单击“是”，对编辑修改过的变量进行操作，没有做任何编辑的变量保持原来的状态不变；单击“否”，继续对“批量替换”窗口进行操作。

◇ 删除变量：

选中一条或多条记录，单击“删除”或是在键盘上按下“Delete”键，删除所选变量。全部变量配置好后，单击“确认”，开始对变量列表中的变量进行替换或更名操作。

5.8 关于自定义变量

为方便用户使用，组态王在命令语言中提供了用户自定义变量的功能。用户在命令语言中声明变量类型和变量名称，然后同组态王变量一样，直接在命令语言中用于计算、赋值等操作。自定义变量的作用区域为当前使用的命令语言模块，当命令语言执行完成后，

系统将自动释放该命令语言中的自定义变量。自定义变量有自定义基本变量和自定义结构变量两类。

5.8.1 自定义变量

组态王中的数据类型有 BOOL、LONG、FLOAT 和 STRING 四种类型。自定义变量的数据类型也有这几种。在命令语言中定义变量如下图 5.83 所示。自定义变量在命令语言中使用前，先需要定义。然后才可以使用。自定义变量只有值，没有变量域的概念，所以自定义变量不能引用变量的域。自定义变量可以与组态王变量进行数据交换，相互赋值。可以作为自定义函数的参数使用，也可以作为组态王函数的参数使用。自定义变量名称在使用时区分大小写。

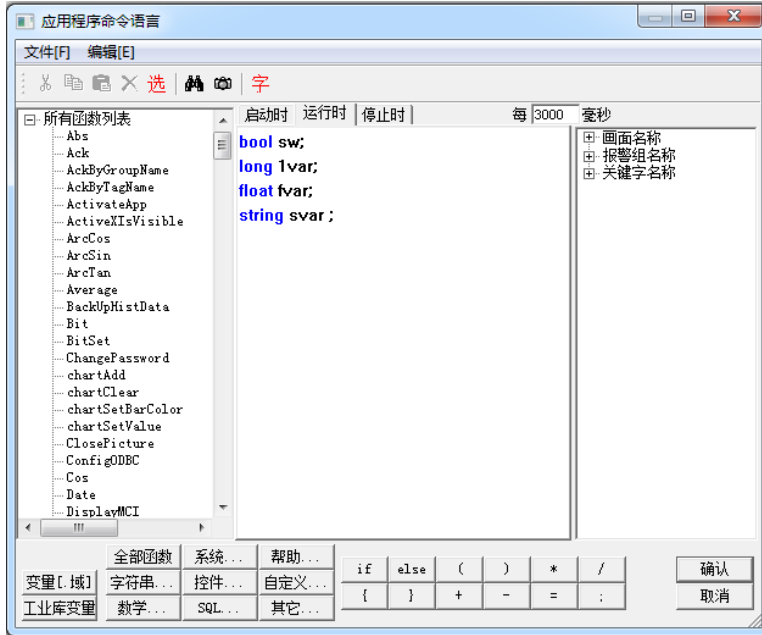


图 5.83 自定义变量

自定义变量在命令语言中可以随时定义，随时使用，不占系统点数。方便灵活。如下图 5.84 所示，为自定义变量在自定义函数中的一个应用。

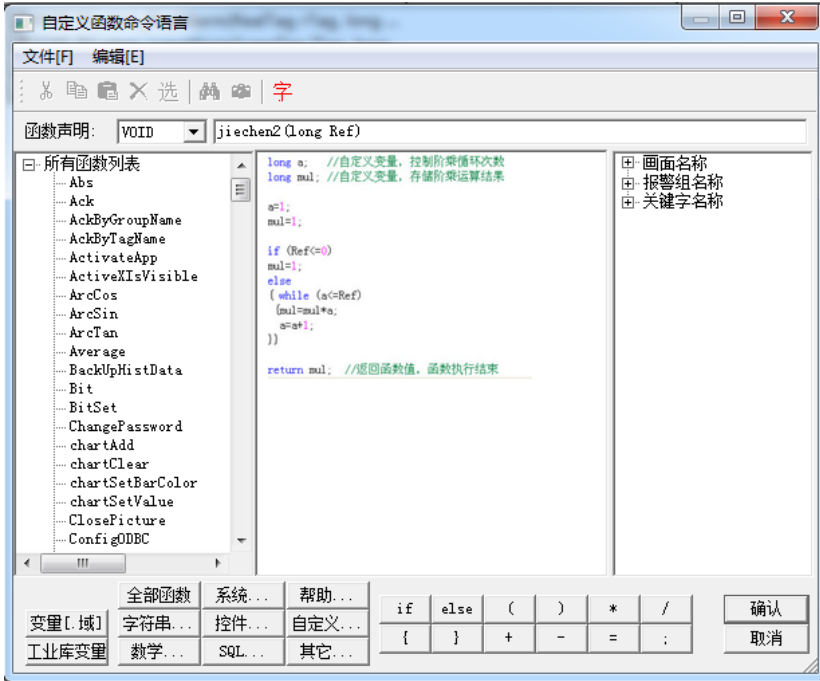


图 5.84 自定义变量的使用

5.8.2 自定义结构变量

如果工程中已经定义了结构模板，那么在命令语言中就可以使用自定义结构变量，其作用域是当前引用的命令语言模块。与自定义变量不同的是，自定义结构变量无须在使用前定义其数据类型，因为引用结构的结构成员的数据类型已经在建立结构变量时定义了。

如在工程中建立结构：Astru，其成员有：Member1、Member2、Member3。则在命令语言中使用时，用下面的方法：

```
Astru a;
```

a. Member1=10;

a. Member2=20;

.....

下面以本章 5.3 节中所创建的结构变量为例,说明自定义结构变量的使用方法。如下图 5.85 所示为创建的结构变量。

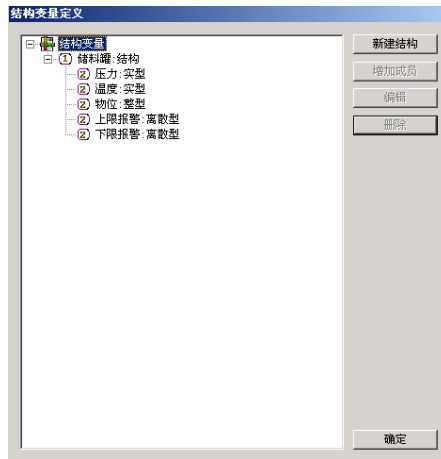


图 5.85 创建的结构变量

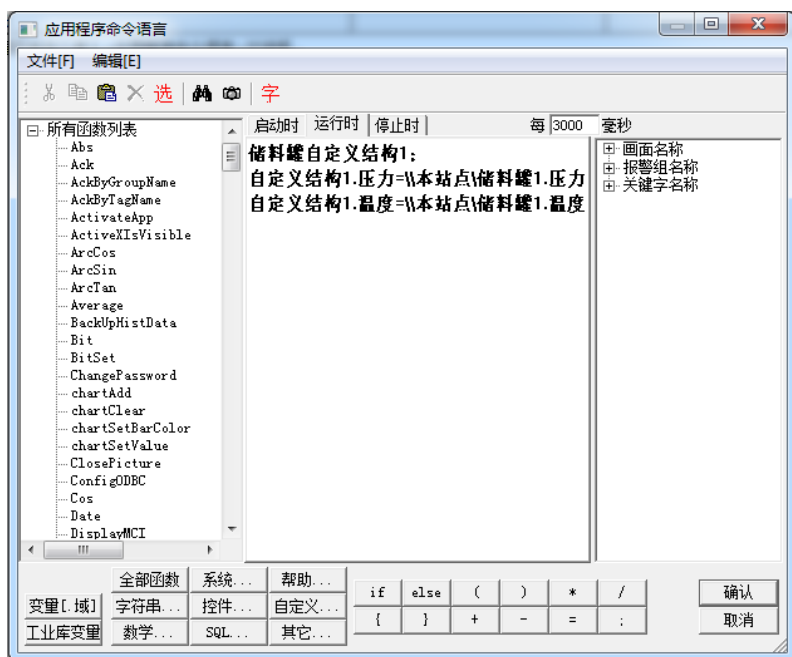


图 5.86 自定义结构变量的使用

自定义结构变量也可以作为自定义函数的参数使用。相同的结构可以做整体赋值。

! 注意:

- 1、自定义变量和自定义结构变量都没有域的概念，所以在使用时不能引用自定义变量的域。
- 2、自定义结构变量目前不支持远程结构。

在组态王中，除了以上介绍的各种变量外，还有一些特殊的变量：

- ◇ **报警窗口名称：**即工程中每定义一个报警窗口，其名称就为组态王中的一个变量
- ◇ **历史趋势曲线名称：**组态王工具箱中和图库中的历史趋势曲线定义后，其名称也将

成为组态王中的一个变量

以上两种特殊类型的变量，在组态王数据词典中是无法看到的，但在变量浏览器中可以看到。该变量也有属性，供用户引用，修改报警窗和历史趋势曲线的参数，具体属性请参见本章 5.4 节。

5.9 变量指针引用

在工程实际中，一个工程中可能有多个同样的设备，所要采集的变量值也相同。比如在电力监控系统中，有多台电机或变压器，需要对它们的电压、电流、功率等参数进行采集，而这些电气设备在定义变量时具有相同的类型和参数。若要对多组变量分别进行监控，要做很多重复性的工作，并且使整个监控画面变的很复杂。为了解决这一问题，我们引入了变量引用的概念，通过变量引用，可以用一组（个）内存变量分时对多组（个）I/O 变量进行监控，这样就会大大的减少工作量，提高工作效率，也会使整个监控画面变的整洁、美观。

5.9.1 普通变量引用

满足普通变量引用的条件是有多个变量类型相同的被引用变量。这里我们以整型变量为例来说明普通变量引用的过程。

假设有三个被引用的 I/O 变量，分别为 P1_CITED、P2_CITED、P3_CITED。这时我们要创建一个与 I/O 变量类型相同的内存变量 P_CITE，在变量引用函数中用作引用变量。如下图 5.87 所示。

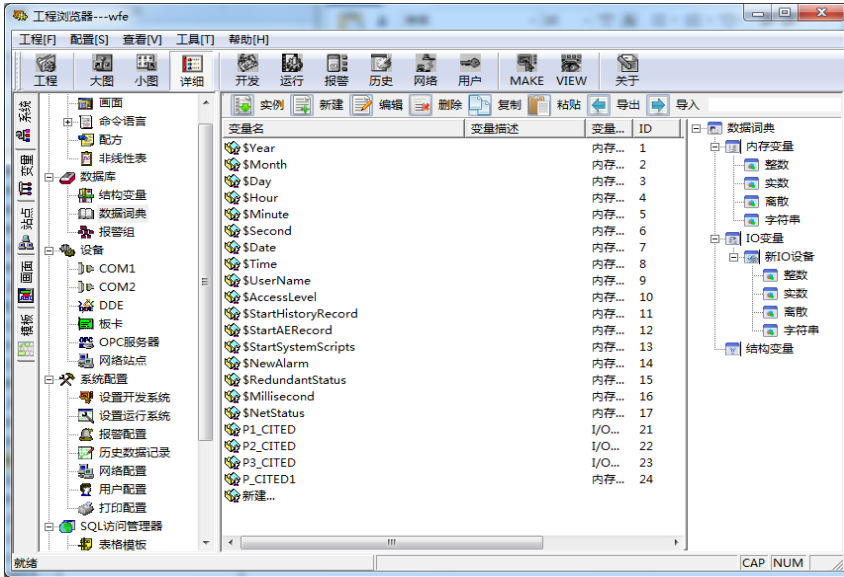


图 5.87 普通变量的引用变量和被引用变量

创建完引用变量后,开始新建一个画面,在画面中分别创建三个按钮,分别在按钮“弹起时”命令语言中加入下面语句:

```
VarRefAddress (“\\本站点\P_CITE”, “\\本站点\P1_CITED”);
```

```
VarRefAddress (“\\本站点\P_CITE”, “\\本站点\P2_CITED”);
```

```
VarRefAddress (“\\本站点\P_CITE”, “\\本站点\P3_CITED”);
```

普通变量的引用过程到此结束,接下来我们对变量引用进行测试。给所有变量进行数值连接,启动运行系统,分别点击三个按钮,我们可以看见随着点击不同的按钮,就可以引用到不同的变量。

使用 `VarRefAddress (“\\本站点\P_CITE”, “”)` 可以解除对变量“\\本站点\P_CITE”

的引用。



注意：

引用变量只能是内存变量，被引用变量可以是内存变量和 IO 变量，但不能作为引用变量；引用变量必须是本地变量；被引用变量可以是本地变量也可以是远程站点下的变量；被引用变量的数值、质量戳、时间戳、描述、最大值和最小值属性将被传递到引用变量中。

引用变量和被引用变量的最大值、最小值一致。如果想引用被引用变量的报警属性，引用变量必须定义报警属性或勾选事件。

Web 上使用变量引用函数 **VarRefAddress**，该函数用法与在组态王运行系统中的用法基本相同，但引用变量和被引用变量都不能是结构变量。

引用远程变量的域必须在画面上有连接，否则无法完成引用。



例如：

当引用远程变量 A 的域 TimeTimeString 时，这个域必须在发布的画面中建立连接，才能被引用到引用变量中。

5.9.2 结构变量引用

结构变量的引用和普通变量的引用非常类似。其条件也是有多组被引用变量，然后再建立一组内存变量。需要注意的是引用结构变量和被引用结构变量成员数相同，成员类型

相同。这里我们也以整型变量为例说明结构变量引用的过程。

假如我们要对三台变压器的电流（I）、电压（V）、功率（P）分别进行监控，那我们首先要创建一组结构变量 Transformer，其中包含三个成员变量 I、V、P。然后在数据词典中建立三个 Transformer 型的变量 Transformer1、Transformer2 和 Transformer3，I、V、P 的数据类型全选 I/O 整型，最后再建立一个 Transformer 型的变量 Transformer_Cite，I、V、P 的数据类型全选内存整型，用来引用 I/O 结构变量。如下图 5.88 所示。

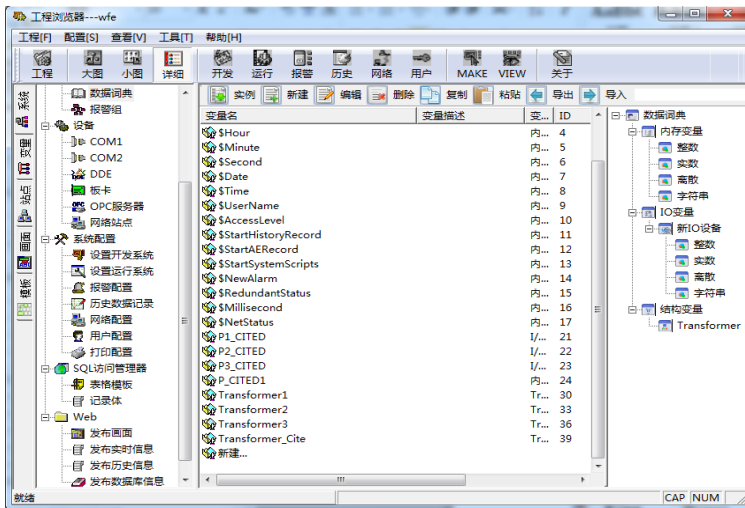


图 5.88 结构变量的引用变量和被引用变量

创建完引用变量后，开始新建一个画面，在画面中分别创建三个按钮，分别在按钮命令语言中加入下面语句：

```
StructVarRefAddress("\\ 本站点 \Transformer_Cite", "\\ 本站点
```

```
\Transformer1");
```

```
    StructVarRefAddress("\\本站点\Transformer_Cite", "\\本站点
```


```
\Transformer2");
```

```
    StructVarRefAddress("\\本站点\Transformer_Cite", "\\本站点
```

```
\Transformer3");
```

结构变量的引用过程到此结束，接下来我们对变量引用进行测试。给所有变量进行数值连接，启动运行系统，分别点击三个按钮，我们可以看见随着点击不同的按钮，就可以引用到不同的变量。

使用 StructVarRefAddress (“\\本站点\Transformer_Cite ”, “”) 可以解除对变量 “\\本站点\Transformer_Cite ”的引用。

 注意：

引用结构变量和被引用结构变量成员数相同，成员类型相同，并且成员变量的限制条件和非结构变量相同。

5.9.3 结构变量和普通变量之间的引用

下面我们来讨论一下结构变量和普通变量之间的引用。由于普通变量是单个变量，而结构变量是多个变量的集合，所以普通变量只能和结构变量的成员变量之间建立引用关系。拿上例中的 P_CITE 和 Transformer1 中的 I 变量来说，要在它们之间实现引用，P_CITE 应为引用变量，Transformer1.I 应为被引用变量，可用下面的语句实现：

```
VarRefAddress("\\本站点\P_CITE", "\\本站点\Transformer1.I");
```

但在这里同样需要满足变量引用的条件，即引用变量只能是内存变量，被引用变量不能作为引用变量，引用变量和被引用变量的最大值、最小值一致。

5.10 变量的驱动转发

上一节介绍了变量指针引用的相关知识，该节将讲述一种简单的数据转发和回写的功能（两个组态王之间）- 组态王的一个新特性，即驱动转发。运用该功能可以解决在进行大量数据转发时的脚本编写工作，在很大程度上提高了工作效率和准确度。下面以 ModbusTcp 为例讲述基于 TCP 协议的驱动转发。

- ◇ 前提条件：在两个不同的电脑上，安装组态王 A 和 B。
- ◇ 要实现的功能：组态王 A 采集到的数据，通过驱动转发的方式发送到组态王 B，同时通过组态王 B 往组态王 A 所连接的设备上写数据。
- ◇ 该功能使用情况：用于两台组态王不能通过网络通信的情况下。
- ◇ 工作原理：如下图所示。

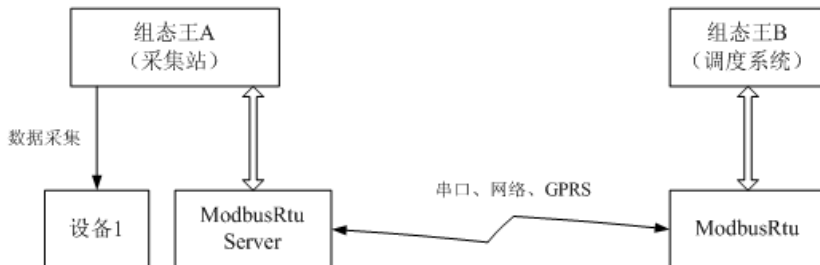


图 5.89 驱动转发原理图

在组态王 A 上定义一个 I/O 设备，如仿真 PLC（详细步骤请参考“第六章 I/O 设备管理”）和一个额外的逻辑设备 ModbusTcpServer（在设备配置向导中，选择 PLC/莫迪康 /ModbusTcpServer），如图 5.90 所示。



图 5.90 创建逻辑设备 ModbusTcpServer

在组态王 A 中定义一个只读的 short 型的 I/O 变量“原料油液位”关联仿真 PLC 的某个寄存器，同时定义一个只写的 short 型 I/O 变量“原料油液位_ref”关联 ModbusTcpServer 中的寄存器 400001(关于 I/O 变量定义的详细步骤请参考该章前面的内容)。

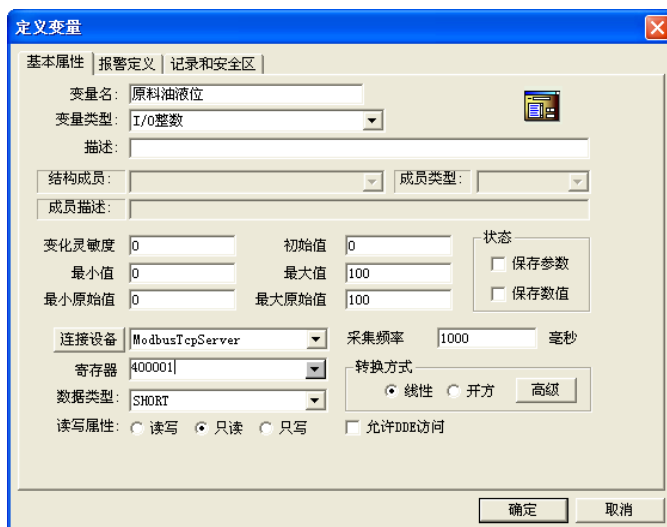


图 5.91 ModbusTcpServer I/O 变量的定义

在组态王 B 中定义了一个逻辑设备 ModbusTcp（在设备配置向导中，选择 PLC/莫迪康/ModbusTcp）。这里要设置 ModbusTcp 地址中的 IP 地址为组态王 A 的 IP 地址，实现组态王 A 和 B 的链接。

定义一个只读的 short 型 I/O 变量“参考液位”，并将它关联到 ModbusTcp 的寄存器 400001。

这样，当组态王 A 采集到变量“原料油液位”的数据后，发现有和它相关的变量“原料油液位_ref”，就会把采集到的值也写到“原料油液位_ref”的寄存器 400001 中一份，从而，当在组态王 B 采集周期到来时，就可以采集到 ModbusTcpServer 中寄存器 400001 中的数据，完成两个组态王之间的数据转发工作。

对于组态王 B 往组态王 A 所连接的设备上写数据，只需要按照下表的第三行的规定在

组态王 A 和 B 上定义相关的变量，即可实现。

上面的操作步骤，用户需要注意以下的两个方面：

✧ 组态王 A 和 B 中定义的变量名称和读写类型必须满足如下的规定：

	组态王 A		组态王 B
读变量	xx（只读）	xx_ref（只写）	yy（只读）
写变量	xxx（只写）	xxx_ref（读写）	yyy（只写）

注：1) 表中“xx”，“xxx”，“yy”，“yyy”可以是任意变量名，即是上例中“原料油液位”和“参考液位”的推广。

2) “_ref”必须小写。

3) xx_ref 和 yy 必须关联同一个寄存器，例如 400001。

4) xxx_ref 和 yyy 必须关联同一个寄存器。

5) yyy 可以是读写类型。

✧ 组态王 A 中定义的和 ModbusTcpServer 相关联的变量 xx_ref 和 xxx_ref 不需要在画面中定义，所以定义 ModbusTcpServer 设备时取消动态优化。

5.11 自动建变量

组态王 6.55 以后版本增加的一个与变量有关的新特性 - 自动建变量。该功能配合自动建变量的驱动，使用户可以在完成设备定义之前（关于设备定义的详细步骤请参考“第六章 I/O 设备的管理”），选择是否自动创建批量的变量到数据词典中。

在“设备安装向导---信息总结”页面中增加了“自动建变量”复选框，如图 5.92

所示。



图 5.92 自动建变量

若勾选“自动建变量”选项，点击“完成”按钮，则系统将调用 XXX_BatchVar.dll 文件(XXX 表示组态王下某个设备驱动名,例如组态王下某个驱动为 KingViewDriver.dll, 则用于该驱动自动批量建变量的产品为 KingVieDriver_BatchVar.dll)的接口,完成 MDB 的生成。返回给一个字符串,表示 MDB 文件的路径。打开“导入变量”窗口,通过这个路径完成变量的导入。

⚠ 注意：“自动建变量”的复选框,依据驱动目录下是否有 XXX_BatchVar.dll 文件来判断是否为灰色。即：没有 XXX_BatchVar.dll,为灰色(不可选)；有,为可选。

第六章 I/O 设备管理

- ☞ 介绍组态王 7.0 与外围设备的连接。
- ☞ 教您如何配置 DDE、板卡、PLC 等设备的驱动程序。
- ☞ 介绍组态王 7.0 仿真 PLC 的使用
- ☞ 介绍组态王 7.0 对设备进行远程拨号采集数据
- ☞ 介绍组态王 7.0 对设备通过 GPRS 进行采集数据

组态王 7.0 软件系统与最终工程人员使用的具体的 PLC 或现场部件无关。对于不同的硬件设施，只需为组态王 7.0 配置相应的通信驱动程序即可。组态王 7.0 驱动程序采用最新软件技术，使通讯程序和组态王 7.0 构成一个完整的系统。这种方式既保证了运行系统的高效率，也使系统能够达到很大的规模。

组态王 7.0 支持的硬件设备包括：可编程控制器（PLC）、智能模块、板卡、智能仪表，变频器等等。工程人员可以把每一台下位机看作一种设备，不必关心具体的通讯协议，只需要在组态王 7.0 的设备库中选择设备的类型，然后按照“设备配置向导”的提示一步步完成安装即可，使驱动程序的配置更加方便。

组态王 7.0 支持的几种通讯方式：

- ◇ 串口通讯
- ◇ 数据采集板
- ◇ DDE 通讯
- ◇ 人机界面卡
- ◇ 网络模块
- ◇ OPC

6.1 设备管理

组态王 7.0 的设备管理结构列出已配置的与组态王 7.0 通讯的各种 I/O 设备名, 每个设备名实际上是具体设备的逻辑名称(简称逻辑设备名, 以此区别 I/O 设备生产厂家提供的实际设备名), 每一个逻辑设备名对应一个相应的驱动程序, 以此与实际设备相对应。组态王 7.0 的设备管理增加了驱动设备的配置向导, 工程人员只要按照配置向导的提示进行相应的参数设置, 选择 I/O 设备的生产厂家、设备名称、通讯方式, 指定设备的逻辑名称和通讯地址, 则组态王 7.0 自动完成驱动程序的启动和通信, 不再需要工程人员人工进行。

组态王 7.0 采用工程浏览器界面来管理硬件设备, 已配置好的设备统一列在工程浏览器界面下的设备分支。如图 6.1 所示:

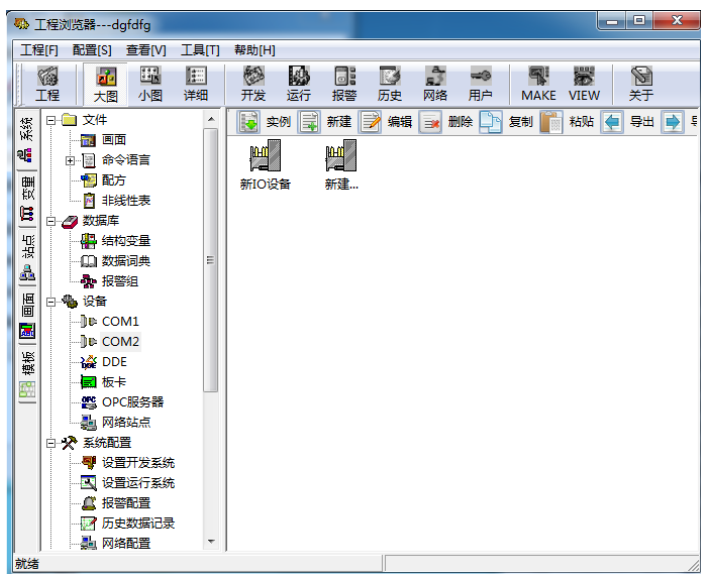
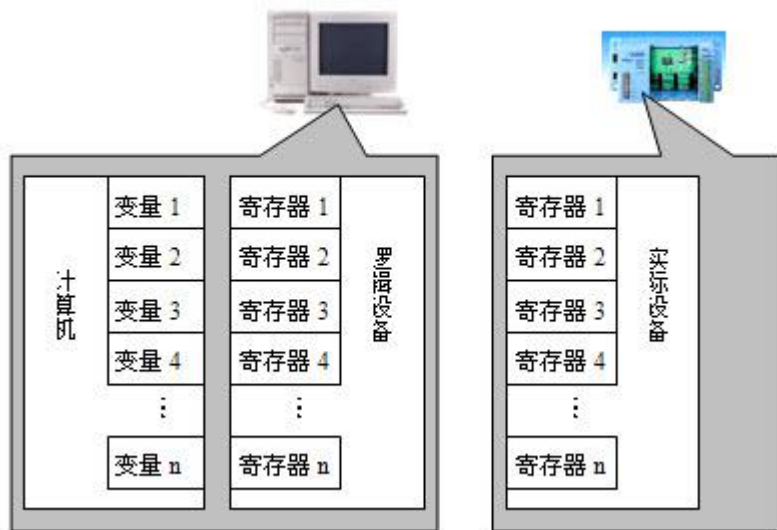


图 6.1 I/O 设备

6.1.1 了解组态王 7.0 逻辑设备概念

组态王 7.0 对设备的管理是通过对逻辑设备名的管理实现的,具体讲就是每一个实际 I/O 设备都必须在组态王 7.0 中指定一个唯一的逻辑名称,此逻辑设备名就对应着该 I/O 设备的生产厂家、实际设备名称、设备通讯方式、设备地址、与上位 PC 机的通讯方式等信息内容。(逻辑设备名的管理方式就如同对城市长途区号的管理,每个城市都有一个唯一的区号相对应,这个区号就可以认为是该城市的逻辑城市名,比如北京市的区号为 010,则查看长途区号时就可以知道 010 代表北京)。

在组态王 7.0 中,具体 I/O 设备与逻辑设备名是一一对应的,有一个 I/O 设备就必须指定一个唯一的逻辑设备名,特别是设备型号完全相同的多台 I/O 设备,也要指定不同的逻辑设备名。组态王 7.0 中变量、逻辑设备与实际设备对应的关系如图 6.2 所示。



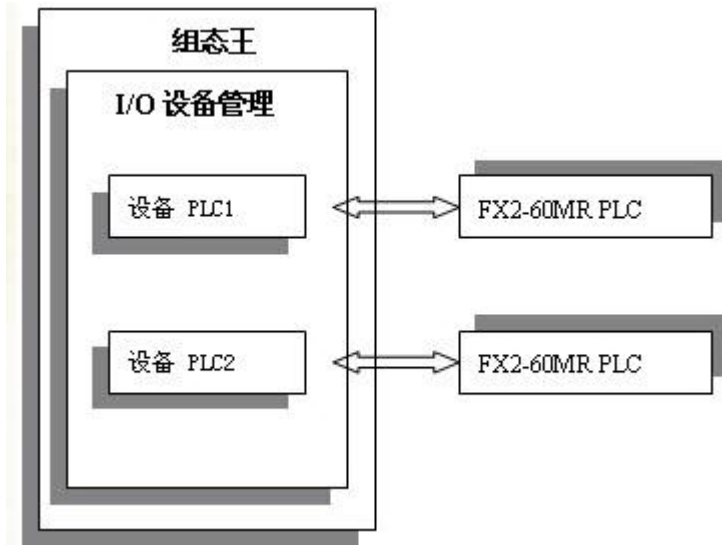
6.2 变量、逻辑设备与实际设备的对应关系



例如:

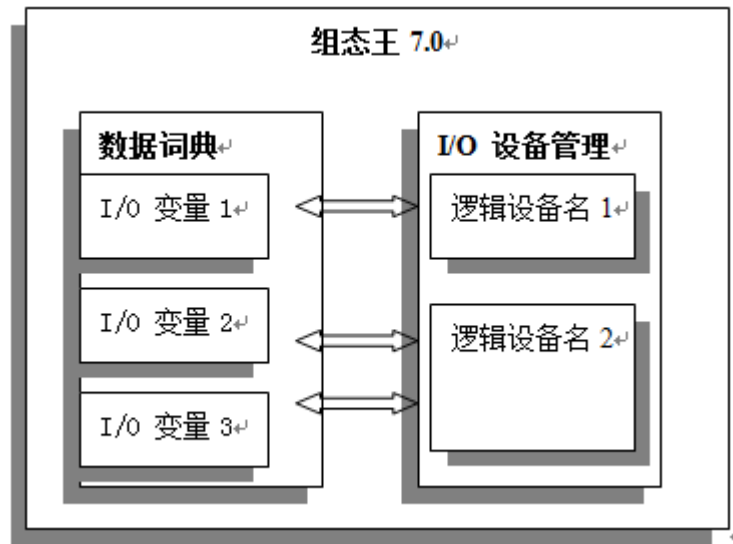
设有二台型号为三菱公司的 FX2-60MR PLC 作下位机控制工业生产现场，同时这两台 PLC 均要与装有组态王 7.0 的上位机通讯，则必须给两台 FX2-60MR PLC 指定不同的逻辑名，如图 6.3 所示：

其中 PLC1, PLC2 是由组态王 7.0 定义的逻辑设备名(此名由工程人员自己确定)，而不一定是实际的设备名称。



6.3 逻辑设备与实际设备示例

另外，组态王 7.0 中的 I/O 变量与具体 I/O 设备的数据交换就是通过逻辑设备名来实现的，当工程人员在组态王 7.0 中定义 I/O 变量属性时，就要指定与该 I/O 变量进行数据交换的逻辑设备名，I/O 变量与逻辑设备名之间的关系如图 6.4 所示：



6.4 变量与逻辑设备间的对应关系

一个逻辑设备,可与多个 I/O 变量对应。

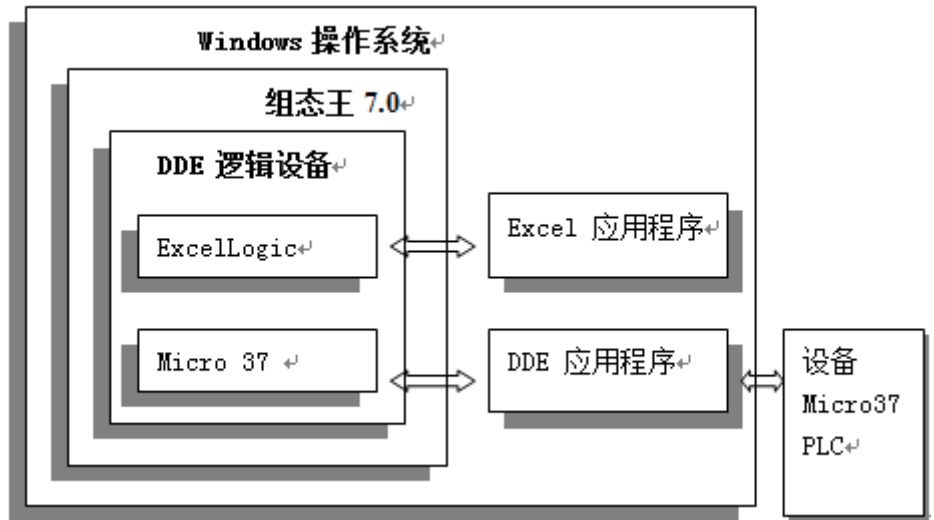
6.1.2 组态王 7.0 逻辑设备的分类

组态王 7.0 设备管理中的逻辑设备分为 DDE 设备、板卡类设备（即总线型设备）、串口类设备、人机界面卡、网络模块，工程人员根据自己的实际情况通过组态王 7.0 的设备管理功能来配置定义这些逻辑设备，下面分别介绍这五种逻辑设备。

6.1.2.1 DDE 设备

DDE 设备是指与组态王 7.0 进行 DDE 数据交换的 Windows 独立应用程序，因此，DDE 设备通常就代表了一个 Windows 独立应用程序，该独立应用程序的扩展名通常为 .EXE 文件，组态王 7.0 与 DDE 设备之间通过 DDE 协议交换数据，如：EXCEL 是 Windows 的独立应用程序，当 EXCEL 与组态王 7.0 交换数据时，就是采用 DDE 的通讯方式进行；

又比如，北京亚控公司开发的莫迪康 MICRO37 的 PLC 服务程序也是一个独立的 Windows 应用程序，此程序用于组态王 7.0 与莫迪康 Micro37PLC 之间进行数据交换，则可以给服务程序定义一个逻辑名称作为组态王 7.0 的 DDE 设备，组态王 7.0 与 DDE 设备之间的关系如图 6.5 所示：



6.5 组态王 7.0 与 DDE 设备之间的关系

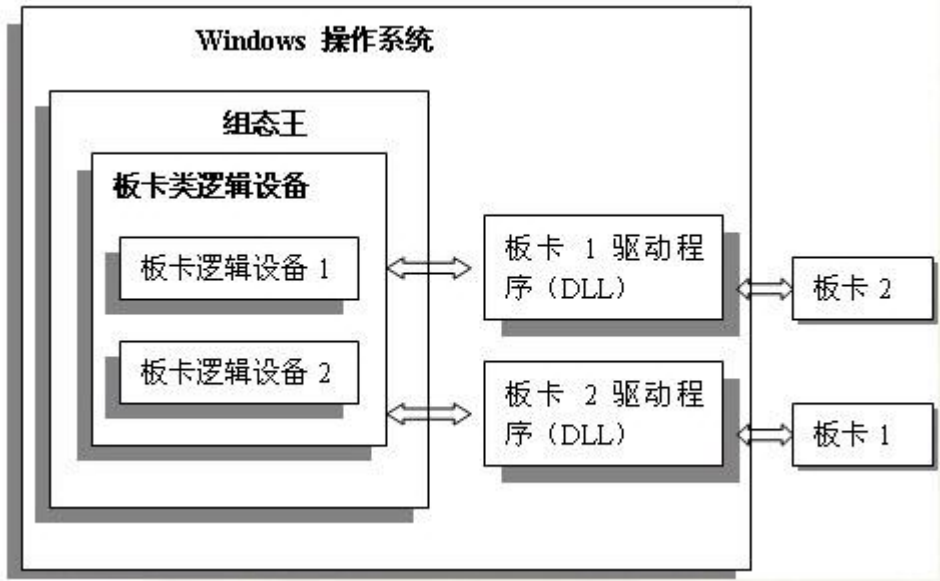
通过此结构图，可以进一步理解 DDE 设备的含义，显然，组态王 7.0、Excel、Micro37 都是独立的 Windows 应用程序，而且都要处于运行状态，再通过给 Excel、Micro37 DDE 分别指定一个逻辑名称，则组态王 7.0 通过 DDE 设备就可以和相应的应用程序进行数据交换。

具体 DDE 的通讯方法及操作等请参见第 18 章 组态王 7.0 与其它应用程序的动态数据交换（DDE）。

6.1.2.2 板卡类设备

板卡类逻辑设备实际上是组态王 7.0 内嵌的板卡驱动程序的逻辑名称，内嵌的板卡驱动

程序不是一个独立的 Windows 应用程序，而是以 DLL 形式供组态王 7.0 调用，这种内嵌的板卡驱动程序对应着实际插入计算机总线扩展槽中的 I/O 设备，因此，一个板卡逻辑设备也就代表了一个实际插入计算机总线扩展槽中的 I/O 板卡。组态王 7.0 与板卡类逻辑设备之间的关系如图 6.6 所示：



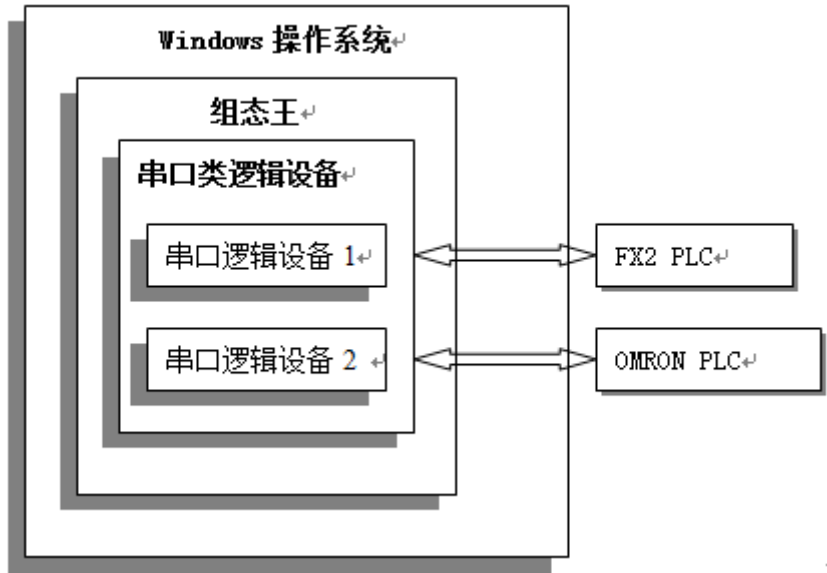
6.6 组态王 7.0 与板卡设备之间的关系

显然，组态王 7.0 根据工程人员指定的板卡逻辑设备自动调用相应内嵌的板卡驱动程序，因此对工程人员来说只需要在逻辑设备中定义板卡逻辑设备，其它的事情就由组态王 7.0 自动完成。

6.1.2.3 串口类设备

串口类逻辑设备实际上是组态王 7.0 内嵌的串口驱动程序的逻辑名称，内嵌的串口驱动程序不是一个独立的 Windows 应用程序，而是以 DLL 形式供组态王 7.0 调用，这种内嵌

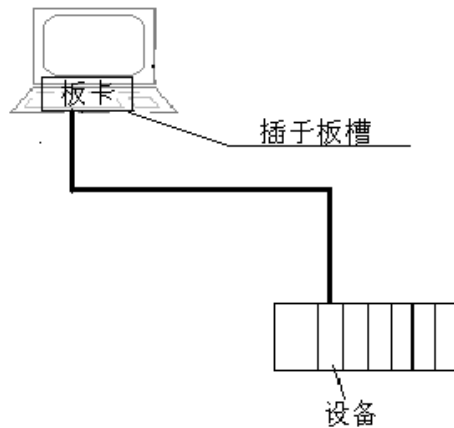
的串口驱动程序对应着实际与计算机串口相连的 I/O 设备，因此，一个串口逻辑设备也就代表了一个实际与计算机串口相连的 I/O 设备。组态王 7.0 与串口类逻辑设备之间的关系如图 6.7 所示：



6.7 组态王7.0与串口设备之间的关系

6.1.2.4 人机界面卡

人机界面卡又可称为高速通讯卡，它既不同于板卡，也不同于串口通讯，它往往由硬件厂商提供，如西门子公司的S7-300用的MPI卡、莫迪康公司的SA85卡。其工作原理和通讯示意图如图6.8所示：

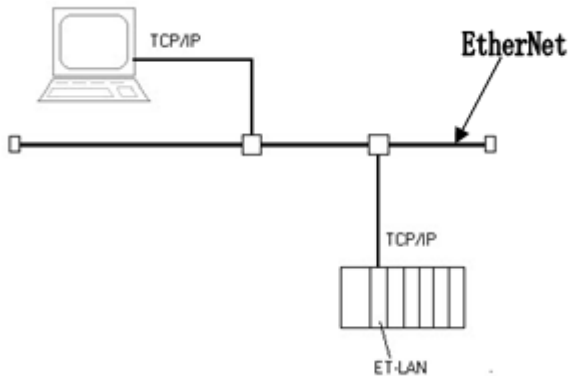


6.8 组态王 7.0 与人机界面卡设备之间的关系

通过人机界面卡可以使设备与计算机进行高速通讯，这样不占用计算机本身所带 RS232 串口，因为这种人机界面卡一般插在计算机的 ISA 板槽上。

6.1.2.5 网络模块

组态王 7.0 利用以太网和 TCP/IP 协议可以与专用的网络通讯模块进行连接，例如选用松下 ET-LAN 网络通讯单元通过以太网与上位机相连，该单元和其他计算机上的组态王 7.0 运行程序使用 TCP/IP 协议，连接示意图如图 6.9 所示：



6.9 组态王 7.0 与网络模块设备之间的关系

关于 OPC 设备的概念及通讯请参见“第二十章 OPC 设备”。

6.1.3 如何定义 IO 设备

在了解了组态王 7.0 逻辑设备的概念后,工程人员可以轻松的在组态王 7.0 中定义所需的设备了。进行 I/O 设备的配置时将弹出相应的配置向导页,使用这些配置向导页可以方便快捷地添加、配置、修改硬件设备。组态王 7.0 提供大量不同类型的驱动程序,工程人员根据自己实际安装的 I/O 设备选择相应的驱动程序即可。

6.1.3.1 如何定义 DDE 设备

工程人员根据设备配置向导就可以完成 DDE 设备的配置,操作步骤如下:

- 1、在工程浏览器的目录显示区,用鼠标左键单击大纲项设备下的成员 DDE,则在目录内容显示区出现“新建”图标,如图 6.10 所示:

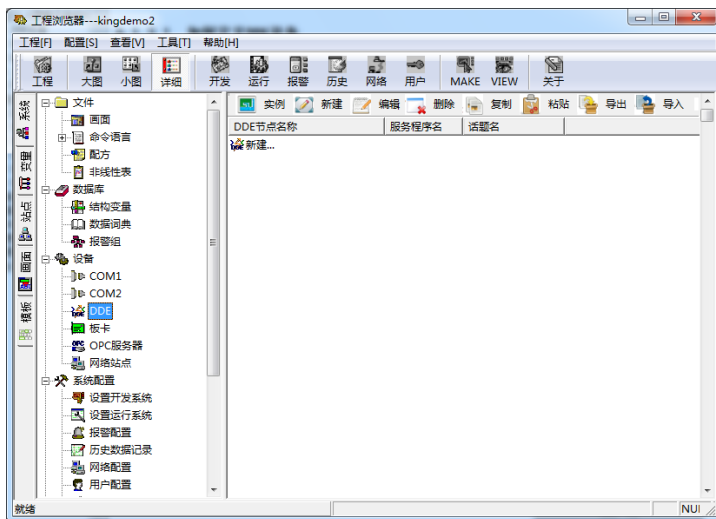


图6.10 DDE 设备配置

选中“新建”图标后用左键双击，弹出“设备配置向导”对话框；或者用右键单击，则弹出浮动式菜单，选择菜单命令“新建 DDE 节点”，也弹出“设备配置向导”对话框，如图 6.11 所示：

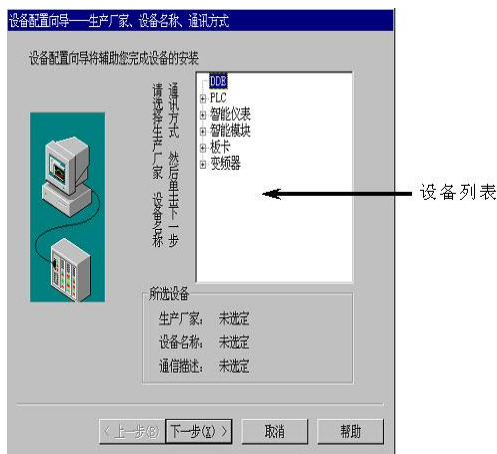


图6.11 设备配置向导

工程人员从树形设备列表区中选择 DDE 节点。

- 2、单击“下一步”按钮，则弹出“设备配置向导——逻辑名称”对话框，如图 6.12 所示：



图6.12 填入设备逻辑名称

在对话框的编辑框中为 DDE 设备指定一个逻辑名称。如“ExcelToView”。单击“上一步”按钮，则可返回上一个对话框。

- 3、单击“下一步”按钮，则弹出配置向导对话框，如图 6.13 所示：



图 6.13 填入 DDE 服务器配置信息

工程人员要为 DDE 设备指定 DDE 服务程序名、话题名、数据交换方式。若要修改 DDE 设备的逻辑名称，单击“上一步”按钮，则可返回上一个对话框。对话框中各项的含义为：

- **服务程序名：**是与“组态王 7.0”交换数据的 DDE 服务程序名称，一般是 I/O 服务程序，或者是 Windows 应用程序。本例中是 Excel.exe。
- **话题名：**是本程序和服务程序进行 DDE 连接的话题名 (Topic)。如图为 Excel 程序的工作表名 sheet1。
- **数据交换形式：**是指 DDE 会话的两种方式，“高速块交换”是本公司开发的通信程序采用的方式，它的交换速度快；如果工程人员是按照标准的 Windows DDE 交换协议开发自己的 DDE 服务程序，或者是在“组态王”和一般的 Windows 应用程序之间交换数据，则应选择“标准的 WINDOWS 项目交换”选项。

4、单击“下一步”按钮，则弹出设备配置向导“信息总结”对话框，如图 6.14 所示：



图 6.14 DDE 设备配置信息汇总

此向导页显示已配置的 DDE 设备的全部设备信息，供工程人员查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，则工程浏览器设备节点下的 DDE 节点处显示已添加的 DDE 设备。

5、DDE 设备配置完后，分别启动 DDE 服务程序和组态王 7.0 的 Touchview 运行环境。

6.1.3.2 如何定义板卡类设备

工程人员根据设备配置向导就可以完成板卡设备的配置，操作步骤如下：

在工程浏览器的目录显示区，用鼠标左键单击大纲项**设备**下的成员**板卡**，则在目录内容显示区出现“新建”图标，如图 6.15 所示：

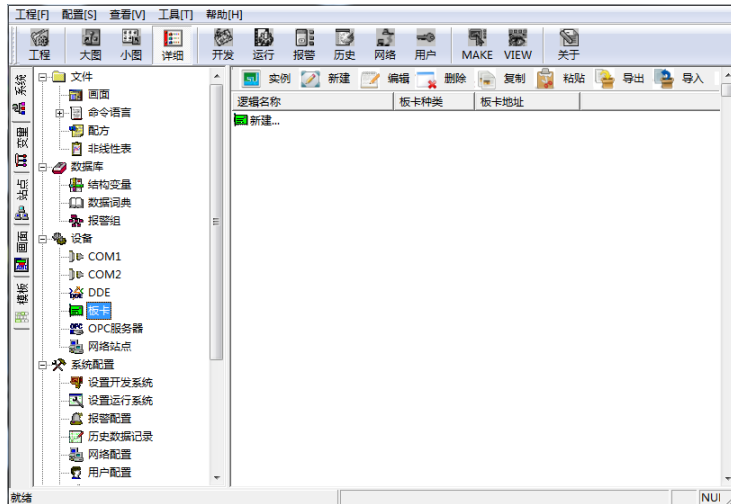


图6.15 板卡配置

下面以研华PCL_724（24通道数字量输出/输入，采用8255控制方式）介绍板卡设备的配置。

1. 选中“新建”图标后用左键双击，弹出“设备配置向导”列表对话框；或者用右键单击，则弹出浮动式菜单，选择菜单命令“新建板卡”，也弹出“设备配置向导”列表对话框，如图 6.16 所示：



图 6.16 板卡配置向导

从树形设备列表区中选择板卡节点。然后选择要配置板卡设备的生产厂家、设备名称。如“板卡/研华/PCL724”。

2. 单击“下一步”按钮，则弹出如下设备配置向导——“设备名称”，如图 6.17 所示：



图 6.17 填入板卡逻辑名称

工程人员给要配置的板卡设备指定一个逻辑名称。单击“上一步”按钮，则可返回上一个对话框。

3. 继续单击“下一步”按钮，则弹出如下设备配置向导——“板卡地址”如图 6.18 所示：



图 6.18 填入板卡配置信息


工程人员要为板卡设备指定板卡地址、初始化字（初始化字以 port, dat, port, dat……形式输入，其中 port 为芯片初始化地址偏移量，dat 为初始化字）、AD 转换器的输入方式（单端或双端）。

⚠ 注意：

初始化字是针对某些需要特殊控制的板卡提供的，如有 8255 芯片的板卡，用户需要通过初始化字来确定每个通道的输入、输出状态。另外，有一些带有计数器的板卡也需要相应的初始化字配置。

下表为 8255 芯片的初始化表：

↕	D4↕	D3↕	D1↕	D0↕
Cw↕	Pa0-pa7↕	Pc4-pc7↕	Pb0-pb7↕	Pc0-pc3↕
80H↕	output↕	Output↕	Output↕	Output↕
81H↕	Output↕	Output↕	Output↕	Input↕
82H↕	Output↕	Output↕	Input↕	Output↕
83H↕	Output↕	Output↕	Input↕	Input↕
88H↕	Output↕	Input↕	Output↕	Output↕
89H↕	Output↕	Input↕	Output↕	Input↕
8AH↕	Output↕	Input↕	Input↕	Output↕
8BH↕	Output↕	Input↕	Input↕	Input↕
90H↕	Input↕	Output↕	Output↕	Output↕
91H↕	Input↕	Output↕	Output↕	Input↕
92H↕	Input↕	Output↕	Input↕	Output↕
93H↕	Input↕	Output↕	Input↕	Input↕
98H↕	Input↕	Input↕	Output↕	Output↕
99H↕	Input↕	Input↕	Output↕	Input↕
9AH↕	Input↕	Input↕	Input↕	Output↕
9BH↕	Input↕	Input↕	Input↕	Input↕

 注意:

单端，双端是针对于模拟信号的输入，根据模拟信号放大器的不同选用不同的输入方式，详见各硬件厂商板卡说明书。

若要修改板卡设备的逻辑名称，单击“上一步”按钮，则可返回上一个对话框。

4. 继续单击“下一步”按钮，则弹出如下设备配置向导——“信息总结”对话框，汇总当前定义的设备的的全部信息，如下图6.19所示：



图6.19 板卡配置信息汇总

此向导页显示已配置的板卡设备的设备信息，供工程人员查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，则工程浏览器设备节点下的板卡节点处显示已添加的板卡设备。



注意：

关于板卡各寄存器的说明见附录 C

6.1.3.3 如何定义串口类设备以及设置串口参数

6.1.3.3.1 如何定义串口类设备

工程人员根据设备配置向导就可以完成串口设备的配置，组态王 7.0 最多支持 128 个串口。操作步骤如下：

1. 在工程浏览器的目录显示区，用鼠标左键单击大纲项设备下的成员 COM1 或 COM2，则在目录内容显示区出现“新建”图标，如下图 6.20 所示：

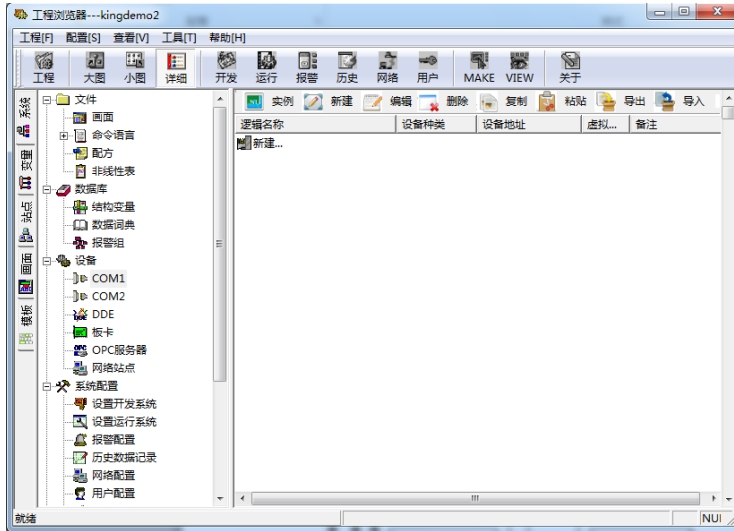


图6.20 新建串口设备

选中“新建”图标后用左键双击，弹出“设备配置向导”对话框；或者用右键单击，则弹出浮动式菜单，选择菜单命令“新建逻辑设备”，也弹出“设备配置向导”对话框，如下图6.21所示：



图6.21 串口配置向导

工程人员从树形设备列表区中可选择 PLC、智能仪表、智能模块、板卡、变频器等节点中的一个。然后选择要配置串口设备的生产厂家、设备名称、通讯方式；PLC、智能仪表、智能模块、变频器等设备通常与计算机 的串口相连进行数据通讯。

- 单击“下一步”按钮，则弹出如下设备配置向导——“设备名称”对话框，如下图 6.22 所示：



图6.22 设备逻辑名称

工程人员给要配置的串口设备指定一个逻辑名称。单击“上一步”按钮，则可返回上一个对话框。

3. 继续单击“下一步”按钮，则弹出如下设备配置向导——“选择串口号”对话框，如下图6.23所示：



图6.23 选择设备连接的串口

工程人员为配置的串行设备指定与计算机相连的串口号，该下拉式串口列表框共有 128 个串口号供工程人员选择。

4. 继续单击“下一步”按钮，则弹出如下设备配置向导——“设备地址设置”对话框，如下图 6.24 所示：



图6.24 填入PLC 设备地址

工程人员要为串口设备指定设备地址，该地址应该对应实际的设备定义的地址，具体请参见组态王 7.0 设备帮助。若要修改串口设备的逻辑名称，单击“上一步”按钮，则可返回上一个对话框。

5. 继续单击“下一步”按钮，则弹出如下设备配置向导——“通信参数”对话框，如下图 6.25 所示：



图6.25 填入通信参数

此向导页配置一些关于设备在发生通信故障时，系统尝试恢复通信的策略参数：

- ◇ **尝试恢复时间：**在组态王 7.0 运行期间，如果有一台设备如 PLC1 发生故障，则组态王 7.0 能够自动诊断并停止采集与该设备相关的数据，但会每隔一段时间尝试恢复与该设备的通讯，如图所示尝试时间间隔为 30 秒。
- ◇ **最长恢复时间：**若组态王 7.0 在一段时间之内一直不能恢复与 PLC1 的通讯，则不再尝试恢复与 PLC1 通讯，这一时间就是指最长恢复时间。如果将此参数设为 0，则表示最长恢复时间参数设置无效，也就是说，系统对通讯失败的设备将一直进行尝试恢复，不再有时间上的限制。
- ◇ **使用动态优化：**组态王 7.0 对全部通讯过程采取动态管理的办法，只有在数据被上位机需要时才被采集，这部分变量称之为活动变量。活动变量包括：
 - 当前显示画面上正在使用变量。
 - 历史数据库正在使用的变量。
 - 报警记录正在使用的变量。
 - 命令语言中（应用程序命令语言、事件命令语言、数据变化命令语言、热

键命令语言、当前显示画面用的画面命令语言)正在使用的变量。

同时,组态王 7.0 对于那些暂时不需要更新的数据则不进行通讯。这种方法可以大大缓解串口通讯速率慢的矛盾。有利于提高系统的效率和性能。



例如:

工程人员为一台 OMRON PLC 定义了 1000 多个 I/O 变量,但在某一时刻,显示画面上的动态连接、历史记录、报警、命令语言等,可能只使用 1000 个 I/O 变量中的一部分,在这种情况下组态王 7.0 通过动态优化将只采集这些活动变量。当系统中 I/O 变量数目明显增加时,这种通讯方式可以保证数据采集周期不会有太大变化。如果对与通讯参数还需要修改,单击“上一步”按钮,则可返回上一个对话框进行修改,如果不需要修改,单击“下一步”。

6. 继续单击“下一步”按钮,则弹出如下设备配置向导——“信息总结”对话框,如下图所示:



图6.26 配置信息汇总

此向导页显示已配置的串口设备的设备信息，供工程人员查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，则工程浏览器设备节点处显示已添加的串口设备。

6.1.3.3.2 如何设置串口参数

对于不同的串口设备，其串口通讯的参数是不一样的，如波特率、数据位、校验位等。所以在定义完设备之后，还需要对计算机通讯时串口的参数进行设置。如上节中定义设备时，选择了COM1口，则在工程浏览器的目录显示区，选择“设备”，双击“COM1”图标，弹出“设置串口——COM1”对话框，如下图6.27所示。



图6.27 设置串口参数

在“通讯参数”栏中，选择设备对应的波特率、数据位、校验类型、停止位等，这些参数的选择可以参考组态王7.0的相关设备帮助或按照设备中通讯参数的配置。“通讯超时”为默认值，除非特殊说明，一般不需要修改。“通讯方式”是指计算机一侧串口的通讯方式，是RS232或RS485，一般计算机一侧都为RS232，按实际情况选择相应的类型即可。

6.1.3.4 如何定义 OPC 服务器设备

请见第二十章 OPC 设备。

6.1.3.5 如何定义带网络模块的设备

有些设备如 PLC 的通讯模块为网络模块，支持 TCP/IP 协议，通过该模块与上位机进行数据交换。例如，定义 OMRON PLC 的 CS 系列以太网通讯设备。



图 6.28 选择以太网设备

- 在组态王 7.0 工程浏览器中双击“设备/新建...”图标，弹出设备配置向导，依次选择节点“PLC/欧姆龙/CS 系列/以太网”，如图 6.28 所示。
- 单击“下一步”，弹出设备配置向导——“设备逻辑名”对话框，如下图 6.29 所示。在编辑框中输入设备在组态王 7.0 中的逻辑名称，如“OMRON_PLC”。



图 6.29 设备逻辑名称

- 单击“上一步”修改设备的选择，单击“下一步”弹出设备配置向导——“设备地址设置”，如下图 6.30 所示。在地址编辑栏中输入 192.168.1.34:34.225(具体含义请参见组态王 7.0 驱动帮助)。

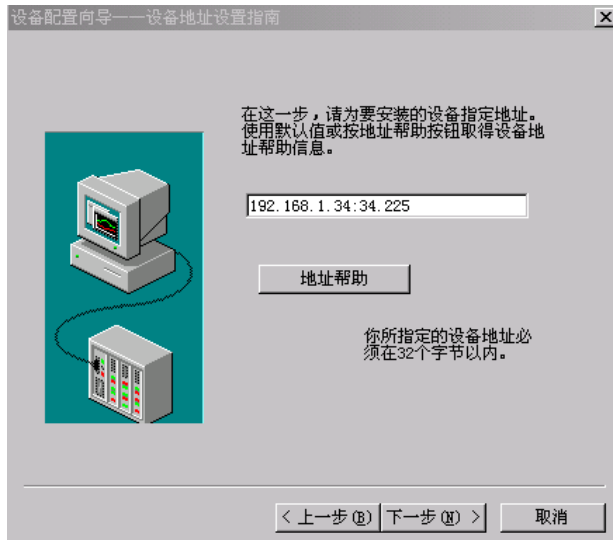


图 6.30 设备地址设置

- 单击“上一步”修改配置，单击“下一步”弹出设备配置向导——“通信参数”，如下图 6.31 所示，修改设备通讯出现故障时的尝试恢复策略。各选项的具体含义请参见 6.1.3.3.1 节的描述。

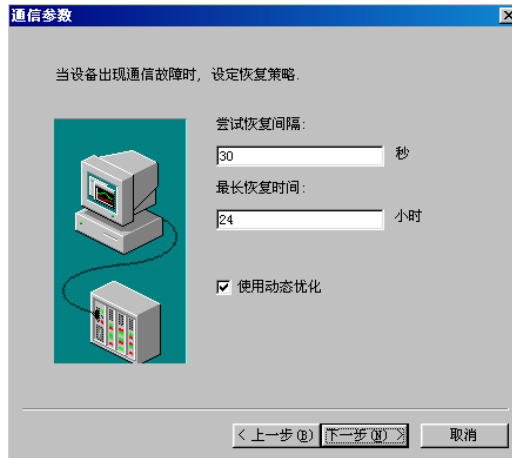


图 6.31 通信参数设置

- 单击“上一步”修改配置，单击“下一步”弹出设备配置向导——“信息总结”，如下图 6.32 所示。

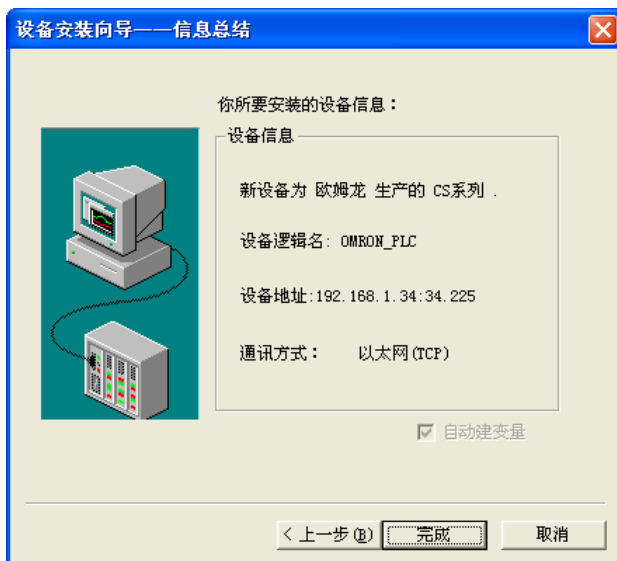


图 6.32 设备配置信息总结

- 单击“上一步”修改配置，单击“完成”完成设备配置。

6.1.3.6 如何配置组态王 7.0 作为网络设备

分布在控制系统中的组态王 7.0 之间可以通过网络进行通讯，访问实时数据。远程访问组态王 7.0 的实时数据有两种方式：其一是在客户端上定义服务器站点为一个网络站点设备，然后在客户端上定义变量与该网络站点上的变量连接，访问实时数据；第二种是使用组态王 7.0 的网络功能直接引用远程站点上的变量，而无需在客户端上定义变量。第二种方法请参见第二十一章 网络功能。这两种方法的特点为：

- ◇ 客户端均可以访问到服务器上的实时数据。
- ◇ 第一种方法需要在客户端上定义变量，如果需要访问的数据较多时，工作量较大，客户端系统的点数也会增加，但可以在本机上直接进行历史数据记录、产

生报警等。第二种方法无需在客户端上定义变量，直接引用服务器上的组态王 7.0 变量，系统的点数也不会额外增加。但历史数据的访问等只能从历史数据服务器上获得。这里主要讲述第一种方式的配置方法。

6.1.3.6.1 定义网络站点设备

该功能是使用在组态王 7.0 “NET VIEW” 方式下。在工程浏览器的目录显示区，选择大纲项“设备/网络站点”，在右侧的内容显示区显示“新建...”。如下图 6.33 所示：

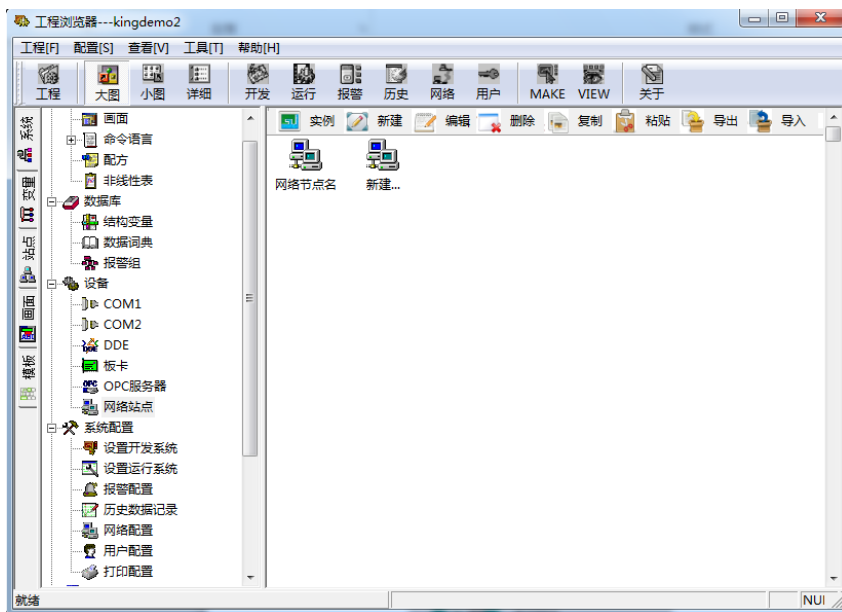


图 6.33 建立网络站点

双击“新建...”，弹出网络节点对话框，如下图 6.34 所示。



图 6.34 建立网络站点

在“机器名”文本框中输入远程站点的计算机名称或 IP 地址，如“数据采集站”。**此处建议**使用的机器名不要带“-”，如果带，单击确定后会变成 IP 地址（所以在使用机器名之前，请先判断一下 m_LocalName 是否包含“-”字符，如包含，请采用 IP 地址方式）。

如果远程站点有备份机，选择“本节点有备份机”选项，并在“备份机机器名”文本框中输入备份机的名称。这样，当远程站点出现故障切换到备份机时，本地站点也可以自动切换到备份机与备份机进行通讯，保证数据的完整性。输入完成后，单击“确定”按钮。这样一个网络站点设备就建立完成了。在工程浏览器“设备/网络站点”下会出现一个名为“数据采集站”的网络站点设备。

6.1.3.6.2 网络站点设备的使用

1、定义网络方式

建立完网络站点设备后，使用该设备之前，应对客户端和服务器端的网络功能进行一些配置。将两端均定义为“连网”模式。选择工程浏览器大纲项“系统配置/网络配置”，双击该项，弹出网络配置对话框，如下图 6.35 所示。

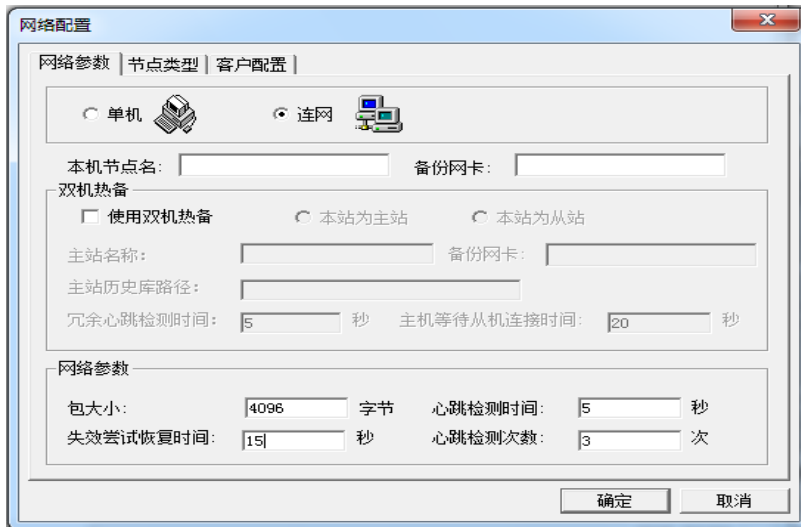


图 6.35 配置网络

选择“连网”选项，在“本机节点名”中输入本机的机器名或 IP 地址，如客户端为“客户端”。在“节点类型”属性页中，选择所有选项。对话框中的其它各项的定义和修改请参见“第二十一章 网络功能”。

2、定义变量：

定义变量的具体方法请参见第五章 变量的定义和管理。在变量的“连接设备”列表中选择网络站点设备，在“远程变量”编辑框中输入对应的远程变量的变量名。如图所示。如远程变量为“原料罐液位”。如下图 6.36 所示：



图 6.36 定义变量

这样可以将远程站点上的组态王 7.0 实时数据采集到客户端上来。实现网络上组态王之间的互相通讯。

6.1.4 选择驱动

在上一节定义 I/O 设备的过程中，我们发现设备配置向导左侧有个“高级查找”按钮，用来帮助用户选择驱动类型，如图 6.37 所示。



图6.37 设备配置向导

点击“高级查找”按钮，弹出对话框，如图 6.38 所示，左边为设备类型、厂家信息的树，右边为对应的驱动信息。对话框可以最大化，自由拖拉任意尺寸。

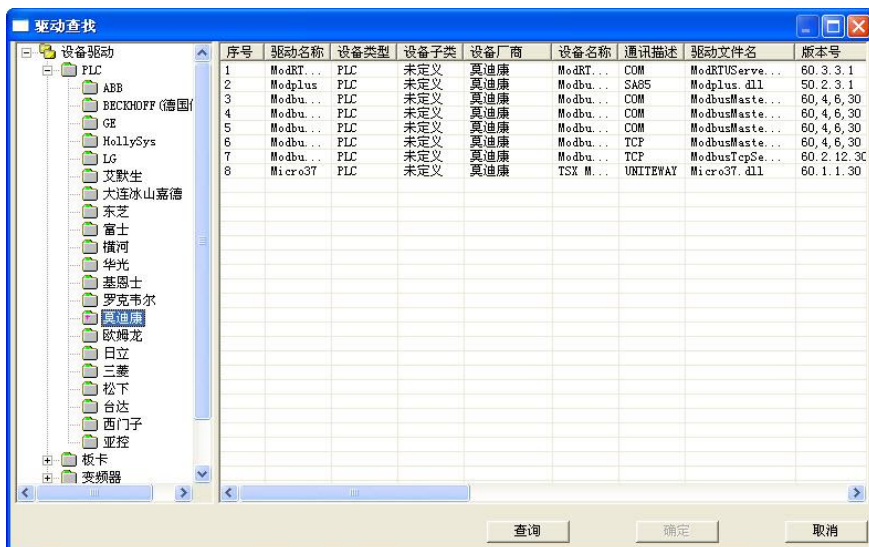


图 6.38 设备配置向导

选中左边树上的文件夹,可以在右边表格中列出相应的设备驱动信息,包含设备类型、子类型、生产厂商、设备名称等,可以从中选择驱动,这与从向导页树上选择的驱动,是可以互相定位的。

6.1.5 驱动查找

单击图 6.38 的“查询”按钮,弹出驱动查找画面,所列驱动信息为当前所选择文件夹及其子文件夹下的设备驱动信息,如图 6.39,单击树目录的右键,亦可以弹出该文件夹下驱动查询界面。

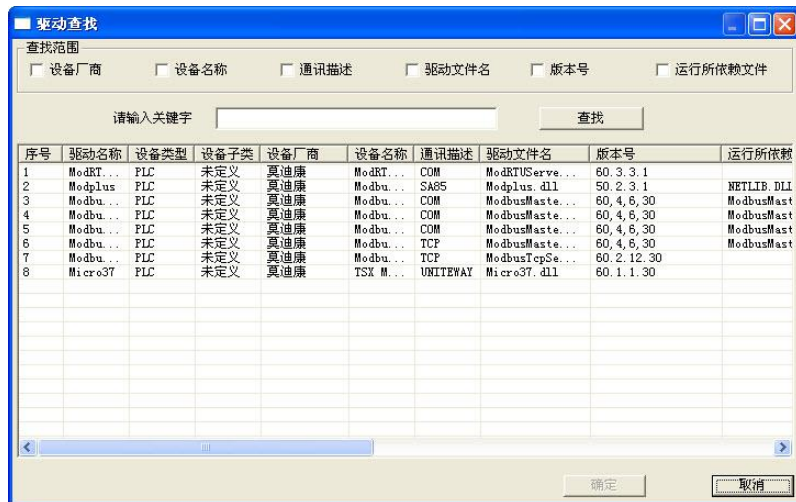



图 6.39 设备配置向导

输入查询的关键词后，可以按“设备厂商”、“设备名称”、“通讯描述”、“驱动文件名”、“版本号”、“运行时所依赖的文件”等查询。在列出的驱动中，只有选中一条以后，“确定”按钮才有效。

6.1.6 显示本地信息

对于图 6.38、图 6.39，其中的列表框，鼠标选择一条或多条驱动，点击右键可以“显示本地信息”，即从本地计算机上读取驱动信息，填充到列表上的“本地信息”一列。

 注意：

1、列表中除“本地信息”一列是程序读取本地信息外，其余的是读取的列表文件中的内容，是要求程序员开发时填写的。

2、用鼠标选择的地方，也可以用键盘选择，按上、下键可以上下选择，对于树，按左、右键可以折叠、展开树。

6.2 组态王 7.0 提供的模拟设备—仿真 PLC

程序在实际运行中是通过 I/O 设备和下位机交换数据的，当程序在调试时，可以使用仿真 I/O 设备模拟下位机向画面程序提供数据，为画面程序的调试提供方便。

组态王 7.0 提供一个仿真 PLC 设备，用来模拟实际设备向程序提供数据，供用户调试。

6.2.1 仿真 PLC 的定义

在使用仿真 PLC 设备前，首先要定义它，实际 PLC 设备都是通过计算机的串口向组态王 7.0 提供数据，所以仿真 PLC 设备也是模拟安装到串口 COM 上，定义过程和步骤为：

1. 在组态王 7.0 的工程浏览器中，从左边的工程目录显示区中选择大纲项**设备**下的成员名 COM1 或 COM2，然后在右边的目录内容显示区中用左键双击“新建”图标，则弹出“设备配置向导”对话框如下图 6.40 所示：



图 6.40 设备配置向导

在 I/O 设备列表显示区中，选中 PLC 设备，单击符号“+”将该节点展开，再选中“亚控”，单击符号“+”将该节点展开，选中“仿真 PLC”设备，再单击符号“+”将该节点展开，选中“串行”。

- 单击“下一步”按钮，则弹出“设备配置向导——逻辑名称”对话框如下图 6.41 所示：



图 6.41 填入逻辑名称


在编辑框输入一个仿真 PLC 设备的逻辑名称，例如设定为“simu”。

3. 继续单击“下一步”按钮，则弹出“设备配置向导——选择串口号”对话框如下图 6.42 所示：



图 6.42 选择串口

在下拉式列表框中列出了 32 个串口设备（COM1–COM32）供用户选择，例如从下拉式列表框中选 COM2 串口。

 注意：

这里定义的串口是虚拟的，实际仿真 PLC 设备并不使用计算机的 COM 口，而且 COM 口也不需要配置。

4. 继续单击“下一步”按钮，则弹出“设备配置向导——设备地址设置指南”对话框如下图 6.43 所示：



图 6.43 设备地址设置

在编辑框中输入仿真 PLC 设备的地址。

5. 继续单击“下一步”按钮，则弹出“设备配置向导——通讯参数”对话框，如下图 6.44 所示：

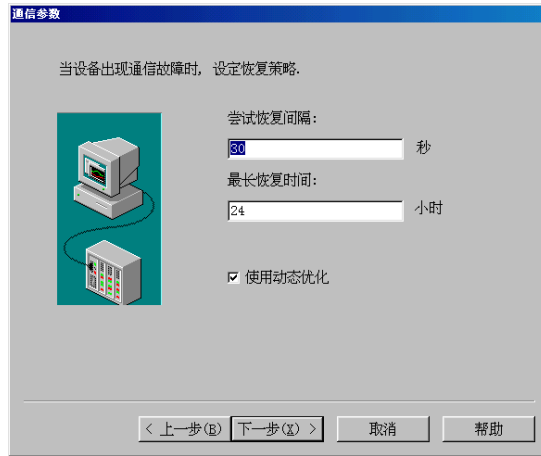


图 6.44 通讯参数设置

6. 继续单击“下一步”按钮，则弹出“设备配置向导——信息总结”对话框如下图 6.45 所示：



图 6.45 设备配置信息总结

单击“完成”按钮，则设备安装完毕，单击“上一步”，可返回上一次操作进行修改。

仿真 PLC 设备安装完毕后，可在工程浏览器进行查看，选择大纲项设备下的成员名 COM2，则在右边的目录内容显示区可以已安装的设备，如下图 6.46 所示：

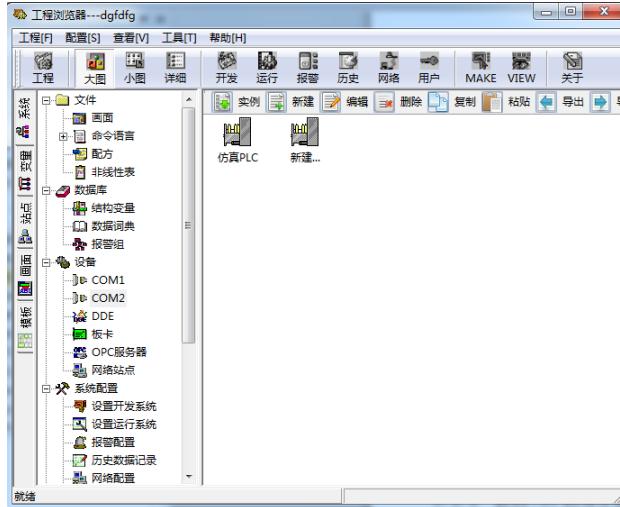


图 6.46 定义的仿真 PLC 设备

6.2.2 仿真 PLC 的寄存器

仿真 PLC 提供六种类型的内部寄存器变量 INCREA、DECREA、RADOM、STATIC、STRING、CommErr，其中 INCREA、DECREA、RADOM、STATIC 寄存器变量的编号从 1-1000，变量的数据类型均为整型（即 Short），STRING 寄存器变量的编号从 1-2，对这六类寄存器变量分别介绍如下：

6.2.2.1 自动加 1 寄存器 INCREA

该寄存器变量的最大变化范围是 0-1000，寄存器变量的编号原则是在寄存器名后加上整数值，此整数值同时表示该寄存器变量的递增变化范围，例如，INCREA100 表示该寄存器变量从 0 开始自动加 1，其变化范围是 0 到 100，关于寄存器变量的编号及变化范围

如下表所示：

寄存器变量	变化范围
INCREA1	0-1
INCREA2	0-2
INCREA3	0-3
⋮	⋮
INCREA1000	0-1000

6.2.2.2 自动减 1 寄存器 DECREA

该寄存器变量的最大变化范围是 0-1000，寄存器变量的编号原则是在寄存器名后加上整数值，此整数值同时表示该寄存器变量的递减变化范围，例如，DECREA100 表示该寄存器变量从 100 开始自动减 1，其变化范围是 0 到 100，关于寄存器变量的编号及变化范围如下表所示：

寄存器变量	变化范围
DECREA1	0-1
DECREA2	0-2
DECREA3	0-3
⋮	⋮
DECREA1000	0-1000

6.2.2.3 静态寄存器 STATIC

该寄存器变量是一个静态变量，可保存用户下发的数据，当用户写入数据后就保存下来，并可供用户读出，直到用户再一次写入新的数据，此寄存器变量的编号原则是在寄存器名后加上整数值，此整数值同时表示该寄存器变量能存储的最大数据范围，例如，STATIC100 表示该寄存器变量能接收 0-100 中的任意一个整数，关于寄存器变量的编号及接收数据范围如下表所示：

寄存器变量	接收数据范围
STATIC1	0-1
STATIC2	0-2
STATIC3	0-3
⋮	⋮
STATIC1000	0-1000

6.2.2.4 随机寄存器 RADOM

该寄存器变量的值是一个随机值，可供用户读出，此变量是一个只读型，用户写入的数据无效，此寄存器变量的编号原则是在寄存器名后加上整数值，此整数值同时表示该寄存器变量产生数据的最大范围，例如，RADOM100 表示随机值的范围是 0-100，关于寄存器变量的编号及随机值的范围如下表所示：

寄存器变量	随机值的范围
RADOM1	0-1

RADOM2	0-2
RADOM3	0-3
⋮	⋮
RADOM1000	0-1000

6.2.2.5 STRING 寄存器

该寄存器变量的值是一字符串，只读类型。该寄存器的编号范围为 1—2。字符串值形式为“hello: 数字—数字”，“数字”值自动加 1。

6.2.2.6 CommErr 寄存器

该寄存器变量为可读写的离散变量，用来表示组态王 7.0 与设备之间的通讯状态。CommErr=0 表示通讯正常；CommErr=1 表示通讯故障。用户通过控制 CommErr 寄存器状态来控制运行系统与仿真 PLC 通讯，将 CommErr 寄存器置为打开状态时中断通讯，置为关闭状态后恢复运行系统与仿真 PLC 之间的通讯。

6.2.3 仿真 PLC 使用举例

以对常量寄存器 STATIC100 读写操作为例来说明如何使用仿真 PLC 设备。

6.2.3.1 仿真 PLC 的定义

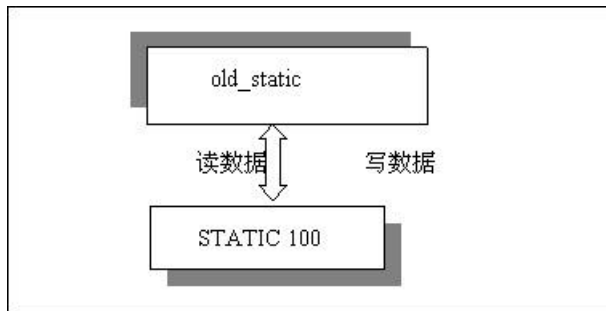
仿真 PLC 的定义过程详见 6.2.1 节，假定定义后的设备信息如下图 6.47 所示：



图 6.47 设备信息

6.2.3.2 定义 I/O 变量

定义一个 I/O 型变量 `old_static`，用于读写常量寄存器 `STATIC100` 中的数据，示意图如下所示：



定义变量 old_static 的过程如下：

在工程浏览器中，从左边的工程目录显示区中选择大纲项数据库下的成员数据词典，然后在右边的目录内容显示区中用左键双击“新建”图标，弹出“变量属性”对话框，如下图所示：



图 6.48 定义变量

在此对话框中，变量名定义为 old_static，变量类型为 I/O 实数，连接设备选择 simu，寄存器定为 STATIC100，寄存器的数据类型定为 INT，读写属性为读写（根据寄存器类型定义），其它的定义见对话框，单击“确定”按钮，则 old_static 变量定义结束。

 注意：

对于不同的外围设备，有不同的寄存器类型，每种寄存器类型又可分为只读，只写，读写三种属性，具体见附录。

6.2.3.3 制作画面

在工程浏览器中，单击菜单命令“工程\切换到 Make”，进入到组态王 7.0 开发系统，制作的画面如下图 6.49 所示。并对读数据和写数据的两个输出文本串“###”分别进行动画连接。

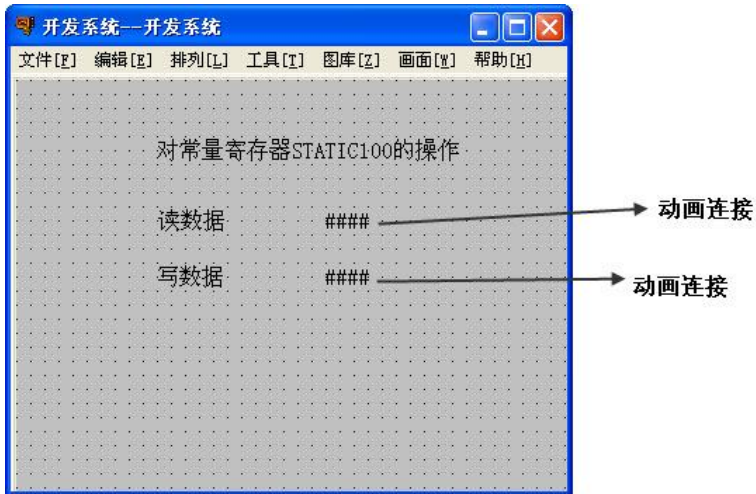
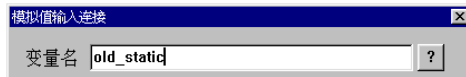


图 6.49 定义动画连接

其中写数据的输出文本串“###”要进行“模拟值输入”连接，连接的表达式是变量 old_static，如下所示：





读数据的输出文本串“###”要进行“模拟值输出”连接，连接的表达式是变量 old_static，方法同上。



6.2.3.4 运行画面程序

运行组态王 7.0 运行程序，打开画面，运行画面如下：



对常量寄存器 STATIC100 写入数据 80，则可看到读出的数据值也是 80。

6.3 组态王 7.0 提供的通讯的其它特殊功能

6.3.1 开发环境下的设备通讯测试

为保证用户对硬件的方便使用，在完成设备配置与连接后，用户在组态王 7.0 开发环境中即可以对硬件进行测试。对于测试的寄存器可以直接将其加入到变量列表中。当用户

选择某个设备后，单击鼠标右键弹出浮动式菜单，除 DDE 外的设备均有菜单项“测试 设备名”。如定义亚控仿真 PLC 设备，在设备名称上单击右键，弹出快捷菜单，如下图 6.50 所示。

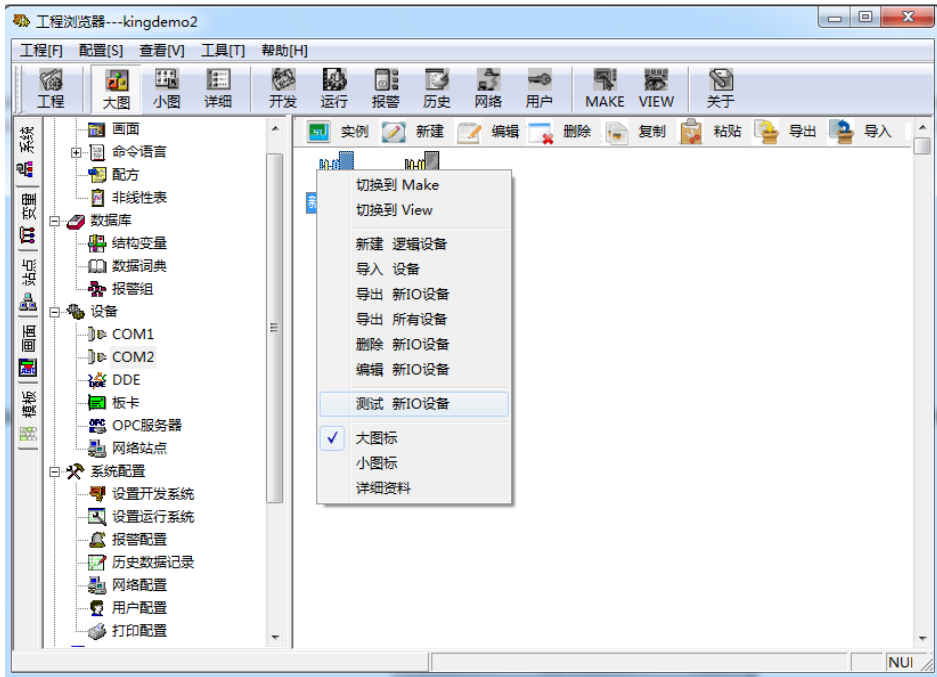


图6.50 硬件设备测试

使用设备测试时，点击“测试...”对于不同类型的硬件设备将弹出不同的对话框，如：对于串口通讯设备（如串口设备—亚控仿真 PLC）将弹出如下图 6.51 所示的对话框：



图 6.51 串口设备测试—通讯参数属性页

对话框共分为两个属性页：通讯参数、设备测试。

- 1、“通讯参数”属性页中主要定义设备连接的串口的参数、设备的定义等。这些参数的选择请参照本章相关章节或组态王 7.0 设备帮助。



图 6.52 串口设备测试—设备测试属性页

2、设备测试页如上图 6.52 所示。选择要进行通讯测试的设备的寄存器。

- **寄存器：**从寄存器列表中选择寄存器名称，并填写寄存器的序号（参见组态王 7.0 设备帮助）。如本例中的“INCREA”寄存器的“INCREA100”。然后从“数据类型”列表框中选择寄存器的数据类型。
- **添加：**单击该按钮，将定义的寄存器添加到“采集列表”中，等待采集。
- **删除：**如果不再需要测试某个采集列表中的寄存器，在采集列表中选择该寄存器，单击该按钮，将选择的寄存器从采集列表中删除。
- **读取/停止：**当没有进行通讯测试的时候，“读取”按钮可见，单击该按钮，对采集列表中定义的寄存器进行数据采集。同时，“停止”按钮变为可见。当需要停止通讯测试时，单击“停止”按钮，停止数据采集，同时“读取”按钮变为可见。

- **向寄存器赋值：**如果定义的寄存器是可读写的，则测试过程中，在“采集列表”中双击该寄存器的名称，弹出“数据输入”对话框，如下图 6.53 所示。在“输入数据”编辑框中输入数据，单击确定按钮，数据便被写入该寄存器。



图 6.53 数据输入对话框

- **加入变量：**将当前在采集列表中选择的寄存器定义一个变量添加到组态王 7.0 的数据词典中。单击该按钮，弹出变量名对话框，如图 6.54 所示：

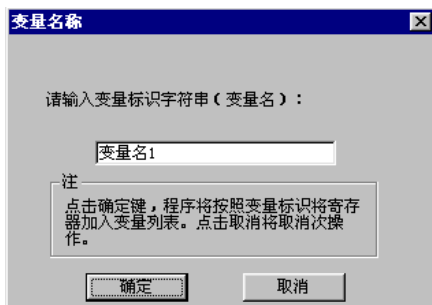


图 6.54 加入变量—输入变量名称

在编辑框中输入该寄存器所对应的变量名称，单击“确定”，该变量便加入到了组态王 7.0 的变量列表中，连接设备和寄存器为当前的设备和寄存器。

- **全部加入:** 将当前采集列表中的所有寄存器按照给定的第一个变量名称全部增加到组态王 7.0 的变量列表中, 各个变量的变量名称为定义的第一个变量名称后增加序号。如定义的第一个变量名称为“变量”, 则以后的变量依次为“变量 1”、“变量 2” …
- **采集列表:** 采集列表主要为显示定义的通讯测试的寄存器, 以及进行通讯时显示采集的数据、数据的时间戳、质量戳等。

开发环境下的设备通讯测试, 使用户很方便的就可以了解设备的通讯能力, 而不必先定义很多的变量和做一大堆的动画连接, 省去了很多工作, 而且也方便了变量的定义。



注意:

可以进行设备测试的有串口类设备、板卡类设备和 OPC 类设备。其它如 DDE、一些特殊通讯卡等都暂不支持该功能。如有疑问, 请与亚控技术支持联系。

6.3.2 如何在运行系统中判断和控制设备通讯状态

组态王 7.0 的驱动程序 (除 DDE 外) 为每一个设备都定义了 CommErr 寄存器, 该寄存器表征设备通讯的状态, 是故障状态还是正常。另外用户还可以通过修改该寄存器的值控制设备通讯的通断。

在使用该功能之前, 应该先为该寄存器定义一个 I/O 离散型变量, 变量为读写型。当该变量的值为 0 或被置为 0 时, 表示通讯正常或恢复通讯。当变量的值为 1 或被置为 1 时, 表示通讯出现故障或暂停通讯。

另外, 当某个设备通讯出现故障时, 画面上与故障设备相关联的 I/O 变量的数值输出显示都变为“???”号, 表示出现了通讯故障。当通讯恢复正常后, 该符号消失, 恢复为正常数据显示。

6.3.3 如何使用 MODEM 对设备进行远程拨号采集数据

组态王 7.0 支持对标准 232 串口通讯的设备用 MODEM 拨号进行访问的方式。

6.3.3.1 拨号设置

选择组态王 7.0 工程浏览器的大纲项“设备”，选择 MODEM 所连接的串口标识，如“COM2”，双击“COM2”，弹出串口设置对话框，如下图 6.55 所示。



图 6.55 串口设置

- **通讯参数：**设置进行串口通讯时，串口的通讯波特率、检验方式、数据位、停止位、设备与计算机的通讯方式等。该项设置用于任何一个串口通讯的设备。
- **Modem：**选择该项，则该串口为拨号访问设备方式。

主 Modem AT 控制字：设置与 PC 连接的 Modem 的控制字。系统启动时，先将该控制字写入主 Modem。**从 Modem AT 控制字：**设置与设备连接的 Modem 的控制字。

⚠ 注意:

建议用户使用专门的软件（如 Windows 的超级终端）按照 Modem 使用说明在系统运行前各个 Modem 进行控制字设置。所以用户可以不设置这两项。

设置完成后，单击“确定”按钮返回。则在组态王 7.0 设备列表中出现一个 Modem 设备，如下图 6.56 所示：

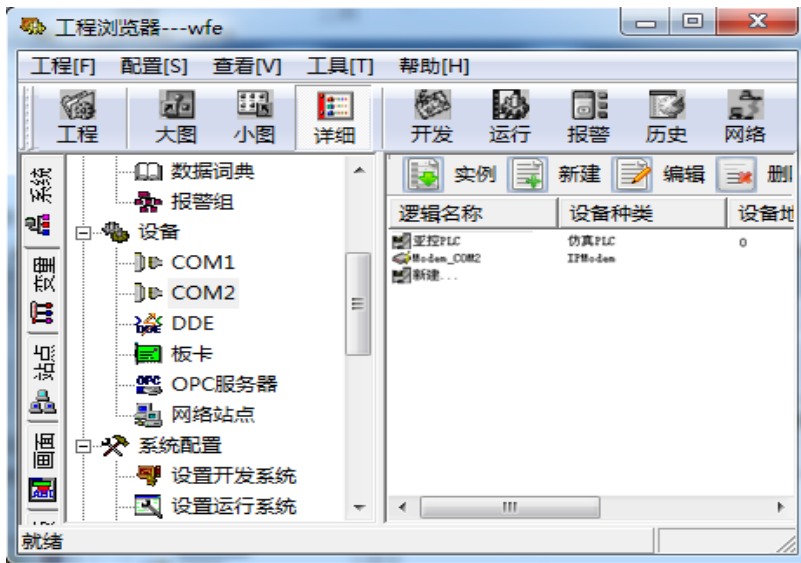


图6.56 定义Modem设备

6.3.3.2 设备拨号使用

6.3.3.2.1 建立 Modem 变量

对于 Modem 设备有四个寄存器：

- ◇ IPC：控制拨号通断寄存器。

- 数据类型：INT 型，可读写
 - 数值标识：1—开始拨号； 99—挂断
- ◇ IPS：拨通状态寄存器。
- 数据类型：INT 型，只读
 - 数值标识：0—未拨通；1—拨通
- ◇ IPN：用户所要拨的电话号码。
- 数据类型：字符串型，可读写
- ◇ CommErr：表示设备通讯状态，并可控制设备通讯状态。
- 数据类型：Bit，离散型，可读写
 - 数值标识：0—通讯正常，或允许通讯；1—通讯故障，或暂停设备通讯

在组态王 7.0 数据词典中新建变量(定义变量方法请参见第五章 变量定义和管理)，如：IPC1、IPS1、IPN1、Cerr1 分别对应以上的四个寄存器。设置变量 Cerr1 的初始值为“关”。

6.3.3.2.2 Modem 拨号的使用方法

在组态王 7.0 中定义数据采集设备，如“亚控 PLC”，其连接的串口为 COM2 口，定义相应的 I/O 数据采集的变量，然后定义设备“亚控 PLC”的“CommErr”寄存器变量，如“PLCCerr”。设置其初始值为“开”。

 注意：

因为系统启动时，会自动启动各个连接的设备，数据采集设备“亚控 PLC”没

有直接连接在 PC 上，所以一般会出现设备连接错误，所以在定义数据采集设备“亚控 PLC”的“CommErr”寄存器变量时，应设置其初始值为“开”，即暂停设备。

系统启动后，用户输入电话号码（变量 IPN1），然后进行拨号，当拨号拨通时，即 IPS1 的值为 1 时，设置 Modem 的 CommErr 寄存器（变量 Cerr1）的值为 1，即暂停 Modem 设备。然后设置数据采集设备的 CommErr 寄存器（变量 PLCerr）的值为 0，即恢复该设备，进行数据采集。当数据采集完成后，可以先暂停数据采集设备，然后恢复 Modem 设备的通讯，然后挂断。

对设备进行 Modem 拨号数据采集在很大程度上方便了用户进行远程调试、监控和数据采集。但用户须慎用：Modem 拨号只适用于简单的标准的 232 串口通讯设备，对于如 232C 链路、电流环等特殊 232 串口设备不支持。如果用户在使用时有问题，请与亚控公司的技术支持联系。

6.3.4 如何使用 GPRS 对设备进行远程通讯

随着移动推出 GPRS 无线数据传输以来，GPRS 的通信速度快、通信费用低、组网灵活等优点，越来越被广大客户看好。GPRS 数传终端，具有 TCP/IP 协议转换功能，不需要用户提供 TCP/IP 的支持。可适用于所有带串口的终端设备，通过 GPRS 网络平台实现数据信息的无线和透明传输，为不具备 TCP/IP 协议处理的终端设备提供了 GPRS 通信的能力。目前，GPRS 数传终端已被广泛的应用于环保、水文水利、油田、电力，工业控制等各个领域，在数据的远程传输和监控方面得到了很好的应用。

北京亚控公司开发的 GPRS 服务程序支持通过 GPRS 数传终端（以下简称 GPRS DTU）与

亚控开发过驱动程序的串口设备之间的通讯。

目前支持以下厂商的 GPRS DTU 设备（仅供参考，详细内容请参见组态王 7.0 提供的联机帮助—组态王 7.0 硬件帮助中的 GPRS 部分）：厦门桑荣，北京艺能，北京汉智通，唐山蓝迪，台湾尉普，北京爱立信，深圳宏电 UDP，深圳宏电 TCP，深圳倚天，北京欧特姆，福建实达 UDP，上海蓝峰，广州从兴，航天金软，博控 ERA5200 控制器，嘉复欣，福建实达 TCP，力创 LQ-8100，力创 LQ-8200，福建实达 TCP，蓝天顶峰（NETJET），江阴宏信，厦门宇能，上海英硕。

用户使用 GPRS 和组态王 7.0 通讯时的示意图，如下图 6.57 所示：

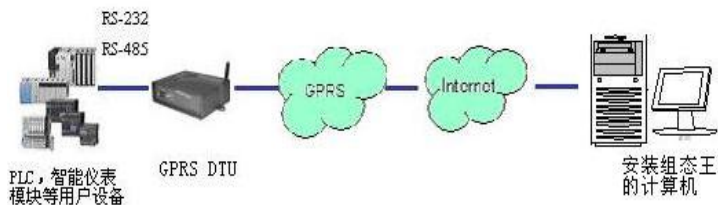


图 6.57 GPRS 和组态王 7.0 通讯时的示意图

6.3.4.1 定义 GPRS——虚拟串口设备

下面我们通过一个具体的例子来说明如何在组态王 7.0 中定义 GPRS DTU 设备。



例如：

用户使用莫迪康 PLC（MODBUS RTU 协议）作为现场的数据采集设备，组态王 7.0 通过桑容的 GPRS 设备和莫迪康 PLC 进行通讯。在组态王 7.0 中定义设备的步骤如下：

1. 选择串口设备，定义实际设备（即莫迪康 PLC），如下图 6.58 所示：



图 6.58 定义串口设备

2. 指定莫迪康 PLC 的逻辑名称，如下图 6.59 所示：

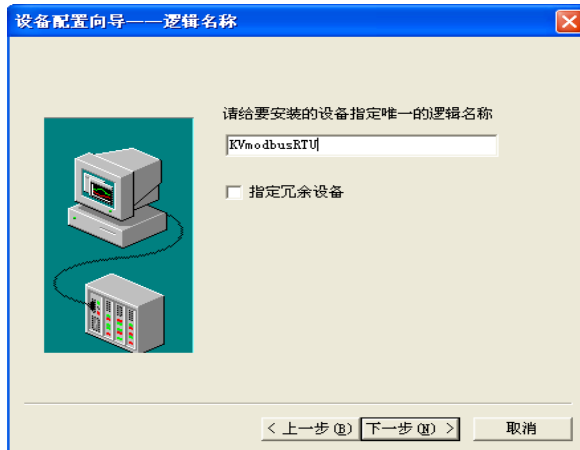


图 6.59 定义设备的逻辑名称

3. 给莫迪康 PLC 选择一个虚拟串口，如下图 6.60 所示：

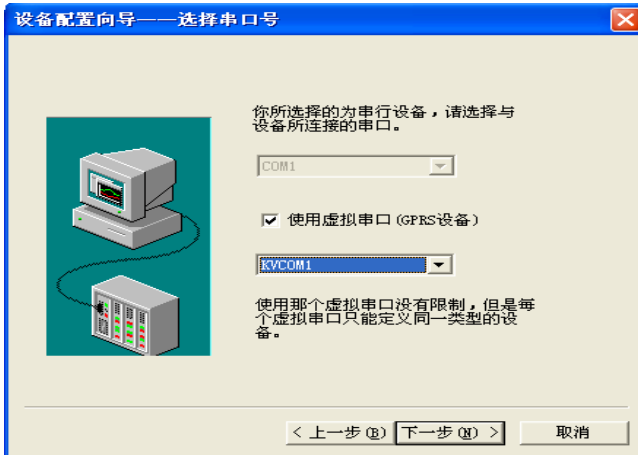


图 6.60 选择使用虚拟串口设备


- **选择虚拟串口 (GPRS 设备)：**选择该项，表示组态王 7.0 通过 GPRS 和串口设备通讯。如果不选用此选项表示组态王 7.0 直接和设备通讯。

4. 在虚拟串口上定义 GPRS 设备，如下图 6.61 所示：



图 6.61 定义虚拟串口设备

- **逻辑名称：**虚拟串口上定义的 GPRS 设备名称，用户自己定义
 - **设备选择：**从下拉菜单中选择组态王 7.0 支持的 GPRS 设备。例如选择桑荣设备，如上图所示。
 - **DTU 设备标识信息和设备端口号：**这两个参数要和 GPRS 硬件中的相应设置一致，组态王 7.0 通过此信息来找相应的 GPRS 设备。
-

 **注意：**

有关如何设置 GPRS 硬件参数的详细说明请参见组态王 7.0 提供的联机帮助——组态王 7.0 硬件帮助中的 GPRS 的使用帮助部分或相应的硬件使用说明书。

选择已定义的虚拟设备：当选定“选择已定义的虚拟设备”复选框时，在下拉框中将显示已经定义的虚拟设备。用户可以选择已经定义的虚拟设备（此项选择用于 1 个具有 485 接口的 GPRS DTU 下连接多个具有相同协议的数据采集终端设备的情况）。

5. **定义设备地址：**GPRS 下挂的实际设备的地址，如下图 6.62 所示：

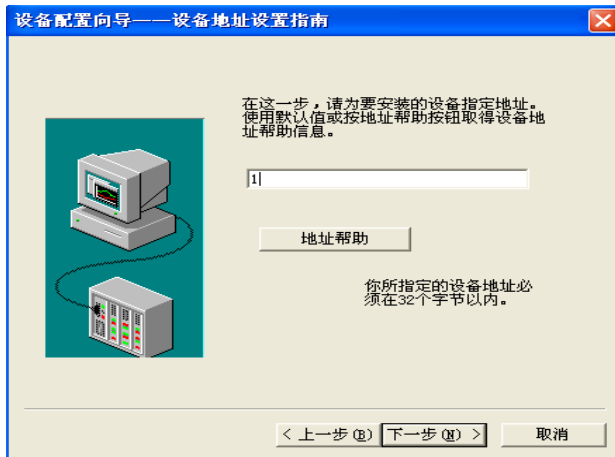


图 6.62 定义设备地址

6. 继续单击“下一步”按钮，则弹出“设备配置向导——通讯参数”对话框，如下图 6.63 所示：

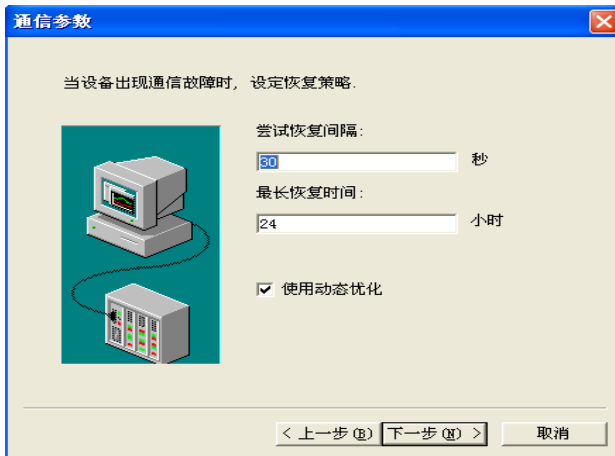


图 6.63 定义设备通讯参数

7. 继续单击“下一步”按钮，则弹出“设备配置向导——信息总结”对话框如

下图 6.64 所示：



图 6.64 定义设备信息总结


至此设备定义完毕，系统会生成两种设备的图标：

虚拟串口设备（即 GPRS DTU 设备）和 GPRS DTU 设备下挂的实际设备如下图 6.65

所示：

逻辑名称	设备种类	设备地址	虚拟设...	备注
虚拟设备	桑荣	1234567...		虚拟设备
KVMdbusRTU	KVMdbusRTUEx	1	虚拟设备	使用虚拟设备
新建...				

图 6.65 定义虚拟串口设备

 注意：

- 1、在一个虚拟串口上只能定义 1 个虚拟设备；
- 2、每个虚拟设备可以关联多个实际设备（设备的协议必须相同）；

- 3、与每个虚拟设备关联的实际设备地址不能重复；
 - 4、工程中所有虚拟设备或实际设备逻辑名称不能重复；
 - 5、不同虚拟设备的 DTU 标识信息不能重复；
 - 6、同一厂家的虚拟设备端口号应该一致；
-

6.3.4.2 定义 GPRS 变量

对于 GPRS 设备有两个寄存器：

寄存器名称	读写属性	数据类型	寄存器说明
V_S	只读	SHORT	虚拟设备（即 GPRS DTU 设备）的状态： 0 表示没有连接到 GPRS 网络。 1 表示已经连接到 GPRS 网络
V_C	读写	SHORT	虚拟设备控制寄存器。 0： 停止虚拟设备工作。 1： 恢复虚拟设备工作。

通讯时还要定义 GPRS 下面所连接的终端数据采集设备的变量，其变量的定义和不接 GPRS 设备时是一致的，在画面上分别连接 GPRS 和终端数据采集设备的变量。正常通讯时组态王 7.0 切换到运行，GPRS 的寄存器 V_S, V_C 两个寄存器的状态均为 1，表示 GPRS 设备已经连接到 GPRS 网络，并且工作正常。终端数据采集设备的数据是实时刷新的。如果用户不想和设备通讯可以将 V_C 的值置为 0，即停止虚拟设备工作，这样组态王 7.0 就

不和设备通讯了。如果用户在使用过程中有问题，请与亚控技术部联系。

6.4 用户如何自己开发驱动程序——驱动程序开发包

组态王 7.0 的驱动程序都是以动态连接库的形式提供的，在组态王 7.0 有设备通讯时，调用相应设备的动态连接库程序，该程序基于 COM 技术，实现了组态王 7.0 与设备通讯程序间的无缝连接。

组态王 7.0 已经提供了几百种设备的驱动程序。如果用户想要自己开发这样的驱动程序，组态王 7.0 将提供一个方便适用的驱动程序开发包，该开发包中有说明文档、程序示例等，可以帮助用户快速、有效的开发出自己的驱动程序。关于开发包的具体情况，请与亚控公司的技术支持或销售人员联系。

第七章 图形画面与动画连接

- ☞ 介绍动画连接的相关知识
- ☞ 介绍图形画面和动画连接的制作工具
- ☞ 教您如何制作动画

7.1 动画连接概述

7.1.1 连接概述

工程人员在组态王开发系统中制作的画面都是静态的,那么它们如何才能反映工业现场的状况呢?这就需要通过实时数据库,因为只有数据库中的变量才是与现场状况同步变化的。数据库变量的变化又如何导致画面的动画效果呢?通过“动画连接”——所谓“动画连接”就是建立画面的图素与数据库变量的对应关系。这样,工业现场的数据,比如温度、液面高度等,当它们发生变化时,通过 I/O 接口,将引起实时数据库中变量的变化,如果设计者曾经定义了一个画面图素——比如指针——与这个变量相关,我们将会看到指针在同步偏转。

动画连接的引入是设计人机接口的一次突破,它把工程人员从重复的图形编程中解放出来,为工程人员提供了标准的工业控制图形界面,并且由可编程的命令语言连接来增强

图形界面的功能。图形对象与变量之间有丰富的连接类型，给工程人员设计图形界面提供了极大的方便。“组态王”系统还为部分动画连接的图形对象设置了访问权限，这对于保障系统的安全具有重要的意义。

图形对象可以按动画连接的要求改变颜色、尺寸、位置、填充百分数等，一个图形对象又可以同时定义多个连接。把这些动画连接组合起来，应用程序将呈现出令人难以想象的图形动画效果。

7.1.2 动画连接对话框

给图形对象定义动画连接是在“动画连接”对话框中进行的。在组态王开发系统中双击图形对象（不能有多个图形对象同时被选中），弹出动画连接对话框。

 注意：

对不同类型的图形对象弹出的对话框大致相同。但是对于特定属性对象，有些是灰色的，表明此动画连接属性不适应于该图形对象，或者该图形对象定义了与此动画连接不兼容的其它动画连接。

 例如：以圆角矩形为例：

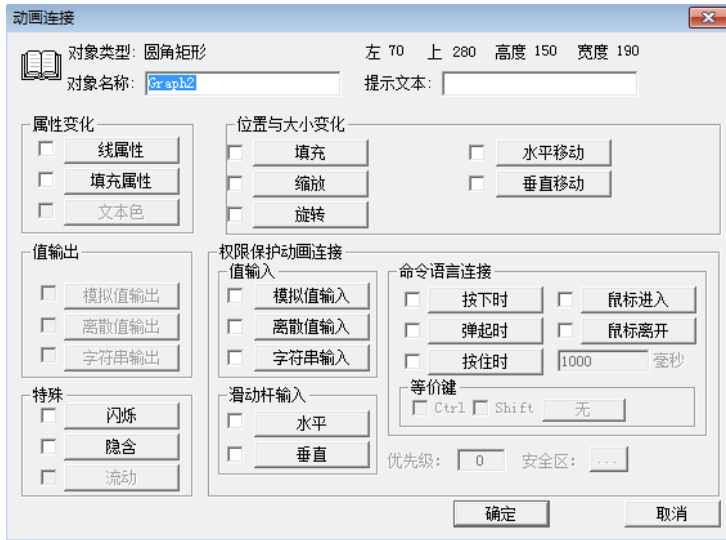


图7.1 动画连接属性对话框


- ◇ 对话框的第一行标识出被连接对象的名称和左上角在画面中的坐标以及图形对象的宽度和高度。
- ◇ 对话框的第二行提供“对象名称”和“提示文本”编辑框。“对象名称”是为图素提供的唯一的名称，供以后的程序开发使用，暂时不能使用。“提示文本”的含义为：当图形对象定义了动画连接时，在运行的时候，鼠标放在图形对象上，将出现开发中定义的提示文本。
- ◇ 下面分组介绍所有的动画连接种类。
 - **属性变化**：共有三种连接（线属性、填充属性、文本色），它们规定了图形对象的颜色、线型、填充类型等属性如何随变量或连接表达式的值变化而变化（“连接表达式”的用法请见 10.2 节）。单击任一按钮弹出相应的连接对话框。线类

型的图形对象可定义线属性连接，填充形状的图形对象可定义线属性、填充属性连接，文本对象可定义文本色连接。

- **位置与大小变化：**这五种连接（水平移动、垂直移动、缩放、旋转、填充）规定了图形对象如何随变量值的变化而改变位置或大小。不是所有的图形对象都能定义这五种连接。单击任一按钮弹出相应的连接对话框。
- **值输出：**只有文本图形对象能定义三种值输出连接中的某一种。这种连接用来在画面上输出文本图形对象的连接表达式的值。运行时文本字符串将被连接表达式的值所替换，输出的字符串的大小、字体和文本对象相同。按动任一按钮弹出相应的输出连接对话框。
- **用户输入：**所有的图形对象都可以定义为三种用户输入连接中的一种，输入连接使被连接对象在运行时为触敏对象。当 TouchView 运行时，触敏对象周围出现反显的矩形框，可由鼠标或键盘选中此触敏对象。按 SPACE 键、ENTER 键或鼠标左键，会弹出输入对话框，可以从键盘键入数据以改变数据库中变量的值。
- **特殊：**所有的图形对象都可以定义闪烁、隐含两种连接，这是两种规定图形对象可见性的连接。按动任一按钮弹出相应连接对话框。
- **滑动杆输入：**所有的图形对象都可以定义两种滑动杆输入连接中的一种，滑动杆输入连接使被连接对象在运行时为触敏对象。当 TouchView 运行时，触敏对象周围出现反显的矩形框。鼠标左键拖动有滑动杆输入连接的图形对象可以改变数据库中变量的值。
- **命令语言连接：**所有的图形对象都可以定义五种命令语言连接中的一种，命令

语言连接使被连接对象在运行时成为触敏对象。当 Touchview 运行时，触敏对象周围出现反显的矩形框，可由鼠标或键盘选中。按 SPACE 键、ENTER 键或鼠标左键，就会执行定义命令语言连接时用户输入的命令语言程序。按动相应按钮弹出连接的命令语言对话框。

- **等价键：**设置被连接的图素在被单击执行命令语言时与鼠标操作相同功能的快捷键。
- **优先级：**此编辑框用于输入被连接的图形元素的访问优先级级别。当软件在 Touchview 中运行时，只有优先级级别不小于此值的操作员才能访问它，这是“组态王”保障系统安全的一个重要功能。
- **安全区：**此编辑框用于设置被连接元素的操作安全区。当工程处在运行状态时，只有在设置安全区内的操作员才能访问它，**安全区与优先级**一样是“组态王”保障系统安全的一个重要功能。

 注意：

对于优先级和安全区只有那些有特定动画连接的图形对象可以设置优先级和安全区，这几种动画连接是：模拟值输入连接、离散值输入连接、字符串输入连接、水平滑动杆输入、垂直滑动杆输入连接、命令语言连接（鼠标或等价键按下时、按住时、弹起时）。

7.2 通用控制项目

7.2.1 图形编辑工具

组态王的工具箱经过精心设计，把使用频率较高的命令集中在一块面板上，非常便于操作，而且节省屏幕空间，方便您查看整个画面的布局。工具箱中的每个工具按钮都有“浮动提示”，帮助您了解工具的用途。

7.2.1.1 工具箱简介

图形编辑工具箱是绘图菜单命令的快捷方式。菜单命令在第四章已经详细介绍过，本节介绍动画制作时常用的图形编辑工具箱和其它几个常用工具。

每次打开一个原有画面或建立一个新画面时，图形编辑工具箱都会自动出现，如下图7.2所示。



图7.2 工具箱

在菜单“工具/显示工具箱”的左端有“✓”号，表示选中菜单；没有“✓”号，屏幕上的工具箱也同时消失，再一次选择此菜单，“✓”号出现，工具箱又显示出来。或使用<F10>键来切换工具箱的显示/隐藏。菜单如下图 7.3 所示。



图7.3 显示工具箱

工具箱提供了许多常用的菜单命令，也提供了菜单中没有的一些操作。当鼠标放在工具箱任一按钮上时，立刻出现一个提示条表明此工具按钮的功能，如下图 7.4 所示。



图7.4 工具箱提示

用户在每次修改工具箱的位置后，组态王会自动记忆工具箱的位置，当用户下次进入组态王时，工具箱返回上次用户使用时的位置。

! 注意:

如果由于不小心操作导致找不到工具箱了，从菜单中也打不开，请进入组态王的安装路径“kingview”下，打开 toolbox.ini 文件，查看最后一项[Toolbox]是否位置坐标不在屏幕显示区域内，用户可以自己在该文件中修改。注意不要修改别的项目。

7.2.1.2 工具箱速览

工具箱中的工具大致分为四类。

- ✧ **画面类**：提供对画面的常用操作，含新建、打开、关闭、保存、删除、全屏显示等。
- ✧ **编辑类**：绘制各种图素（矩形、椭圆、直线、折线、多边形、圆弧、文本、点位图、按钮、菜单、报表窗口、实时趋势曲线、历史趋势曲线、控件、报警窗口）的工具；剪切、粘贴、复制、撤消、重复等常用编辑工具；合成、分裂组合图素，合成、分裂单元；对图素的前移，后移，旋转，镜像等操作工具。
- ✧ **对齐方式类**：这类工具用于调整图素之间的相对位置，能够以上、下、左、右、水平、垂直等方式把多个图素对齐；或者把它们水平等间隔、垂直等间隔放置。
- ✧ **选项类**：提供其它一些常用操作，比如全选、显示调色板、显示画刷类型、显示线形、网格显示/隐藏、激活当前图库、显示调色板等。

7.2.1.3 工具箱详解

工具箱中的按钮是绘图菜单命令的快捷方式，其用法可以参考“第四章 组态王开发环境 - 工程浏览器”的相关内容。本节介绍图形编辑工具箱和其它几个常用工具的使用法。



新画面

与菜单“文件\新画面”效果相同，它用于定义新画面的名称、大小、位置、风格等，以及画面在磁盘上对应的文件名。



打开画面

与菜单“文件\打开”效果相同，它用于打开指定的一个或几个已经存在的画面。



关闭画面

与菜单“文件\关闭”选择项相同，它用于关闭指定的一个或几个已经存在的画面。



保存画面

与菜单“文件\存入”效果相同，它用于保存指定的一个或几个已经存在的画面。



删除画面

与菜单“文件\删除”效果相同，它用于删除指定的一个或几个已经存在的画面。



全屏显示

与菜单“编辑\全屏显示”效果相同，它用于全屏显示当前画面。



选中图素

与菜单“工具\选中图素”效果相同，它用于对象的选择、移动和重定尺寸。这是鼠标器的缺省工作方式，大多数工具在完成操作后自动返回到此方式。比如选用“直线”工具画完一条直线后，“直线”工具按钮自动弹起，“选中图素”按钮自动压下，表明鼠标器自动恢复到“选中图素”状态。



画直线

与菜单“工具\直线”效果相同，以当前线型绘制一条直线。



画扇形或弧形

与菜单“工具\扇形（弧形）”效果相同，以当前线型和填充模式绘制一个扇形。绘

制弧形还要选择适当的填充模式。



画椭圆

与菜单“工具\椭圆”效果相同，选中本工具可画出一个与鼠标拖曳的矩形相内切的椭圆。



画圆角矩形

与菜单“工具\圆角矩形”效果相同，选中本工具可画出直角矩形。若要画圆角矩形还需选用“改变图素形状”工具加以修改。



画折线

与菜单“工具\折线”效果相同，以当前线型绘制一条折线。



画立体管道

与菜单“工具\立体管道”效果相同。选中本工具可画出立体管道。



画多边形

与菜单“工具\多边形”效果相同，以当前线型和填充模式绘制一个多边形。



输入文本

与菜单“工具\文本”效果相同，以当前字体输入文本。



画按钮

与菜单“工具\按钮”效果相同，输入按钮文本请选择菜单“工具\按钮文本”。



画菜单

与菜单“工具\菜单”效果相同，用于在画面上建立菜单。



定义报警窗口

与菜单“工具\报警窗口”效果相同，在选定区域内绘制报警窗口。



报表建立按钮

与菜单“工具\报表窗口”效果相同，在选定区域内绘制报表窗口。



画实时趋势图

与菜单“工具\实时趋势曲线”效果相同，在选定区域内绘制实时趋势曲线。



画历史趋势图

与菜单“工具\历史趋势曲线”效果相同，在选定区域内绘制历史趋势曲线。



画点位图

与菜单“工具\点位图”效果相同，本工具只能确定点位图的位置和大小，输入点位图请选择菜单“编辑\粘贴点位图”。



插入控件

与菜单“编辑\插入控件”效果相同。



插入通用控件

与菜单“编辑\插入通用控件”效果相同。



打开图库

与菜单“图库\打开图库”效果相同，用于激活图库窗口。



恢复

此命令用于取消前面执行过的操作。



重做

此命令用于恢复先前执行过的操作。



剪切

与菜单“编辑\剪切”效果相同，剪切之前请先选中一个或多个图素对象。剪切后图形对象暂存于内存，可用“粘贴”工具恢复到画面上。



拷贝

与菜单“编辑\拷贝”效果相同，可复制一个或多个被选中的对象。



粘贴

与菜单“编辑\粘贴”效果相同，把一个或多个剪切掉的对象从内存恢复到画面上。



复制

与菜单“编辑\复制”效果相同，复制一个或多个被选中的图素对象。



合成组合图素

与菜单“排列\合成组合图素”效果相同，将二个或多个选中的图素对象组合成一

个整体，作为构成画面的基本元素。有动画连接的图素、组合图素、点位图或单元不能作为基本图素来合成新的组合图素。



分裂组合图素

与菜单“排列\分裂组合图素”效果相同，将由多个图素合成的复杂图素分解为基本图素。若该单元有动画连接，则会自动删除动画连接。



合成单元

与菜单“排列\合成单元”效果相同，将所有被选中的图素或单元组合成一个新的单元，各组成部分的动画连接保持不变。



分裂单元

与菜单“排列\分裂单元”效果相同，将单元分解成原来生成单元的各基本图素。若该单元有动画连接，分解后各组成部分的动画连接保持不变。



图素后移

与菜单“排列\图素后移”效果相同，使一个或多个选中的图素对象移至所有与之相交的图素后面。



图素前移

与菜单“排列\图素前移”效果相同，使一个或多个选中的图素对象移至所有与之相交的图素前面。



改变图素形状

与菜单“工具\改变图素形状”效果相同，用于改变圆角矩形的弧度、扇形或弧形的角度、多边形或多边线的各顶点相对位置。



改变字体

与菜单“工具\字体”效果相同，用于改变字体的缺省设置。



图素顺时针转 90 度

与菜单“排列\顺时针旋转 90 度”效果相同，把被选中的单个图素以图素中心为圆心顺时针旋转 90 度，也可以旋转多个图素合成的组合图素，但是不能同时旋转多个图素对象，不能旋转单元。



图素逆时针转 90 度

与菜单“排列\逆时针旋转 90 度”效果相同，把被选中的单个图素以图素中心为圆心逆时针旋转 90 度，也可以旋转多个图素合成的组合图素，但是不能同时旋转多个图素对象，不能旋转单元。



水平翻转

与菜单“排列\水平翻转”效果相同，把被选中的单个图素水平翻转，也可以翻转多个图素合成的组合图素。翻转的轴线是包围图素或复杂图素的矩形框的垂直对称轴。此工具不能同时翻转多个图素对象，不能翻转单元。



垂直翻转

与菜单“排列\水平翻转”效果相同，把被选中的单个图素水平翻转，也可以翻转多个图素合成的组合图素。翻转的轴线是包围图素或复杂图素的矩形框的垂直对称轴。此工具不能同时翻转多个图素对象，不能翻转单元。



图素上对齐

与菜单“排列\对齐\上对齐”效果相同，使所有被选中对象的上边界与最上面的边界平齐。在各种对齐方式中，被选中对象的边界是指包围对象的 8 个小矩形构成的矩形框。对象的水平轴、垂直轴也分别是指这个矩形框的水平对称轴和垂直对称轴。



图素水平对齐

与菜单“排列\对齐\水平对齐”效果相同，把所有被选中对象的水平轴调整到同一水平线上。



图素下对齐

与菜单“排列\对齐\下对齐”效果相同，使所有被选中对象的下边界与最下面的边界平齐。



图素左对齐

与菜单“排列\对齐\左对齐”效果相同，使所有被选中对象的左边界与最左面的边界平齐。



图素垂直对齐

与菜单“排列\对齐\垂直对齐”效果相同，把所有被选中对象的垂直轴调整到同一竖直线上。



图素右对齐

与菜单“排列\对齐\右对齐”效果相同，使所有被选中对象的右边界与最右面的边界平齐。



图素水平等间距

与菜单“排列\水平方向等间隔”效果相同，使所有被选中对象在水平方向以等间距排列，各图素垂直方向位置不变。



图素垂直等间距

与菜单“排列\垂直方向等间隔”效果相同，使所有被选中对象在垂直方向以等间距排列，各图素水平方向位置不变。



对齐栅格

与菜单“排列\对齐网格”效果相同，用于显示/隐藏画面上的网格，并且决定画面上图形对象的边界是否与栅格对齐。对齐网格后，图形对象的移动也将以栅格宽度为距离单位。



显示调色板

与菜单“工具\显示调色板”效果相同，用于显示/隐藏调色板窗口。调色板的使用方法请详见 7.2.1.6 节。



全选

与菜单“编辑\全选”效果相同，使画面上全部图素、复杂图素、单元都处于选中状态。所有对象周围都出现 8 个小矩形。



显示画刷类型

与菜单“工具\显示画刷类型”效果相同。用于显示/隐藏画刷和过渡色类型工具条。画刷类型的使用方法请详见 7.2.1.4 节。



显示线形

与菜单“工具\显示线形”效果相同。用于显示/隐藏线形选择工具条。线形类型的使用方法请详见 7.2.1.5 节。

在工具箱底部的文本框中显示被选中对象的 x、y 坐标和对象大小信息。如图 7.5：

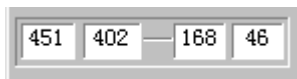



图 7.5 图形对象大小及位置

- 第一个文本框显示被选中对象的 x 坐标（左边界）。
- 第二个文本框显示被选中对象的 y 坐标（上边界）。
- 第三个文本框显示被选中对象的宽度。
- 第四个文本框显示被选中对象的高度。

用户可以修改上述文本框中的任何一项，修改对象的位置或大小，方便地编辑图形。

7.2.1.4 画刷类型工具的使用

组态王提供八种画刷（填充）类型和二十四种画刷（填充）过渡色类型。显示/隐藏画刷类型工具条可通过选择菜单“工具/显示画刷类型”或工具箱的按钮“”（显示画刷类型）来实现。画刷类型工具条可使工程人员方便地选用各种画刷填充类型和不同的过渡色效果。画刷类型工具条如下图 7.6 所示。

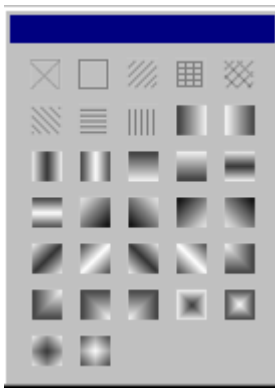


图 7.6 画刷类型工具条

目前支持画刷填充和过渡色的图素有：圆角矩形、椭圆、圆弧（或扇形）、多边形。


✧ 画刷填充类型及使用方法

1. 在画面中选中需改变画刷填充类型的图素；
2. 从画刷类型工具条中单击画刷填充类型按钮。画刷填充支持八种类型：




全部填充 与菜单“工具\填充属性\ ”相同，用填充色将图素实心填充。



透明填充 同菜单“工具\填充属性\ ”，图素全部透明，只留边框。




斜线填充

与菜单“工具\填充属性\  ”相同，图素用左下角→右上角斜线填充，线的颜色为定义的填充色。




斜线填充

与菜单“工具\填充属性\  ”相同，图素用左上角→右下角斜线填充，线的颜色为定义的填充色。




网格填充

与菜单“工具\填充属性\  ”相同，图素用水平垂直网格填充，线的颜色为定义的填充色。




网格填充

与菜单“工具\填充属性\  ”相同，图素用斜线网格填充，线的颜色为定义的填充色。




水平直线填充

与菜单“工具\填充属性\  ”相同，图素用水平直线填充，线的颜色为定义的填充色。



垂直直线填充

与菜单“工具\填充属性\  ”相同，图素用垂直直线填充，线的颜色为定义的填充色。

◇ 过渡色类型的使用方法




- 1、在画面中选中需改变过渡色类型的图素；
- 2、在画刷类型工具条中单击过渡色画刷类型按钮。组态王支持二十四种过渡色效果：

● 水平过渡（从左到右）





A.








颜色过渡为：填充色→背景色。


- B.  颜色过渡为：背景色→填充色。
- C.  颜色过渡为：填充色→背景色→填充色。
- D.  颜色过渡为：背景色→填充色→背景色。


● 垂直过渡（从上到下）


- A.  颜色过渡为：填充色→背景色。
- B.  颜色过渡为：背景色→填充色。
- C.  颜色过渡为：填充色→背景色→填充色。
- D.  颜色过渡为：背景色→填充色→背景色。

● 对角过渡


- A.  颜色过渡为：填充色→背景色，方向为：左上角至右下角。
- B.  颜色过渡为：填充色→背景色，方向为：右上角至左下角。
- C.  颜色过渡为：背景色→填充色，方向为：左上角至右下角。
- D.  颜色过渡为：背景色→填充色，方向为：右上角至左下角。
- E.  颜色过渡为：填充色→背景色→填充色，方向为：左上角至右下角。


F.  颜色过渡为：背景色→填充色→背景色，方向为：左上角至右下角。


G.  颜色过渡为：填充色→背景色→填充色，方向为：右上角至左下角。


H.  颜色过渡为：背景色→填充色→背景色，方向为：右上角至左下角。

● 垂直角过渡


A.  颜色过渡为：填充色→背景色，方向为：左上角至右下角。


B.  颜色过渡为：填充色→背景色，方向为：右上角至左下角。


C.  颜色过渡为：背景色→填充色，方向为：左上角至右下角。


D.  颜色过渡为：背景色→填充色，方向为：右上角至左下角。

● 锥形过渡

A.  颜色过渡为：背景色→填充色，方向为：矩形由内向外。

B.  颜色过渡为：填充色→背景色，方向为：矩形由内向外。

C.  颜色过渡为：背景色→填充色，方向为：菱形由内向外。

D.  颜色过渡为：背景色→填充色，方向为：菱形由内向外。

调整图素过渡色的填充色和背景色，通过调色板的“填充色”和“背景色”选项来完成。填充色和背景色均支持无限色。也可以先用调色板选择图素过渡色的填充色和背景色（内容详见调色板的使用），再选择过渡类型。

7.2.1.5 线形类型工具的使用

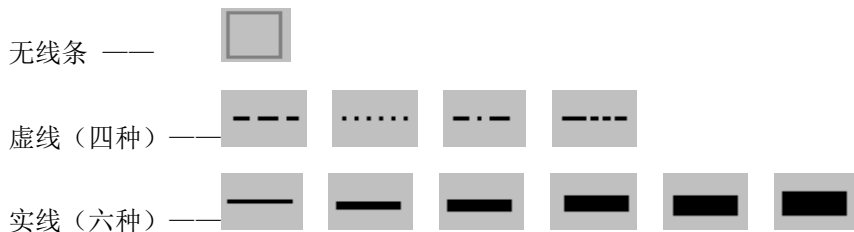
组态王系统支持 11 种线形。线形窗口可方便工程人员改变图素线条的类型。线形窗口如下图 7.7 所示。



图 7.7 线形选择工具条

◇ 线形类型及使用方法

组态王系统支持 11 种线形，如图所示。



其具体使用为：在画面中选定需修改线形的图素后，单击“线形选择工具条”上相应按钮，即可选择图素线条粗细、虚实等属性。

7.2.1.6 调色板的使用

“调色板”就是“颜料盒”，共有无限种颜色。显示/隐藏调色板可通过选择菜单“工

具/显示调色板”或单击工具箱中的“显示调色板按钮”来实现。应用“调色板”可以对各种图形、文本及窗口等进行颜色修改，图形包括圆角矩形、椭圆、直线、折线、扇形、多边形、管道、文本以及窗口背景色等。“调色板”具有无限色功能，即除了可以选定“基本颜色”外，还可以利用“无限色”来编辑各种颜色，并能保存和读取调色信息。“调色板”外观如下图 7.8 所示。



图 7.8 系统调色板

“调色板”的使用是很简单的，真正的困难在于画面上颜色的搭配，工程人员在选择颜色时要考虑到整体的和谐。

调色板工具条

调色板支持无限色。当前显示的调色板显示基本色。如下图 7.9 所示。用户在每次修改调色板的位置后，组态王会自动记忆调色板的位置，当用户下次进入组态王时，调色板返回上次用户使用时的位置。

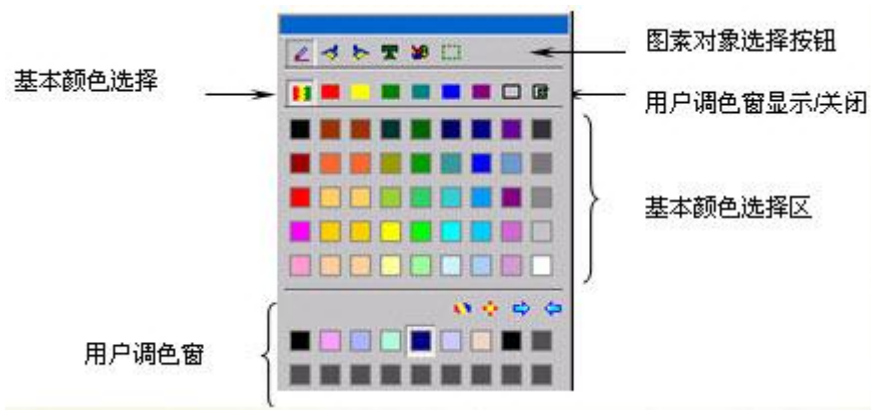


图 7.9 调色板

◇ 对象选择按钮：

选择要填充颜色的对象，从左到右依次为：“线条色（线属性）”、“填充色（填充属性/前景色属性）”、“背景色（背景色属性）”、“字符色（字体颜色）”、“窗口色（窗口背景色）”、“透明色（点位图透明色）”，如下图 7.10 所示。当鼠标箭头指向按钮时，出现相应名称，方便操作。

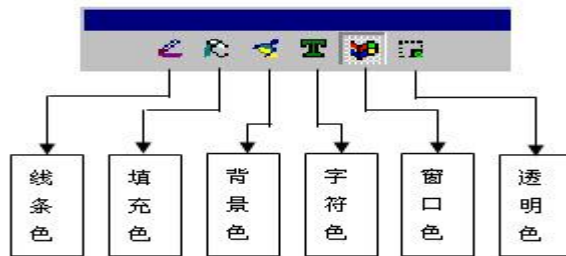


图 7.10 颜色属性选择按钮

● 线条色按钮

按下此按钮后，从调色板中选择一种作为线的颜色，以后绘制直线、折线，和矩形、椭圆等封闭图形的边框都将使用这种颜色，直到在调色板中为“线属性”选用新颜色。如果想要改变画面上某直线、折线或封闭图形边框的线属性，请先选中此对象，然后按下“线属性”按钮，从调色板中选取需要的颜色即可。这种方法也适合于复杂图素，但不适合于单元。

- **填充色按钮**

按下此按钮后，可从调色板中选择一种作为填充封闭图形的缺省颜色。如果想要改变画面上某封闭图形的填充属性，请先选中此对象，然后按下“填充属性”按钮，从调色板中选取需要的颜色。这种方法也适合于复杂图素，但不适合于单元。

- **背景色按钮**

该属性主要是用来构造过渡色效果，如果选择过渡色效果，按下此按钮后，从“调色板”中选择的该颜色将作为过渡色的背景色。默认为黑色。

- **字符色按钮**

按下此按钮后，可从调色板中选择一种作为字体颜色，这种颜色是各种文本的缺省颜色。

- **窗口色按钮**

按下此按钮后，可从调色板中选择一种作为窗口背景色。这种颜色是各种对象的缺省背景颜色。

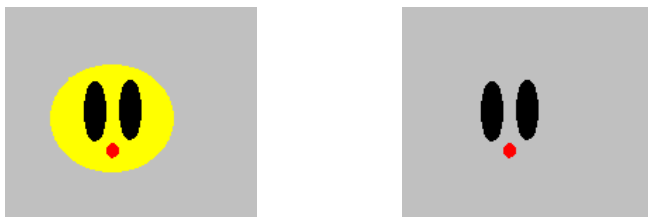
- **点位图透明颜色按钮**

按下此按钮后，可从调色板中选择一种作为点位图透明颜色，这种颜色是各种点位

图透明的缺省颜色。在组态王开发系统中，当从 WINDOWS 的剪切板上粘贴点位图时，设定点位图透明颜色，使用菜单命令“点位图透明”，则点位图中设定的颜色被透明，从而使被点位图覆盖的画面背景透明显示。

 例如：

如果设定点位图透明颜色为黑色，当使用菜单命令“点位图透明”，则点位图中的黑色将被透明。如下图 7.11 所示。



执行点位图透明前

执行点位图透明后

图 7.11 点位图透明前后

- ✧ **基本颜色选择：**直接从当前的调色板中选择用户需要的颜色。第一个为多色选择，其余为单色选择。
- ✧ **用户调色窗：**为用户提供的选择无限色设置，其按钮定义从左到右依次为：系统调色窗口、吸色管、存储调色信息设置，读取调色信息设置。下方为选色区。“调色板”选择按钮下面有一无限色调色和存取区，如下图 7.12 所示。

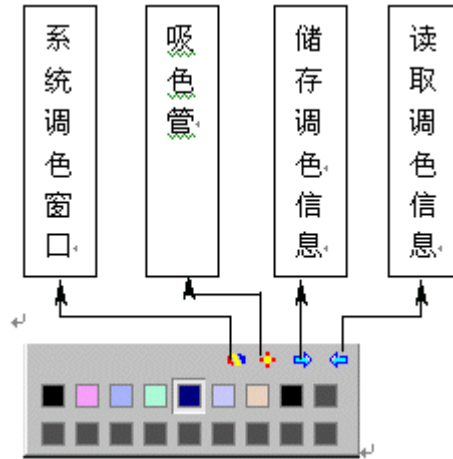
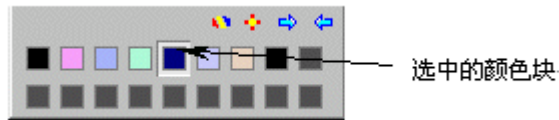


图 7.12 无限色调色和存取区

主要功能有：调色、吸色、储存调色信息、读取调色信息。

- 系统调色窗口

用户若自己定义颜色，首先要从存取区中选中的一个颜色块，如下图所示。



然后左键点击系统调色窗口按钮（用户调色窗的第一个按钮），弹出“真彩色调色盘”窗口，如下图 7.13 所示。

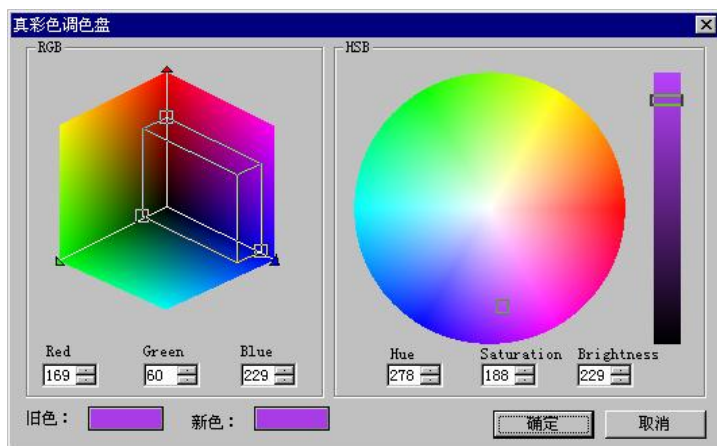


图 7.13 “真彩色调色盘”窗口

“真彩色调色板”提供两种调色方式：RGB、HSB。RGB 是红、绿、蓝三色选取方式；HSB 是颜色、饱和度、亮度选取方式。使用鼠标左键点击窗口颜色显示中“颜色滑块”——小方块，按住并拖动鼠标，颜色滑块随之移动，下边的数字框中的数字随之变化。数字框中的数字分别代表红、绿、蓝、颜色、饱和度、亮度的数值，用户也可直接改变数字框中的数值来改变颜色。选定好的颜色，在“新色”中显示出来。“旧色”是选中对象的原有颜色。定义好的一种玫红色如下图 7.14 所示。

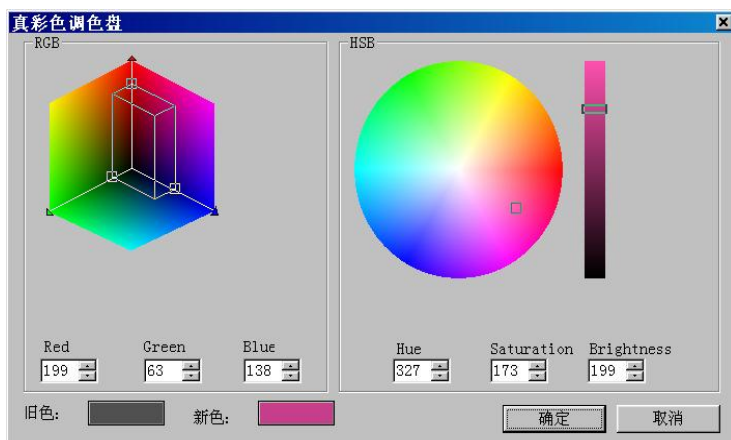
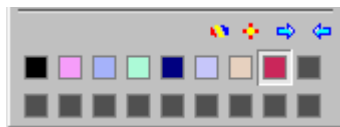


图 7.14 再定义的颜色

上图中在调色区中选好的颜色在“新色”中显示出来。点击“确定”按钮，即可将定义的颜色加入到调色板存取区中先选定好的颜色块中。如下图所示。

- 吸色管



吸色管

在不同区域或对象之间复制颜色，从存取区中选定一个颜色块，单击吸色管按钮，此时鼠标指针变成一个吸管的形状，然后左键点击选择要复制其颜色的对象，颜色复制到所选定的颜色块中，复制的颜色就可以和上面的基本颜色一样使用了（作为基本颜色来使用）。如下图 7.15 所示。

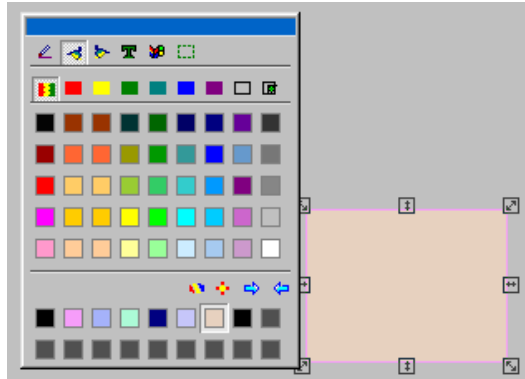


图 7.15 利用吸色管来定义颜色

- 储存调色信息

对于用户自定义的颜色，在开发工程中会经常用到。调色板能够保存用户自定义的颜色，在使用时用户只需调出存储的调色信息。用户可以储存 18 种无限色。首先定义无限色：在自定义颜色存取区中单击一个小颜色块；然后左键点击系统调色窗口按钮，使用“添加颜色窗口”定义无限色；最后单击“确定”，就定义好了一种颜色。定义好所需的颜色后，单击“储存调色信息”按钮，弹出如图 7.16 所示的对话框。



图 7.16 存储调色板信息

可将调色板的自定义颜色以.CPF 文件存储。

- 读取调色信息

若在别的工程中需调用存储的颜色，单击“读取调色信息”按钮打开相应的.CPF 文件即可，如图 7.17 所示。存储的颜色就会在用户调色窗的方块上显示出来。



图 7.17 读取调色信息

若用户选择了“以只读方式打开”选项，则用户在使用过程中不能修改该自定义调色板。

7.2.2 变量浏览器的使用

变量浏览器是供用户在进行动画连接或编写命令语言脚本时选择变量或变量域时用的。在动画连接输入表达式对话框中单击右边“？”按钮可以打开“选择变量名”窗口，用于查看、选择本机和和其它站点已定义的基本变量和结构变量以及变量域。如下图 7.18 所示。

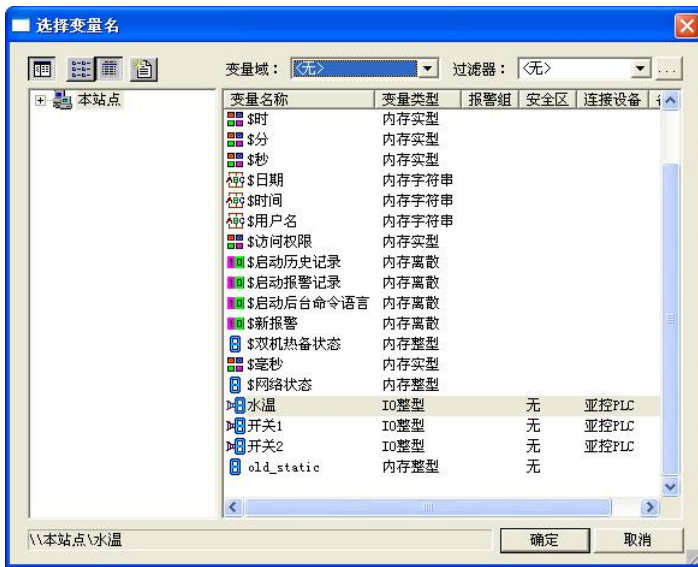


图7.18 选择变量名

◇ 功能按钮

左上角四个按钮功能描述如下 7.19 图所示。

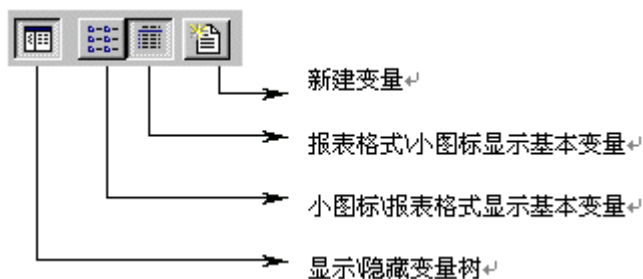


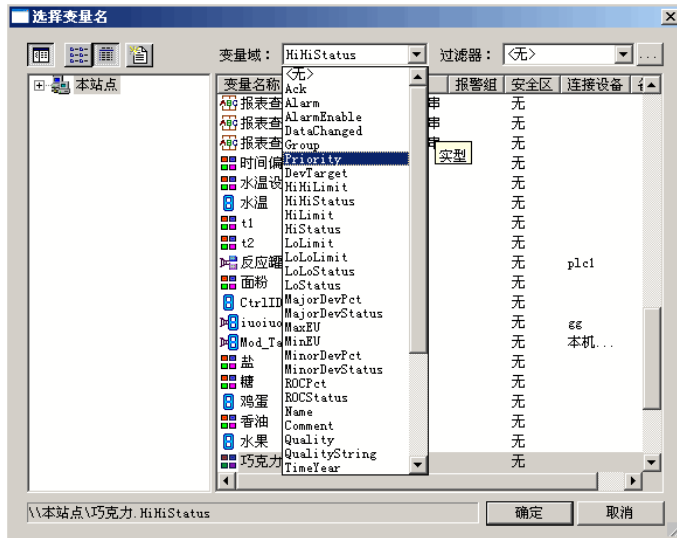
图 7.19 变量浏览器快捷菜单

- **显示/隐藏变量树**：选中此按钮可以显示/隐藏左边的变量树。按钮凹下时显示变量树，凸起时隐藏变量树。
- **小图标/报表格式显示基本变量**：选中此按钮，按钮凹下时，右边变量显示窗口中的变量是以小图标形式显示，没有变量的详细介绍列表。当选择“报表格式/小图标显示基本变量”按钮时，该按钮变为凸起。
- **报表格式/小图标显示基本变量**：选中此按钮，按钮凹下时，右边变量显示窗口中的变量是以报表的形式显示，有变量的详细介绍列表，如：变量类型、报警组、安全区、连接设备、备注等。变量可以根据单击列表第一列表头文本自动排序，当选择“小图标/报表格式显示基本变量”按钮时，该按钮变为凸起。
- **新建变量**：单击此按钮，弹出“定义变量”窗口，可直接新建变量。方法与在数据词典中定义变量相同。

◇ 变量域

单击变量域列表框按钮，下拉框中会显示当前变量的所有可用域。如下图所示，选择变量域及系统对变量域数据类型的提示。具体每种变量如何定义其变量域请参见

“第五章 变量的定义和管理”一章中的相关内容。



◇ 过滤器

用户可以选择过滤条件以滤掉列表中不符合条件的变量，方便用户选择变量。单击过滤器列表框按钮，可以从下拉框中看到定义好的过滤条件，如果用户没有定义过滤条件，则列表框中只显示“无”。单击右侧“...”按钮可弹出定义、浏览和选择过滤器信息的“定义过滤条件”对话框，如下图 7.20 所示。



图7.20 定义过滤条件对话框

在“过滤器名称”中输入定义的过滤条件的名称，在“过滤条件”中输入或选择各项过滤条件，如：变量的变量名、变量类型、报警组、安全区、连接设备、注释等，点击右边“增加”按钮，定义好的过滤条件就保存下来了。例如，定义一个过滤条件：要将变量词典中所有 IO 整型变量筛选出来。在“过滤器名称”栏可以填写“所有 IO 整型变量”；过滤条件中从“变量类型”里选择“IO 整型”，其他过滤条件不做限制。如下图 7.21 所示。

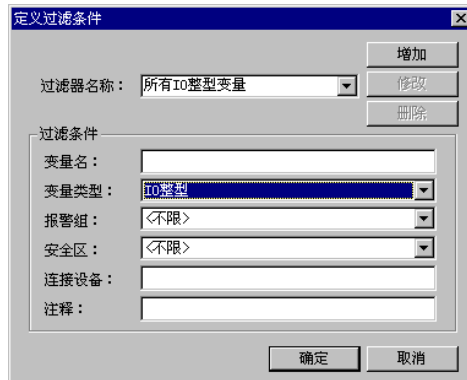


图7.21 定义好的过滤条件

单击“增加”按钮将定义的 IO 整型过滤条件保存下来。单击“确定”返回到“选择变量名”窗口。从“过滤器”下拉框中选择“所有 IO 整型变量”，在变量列表中只显示出变量类型为 IO 整型的变量，其余不符合条件的变量被过滤掉了。如图 7.22 所示。



图 7.22 过滤后的变量显示

定义好的过滤条件可以进行修改和删除。修改过滤条件时，先对过滤名称和过滤条件进行修改，然后点击“修改”按钮，所作的修改就被保存下来了。如果想要删除某一个过滤条件，先选中过滤名称，然后点击“删除”按钮，即可删除过滤条件。定义好过滤条件，单击过滤器列表框按钮，可以从下拉框中选择过滤条件，滤掉列表控件中不符合条件的变量，在“选择变量名”窗口右边显示的变量就是符合过滤条件的变量。

◇ 结构变量

如果定义了结构变量，在“选择变量名”窗口左边“本站点”前有一个“+”标志，单击“+”显示出定义好的结构变量，如下图 7.23 所示。



图 7.23 选择结构变量

在右侧的列表框中选择结构成员，然后单击“确定”按钮。

◇ 其它站点变量

如果定义了其它站点，则会在变量浏览器左边的目录树中显示出站点名称和该站点下结构变量的名称。选择方法与本站点变量的选择方法相同。

7.2.3 表达式和运算符

连接表达式是定义动画连接的主要内容，因为连接表达式的值决定了画面上图素的动画效果。表达式由数据字典中定义的变量、变量域、报警组名、数值常量以及各种运算符组成，与 C 语言中的表达式非常类似。


在连接表达式中不允许出现赋值语句，表达式的值在“组态王”运行时计算。变量名

和报警组名可以直接从变量浏览器中选择, 出现在表达式中, 不必加引号, 但区分大小写, 变量的域名不区分大小写。

7.2.3.1 运算符

连接表达式中可用到的运算符如下:

~	取补码, 将整型变量变成 " 2 " 的补码。
*	乘法
/	除法
%	模运算
+	加法
-	减法 (双目)
&	整型量按位与
	整型量按位或
^	整型量异或
&&	逻辑与
	逻辑或
<	小于
>	大于
<=	小于或等于
>=	大于或等于
==	等于
!=	不等于

 注意：

除上述运算符以外，还可使用如下运算符增强运算功能：

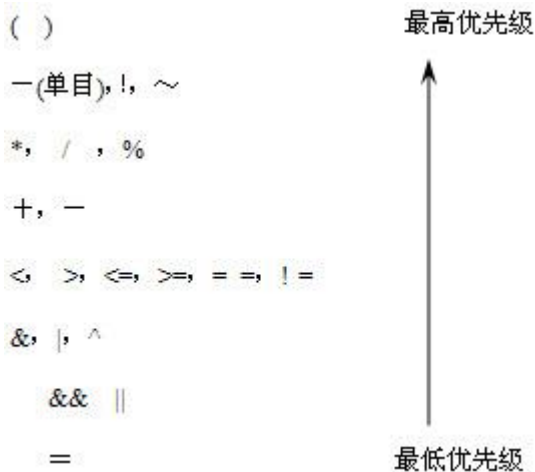
— 取反，将正数变为负数（单目）。

! 逻辑非

() 括号，保证运算按所需次序进行。

7.2.3.2 优先级

下面列出算符的运算次序，首先计算最高优先级的算符，再依次计算较低优先级的算符。同一行的算符有相同的优先级。



7.2.3.3 表达式举例

- ◇ 单独的变量或变量的域：开关、液面高度.alarm
- ◇ 复杂的表达式： 开关= =1 液面高度>50&&液面高度<80 (开关 1||开关 2)&&(液面高度.alarm)

7.3 动画连接详解

在“动画连接”对话框中，单击任一种连接方式，将会弹出设置对话框，本节详细解释各种动画连接的设置。如果要使用动画连接自动向导，请参考本章第四节“使用动画连接向导”。

7.3.1 线属性连接

在“动画连接”对话框中，单击“线属性”按钮，弹出连接对话框。

线属性连接是使被连接对象的边框或线的颜色和线形随连接表达式的值而改变。定义这类连接需要同时定义分段点（阈值）和对应的线属性。利用连接表达式的多样性，可以构造出许多很有用的连接。



例如：

可以用线颜色表示离散变量 EXAM 的报警状态，只须在连接表达式中输入 EXAM.Alarm，然后把下面的两个笔属性颜色对应的值改为 0（蓝色），1（红色）即可。软件在运行时，当警报发生时（EXAM.Alarm==1），线就由蓝色变成了红色；当

警报解除后，线又变为蓝色。在画面上画一个圆角矩形，双击该图形对象，弹出的动画连接对话框如下图示：按上述填好，按确定即可。

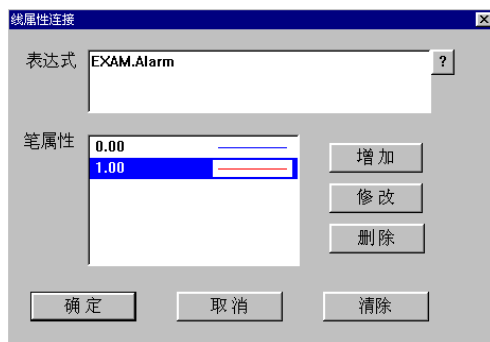


图7.24 线属性连接

线属性连接对话框中各项设置的意义如下：

- ◇ **表达式**：用于输入连接表达式，单击“？”按钮可以查看已定义的变量名和变量域。
- ◇ **增加**：增加新的分段点。单击**增加**弹出的输入新值对话框，在对话框中输入新的分段点（阈值）和设置笔属性。按鼠标左键击中“笔属性—线形”按钮弹出漂浮式窗口，移动鼠标进行选择；也可以使“线属性”按钮获得输入焦点，按空格键弹出漂浮式窗口，用TAB键在颜色和线型间切换，用移动键选择，按空格或回车确定选择。



图 7.25 输入阈值

- ◇ **修改**: 修改选中的分段点。修改对话框用法同输入新值对话框。
- ◇ **删除**: 删除选中的分段点。

7.3.2 填充属性连接

填充属性连接使图形对象的填充颜色和填充类型随连接表达式的值而改变,通过定义一些分段点(包括阈值和对应填充属性),使图形对象的填充属性在一段数值内为指定值。

本例为封闭图形对象定义填充属性连接,阈值为 0 时填充属性为白色,阈值为 100 时为黄色,阈值为 200 时为红色。画面程序运行时,当变量“温度”的值在 0 至 100 之间时,图形对象为白色;在 100 至 200 之间时为黄色,变量值大于 200 时,图形对象为红色。

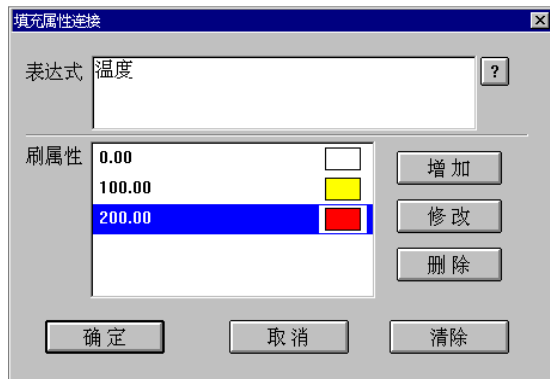


图7.26 填充属性连接

“填充属性”动画连接的设置方法为：在“动画连接”对话框中选择“填充属性”按钮，弹出的对话框（如上图）各项意义如下：

- ◇ **表达式**：用于输入连接表达式，右边的？可以查看已定义的变量名和变量域。
- ◇ **增加**：增加新的分段点。单击**增加**按钮弹出输入新值对话框：



图 7.27 填充属性—输入新值

在输入新值对话框中输入新的分段点的阈值和画刷属性，按鼠标左键击中“画刷属性—类型”按钮弹出画刷类型漂浮式窗口，移动鼠标进行选择；也可以使“填充属

性”按钮获得输入焦点，按空格键弹出漂浮式窗口，用 TAB 键在颜色和填充类型间切换，用移动键选择，按空格键或回车结束选择。按鼠标左键击中“画刷属性—颜色”按钮弹出画刷颜色漂浮式窗口，用法与“画刷属性—类型”选择相同。

- ◇ **修改：**修改选中的分段点。修改对话框用法同输入新值对话框。
- ◇ **删除：**删除选中的分段点。

7.3.3 文本色连接

文本色连接是使文本对象的颜色随连接表达式的值而改变，通过定义一些分段点（包括颜色和对应数值），使文本颜色在特定数值段内为指定颜色。如定义某分段点，阈值是 0，文本色为红色，另一分段点，阈值是 100，则当“压力”的值在 0 到 100 之间时（包括 0），“压力”的文本色为红色，当“压力”的值大于等于 100 时，“压力”的文本色为蓝色。



图7.28 文本色连接

文本色连接的设置方法为：在“动画连接”对话框中选择“文本色”按钮，弹出的对话框（如上图）中各项设置的意义如下：

- ◇ **表达式:** 用于输入连接表达式, 单击右侧的“?”按钮可以查看已定义的变量名。
- ◇ **增加:** 增加新的分段点。单击**增加**按钮弹出输入新值对话框:



图7.29 文本色连接—输入新值

在输入新值对话框中输入新的分段点的阈值和颜色, 按鼠标左键击中“文本色”按钮弹出漂浮式窗口, 移动鼠标进行选择; 也可以使“颜色”按钮获得输入焦点, 按空格键弹出漂浮式窗口, 用移动键选择, 按空格键或回车结束。

- ◇ **修改:** 修改选中的分段点。修改对话框用法同输入新值对话框。
- ◇ **删除:** 删除选中的分段点。

7.3.4 水平移动连接

水平移动连接是使被连接对象在画面中随连接表达式值的改变而水平移动。移动距离以像素为单位, 以被连接对象在画面制作系统中的原始位置为参考基准的。水平移动连接常用来表示图形对象实际的水平运动。

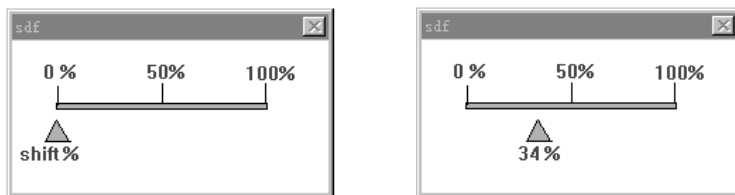


图7.30 水平连接实例

本例中建立一个指示器，在画面上画一个三角形（将其设置“水平移动”动画连接属性），以表示 shift 量的实际大小。左图是设计状态，右图是在 Touchview 中的运行状态。

水平移动连接的设置方法为：在“动画连接”对话框中单击“水平移动”按钮，弹出“水平移动连接”对话框：



图7.31 水平连接

对话框中各项设置的意义如下：

- ◇ **表达式**：在此编辑框内输入合法的连接表达式，单击“？”按钮可查看已定义的变量名和变量域。
- ◇ **向左**：输入图素在水平方向向左移动(以被连接对象在画面中的原始位置为参考基准)

的距离。

- ◇ **最左边**: 输入与图素处于最左边时相对应的变量值, 当连接表达式的值为对应值时, 被连接对象的中心点向左(以原始位置为参考基准)移到最左边规定的位置。
- ◇ **向右**: 输入图素在水平方向向右移动(以被连接对象在画面中的原始位置为参考基准)的距离。
- ◇ **最右边**: 输入与图素处于最右边时相对应的变量值, 当连接表达式的值为对应值时, 被连接对象的中心点向右(以原始位置为参考基准)移到最右边规定的位置。

7.3.5 垂直移动连接

垂直移动连接是使被连接对象在画面中的位置随连接表达式的值而垂直移动。移动距离以象素为单位, 以被连接对象在画面制作系统中的原始位置为参考基准的。垂直移动连接常用来表示对象实际的垂直运动, 单击“动画连接”对话框中的“垂直移动”按钮, 弹出“垂直移动连接”对话框:



图7.32 垂直移动连接

对话框中各项设置的意义如下：

- ◇ **表达式**：在此编辑框内输入合法的连接表达式，单击“？”按钮可以查看已定义的变量名和变量域。
- ◇ **向上**：输入图素在垂直方向向上移动(以被连接对象在画面中的原始位置为参考基准)的距离。
- ◇ **最上边**：输入与图素处于最上边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向上(以原始位置为参考基准)移到最上边规定的位置。
- ◇ **向下**：输入图素在垂直方向向下移动(以被连接对象在画面中的原始位置为参考基准)的距离。
- ◇ **最下边**：输入与图素处于最下边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向下(以原始位置为参考基准)移到最下边规定的位置。

7.3.6 缩放连接

缩放连接是使被连接对象的大小随连接表达式的值而变化，例中建立一个温度计，用一个矩形表示水银柱（将其设置“缩放连接”动画连接属性），以反映变量“温度”的变化。左图是设计状态，右图是在 Touchview 中的运行状态。



图7.33 缩放连接实例

缩放连接的设置方法是：在“动画连接”对话框中单击“缩放连接”按钮，弹出如下图所示的对话框。



图7.34 缩放连接

对话框中各项设置的意义如下：

- ◇ **表达式:** 在此编辑框内输入合法的连接表达式, 单击“?”按钮可以查看已定义的变量名和变量域。
- ◇ **最小时:** 输入对象最小时占据的被连接对象的百分比(占据百分比)及对应的表达式的值(对应值)。百分比为0时此对象不可见。
- ◇ **最大时:** 输入对象最大时占据的被连接对象的百分比(占据百分比)及对应的表达式的值(对应值)。若此百分比为100, 则当表达式值为对应值时, 对象大小为制作时该对象大小。
- ◇ **变化方向:** 选择缩放变化的方向。变化方向共有五种, 用“方向选择”按钮旁边的指示器来形象地表示。箭头是变化的方向, 蓝点是参考点。单击“方向选择”按钮, 可选择五种变化方向之一。



向下变化 向上变化 向中心变化 向左变化 向右变化

图7.35 变化方向

7.3.7 旋转连接

旋转连接是使对象在画面中的位置随连接表达式的值而旋转。



例如:

建立了一个有指针仪表, 以指针旋转的角度表示变量“泵速”的变化。左图是设计状态, 右图是在 Touchview 中的运行状态。

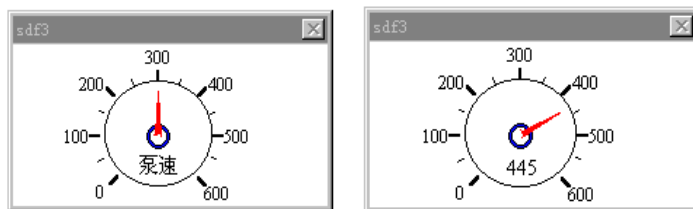


图7.36 旋转连接实例

旋转连接的设置方法为：在“动画连接”对话框中单击“旋转连接”按钮，弹出对话框：

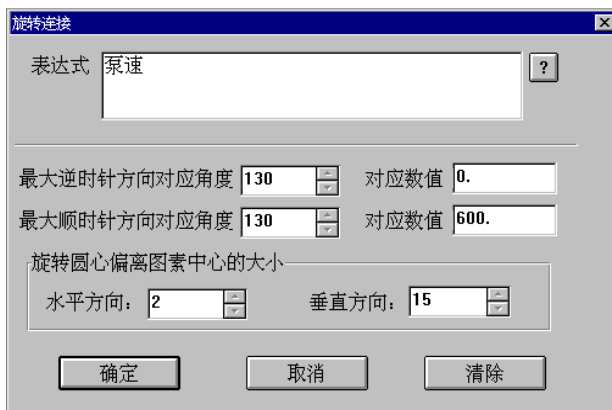


图7.37 旋转连接

对话框中各项设置的意义如下：

- ✧ **表达式：**在此编辑框内输入合法的连接表达式，单击“？”按钮可以查看已定义的变量名和变量域。
- ✧ **最大逆时针方向对应角度：**被连接对象逆时针方向旋转所能达到的最大角度及对应的表达式的值（对应数值）。角度值限于0度至360度之间，Y轴正向是0度。

- ◇ **最大顺时针方向对应角度：**被连接对象顺时针方向旋转所能达到的最大角度及对应的表达式的值（对应数值）。角度值限于 0 度至 360 度之间，Y 轴正向是 0 度。
- ◇ **旋转圆心偏离图素中心的大小：**被连接对象旋转时所围绕的圆心坐标距离被联接对象中心的值，水平方向为圆心坐标水平偏离的像素数（正值表示向右偏离），垂直方向为圆心坐标垂直偏离的像素数（正值表示向下偏离），该值可由坐标位置窗口（在组态王开发系统中用热键 F8 激活）帮助确定。

7.3.8 填充连接

填充连接是使被连接对象的填充物（颜色和填充类型）占整体的百分比随连接表达式的值而变化。



例如：

建立一个矩形对象，以表示变量“液位”的变化。左图是设计状态，右图是在 Touchview 中的运行状态。



图7.38 填充连接实例

填充连接的设置方法是：在“动画连接”对话框中单击“填充连接”按钮，弹出的对话框如下图：



图7.39 填充连接

对话框中各项设置的意义如下：

- ◇ **表达式**：在此编辑框内输入合法的连接表达式，单击“？”按钮可以查看已有的变量名和变量域。
- ◇ **最小填充高度**：输入对象填充高度最小时所占据的被连接对象的高度（或宽度）的百分比（占据百分比）及对应的表达式的值（对应数值）。
- ◇ **最大填充高度**：输入对象填充高度最大时所占据的被连接对象的高度（或宽度）的百分比（占据百分比）及对应的表达式的值（对应数值）。
- ◇ **填充方向**：规定填充方向,由“填充方向”按钮和填充方向示意图两部分组成。共有4种填充方向，单击“填充方向”按钮，可选其中之一。

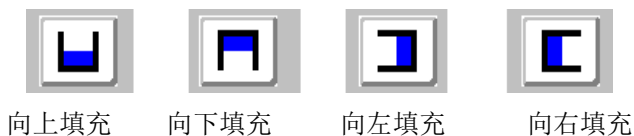
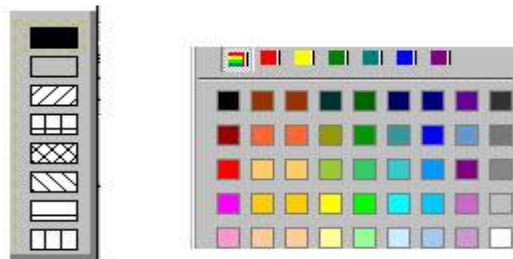


图7.40 填充方向


- ◇ **缺省填充刷**：若本连接对象没有填充属性连接。则运行时用此缺省填充刷。按鼠标左键击中“类型”按钮弹出漂浮式窗口，移动鼠标进行选择；也可以使“类型”按钮获得输入焦点，按空格键弹出浮动窗口，用 TAB 键在颜色和填充类型间切换，用移动键选择，按空格键或回车结束选择。按鼠标左键击中“颜色”按钮弹出漂浮式窗口，移动鼠标进行选择。



类型

颜色

图7.41 缺省填充刷

 **注意：区别填充和缩放**

填充指在图形对象内部以某种颜色，向某方向，均匀的充满整个图形对象；

缩放指将图形对象向某方向整体放大或缩小。见下图。

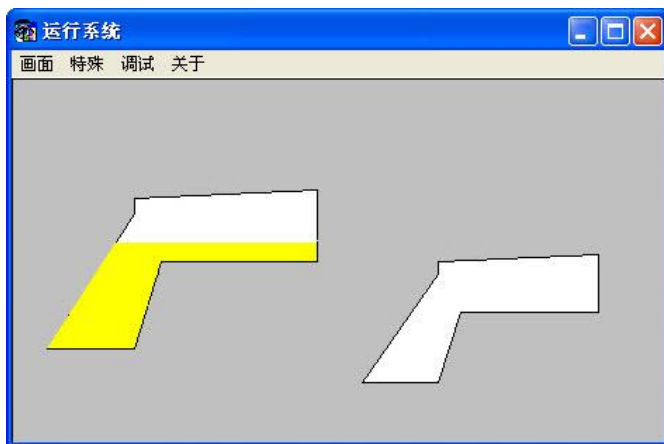


图7.42 填充和缩放

上图中，左为填充连接，右为缩放连接。

7.3.9 模拟值输出连接

模拟值输出连接是使文本对象的内容在程序运行时被连接表达式的值所取代。

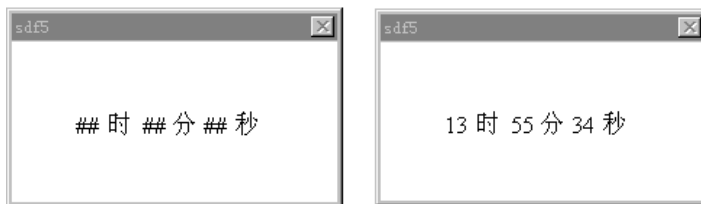


图7.43 模拟值输出实例



例如：

建立文本对象以表示系统时间。为文本对象连接的变量是系统预定义变量\$时、\$分、\$秒。左图是设计状态，右图是在 Touchview 中的运行状态。

模拟值输出连接的设置方法是：在“动画连接”对话框中单击“模拟值输出”按钮，弹出对话框：



图7.44 模拟值输出连接

对话框中各项设置的意义如下：

- ◇ **表达式**：在此编辑框内输入合法的连接表达式，单击右侧的“？”可以查看已定义的变量名和变量域。
- ◇ **整数位数**：输出值的整数部分占据的位数，若实际输出时的值的位数少于此处输入的值，则高位填0。如：规定整数位是4位，而实际值是12，则显示为0012。如果实际输出的值位数多于此值，则按照实际位数输出，实际值是12345，则显示为12345。若不想有前补零的情况出现，则可令整数位数为0。
- ◇ **小数位数**：输出值的小数部分位数。若实际输出时值的位数小于此值，则填0补充。如：规定小数位是4位，而实际值是0.12，则显示为0.1200。如果实际值输出的值

位数多于此值，则按照实际位数输出。

- ◇ **对齐方式**：运行时输出的模拟值字符串与当前被连接字符串在位置上按照左、中、右方式对齐。
- ◇ **十进制**：规定输出值是否用十进制显示。
- ◇ **十六进制**：规定输出值是否用十六进制显示。
- ◇ **科学计数法**：规定输出值是否用科学计数法显示。

7.3.10 离散值输出连接

离散值输出连接是使文本对象的内容在运行时被连接表达式的指定字符串所取代。



例如：

建立一个文本对象“液位状态”，使其内容在变量“液位”的值小于 180 时是“液位正常”，当变量值不小于 180 时，文本对象变为“液位过高”。左图是设计状态，右图是在 Touchview 中的运行状态。



图7.45 离散值输出连接实例

离散值输出连接的设置方法是：在“动画连接”对话框中单击“离散值输出”按钮，弹出对话框：



图7.46 离散值输出连接

对话框中各项设置的意义如下：

- ◇ **条件表达式：**可以输入合法的连接表达式。单击右侧的“？”按钮可以查看已定义的变量名和变量域。
- ◇ **表达式为真时，输出信息：**规定表达式为真时，被连接对象（文本）输出的内容。
- ◇ **表达式为假时，输出信息：**规定表达式为假时，被连接对象（文本）输出的内容。
- ◇ **对齐方式：**运行时输出的离散量字符串与当前被连接字符串在位置上按照左、中、右方式对齐。

7.3.11 字符串输出连接

字符串输出连接是使画面中文本对象的内容在程序运行时被数据库中的某个字符串变量的值所取代。

 例如：

建立文本对象“#####”，使其在运行时输出历史趋势曲线窗口中曲线 1、2 对应的变

量名。为取得此变量名，使用了系统函数 HTGetPenName，函数的用法可参阅第十章“命令语言程序”。左图是设计状态，右图是在 Touchview 中的运行状态。



图7.47 字符串输出连接实例

字符串输出连接的设置方法是：在“动画连接”对话框中单击“字符串输出”按钮，弹出对话框：

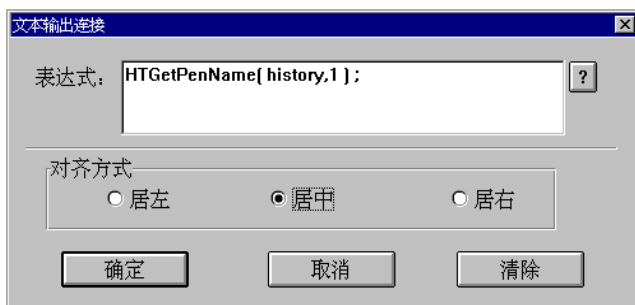


图7.48 字符串输出连接

对话框中各项设置的意义是：

- ✧ **表达式：**输入要显示值内容的字符串变量。单击右侧的“？”按钮可以查看已定义的变量名和变量域。
- ✧ **对齐方式：**选择运行时输出的字符串与当前被连接字符串在位置上的对齐方式。

7.3.12 模拟值输入连接

模拟值输入连接是使被连接对象在运行时为触敏对象，单击此对象或按下指定热键将弹出输入值对话框，用户在对话框中可以输入连接变量的新值，以改变数据库中某个模拟型变量的值。



例如：

建立一个矩形框，设置“模拟值输入”连接以改变变量“温度”的值。下图是在组态王开发系统中的设计状态。

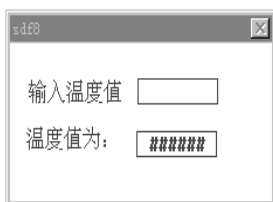


图7.49 模拟值输入连接实例

在运行时单击矩形框，弹出输入对话框：



图7.50 模拟值输入连接

用户在此对话框中可以输入变量的新值。如果在组态王工程浏览器中选中了“系统配置\设置运行系统”下的“特殊”属性页中的“使用虚拟键盘”选项，程序运行中弹出输入对话框的同时还将显示模拟键盘窗口，在模拟键盘上单击按钮的效果与键盘输入相同。

模拟值输入连接的设置方法是：在“动画连接”对话框中单击“模拟值输入”按钮，弹出对话框：

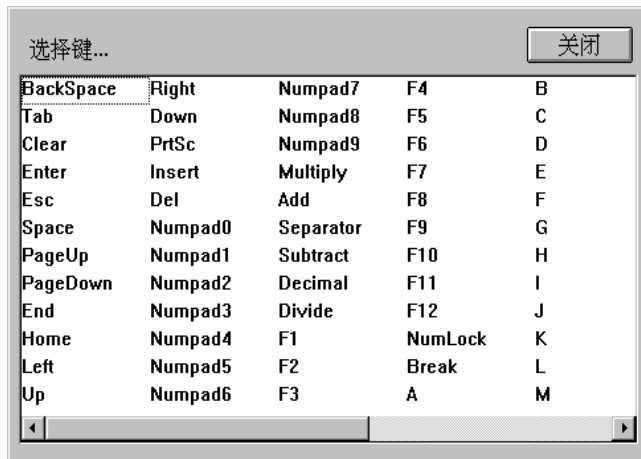


图7.51 模拟值输入连接设置

对话框中各项设置的意义是：

- ◇ **变量名**：要改变的模拟类型变量的名称。单击右侧的“？”按钮可以查看已定义的变量和变量域。
- ◇ **提示信息**：运行时出现在弹出对话框上用于提示输入内容的字符串。
- ◇ **值范围**：规定键入值的范围。它应该是要改变的变量在数据库中设定的最大值和最小值。
- ◇ **激活键**：定义激活键，这些激活键可以是键盘上的单键也可以是组合键（Ctrl, Shift 和键盘单键的组合），在 Touchview 运行画面时可以用激活键随时弹出

输入对话框，以便输入修改新的模拟值。当 **Ctrl** 和 **Shift** 字符左边的选择框中出现“✓”符号时，分别表示 **Ctrl** 键和 **Shift** 键有效，单击“键”按钮，则弹出如下所示的对话框：



在此对话框中用户可以选择一个键，再单击“关闭”按钮完成热键设置。

7.3.13 离散值输入连接

离散值输入连接是使被连接对象在运行时为触敏对象，单击此对象后弹出输入值对话框，可在对话框中输入离散值，以改变数据库中某个离散类型变量的值。



例如：

建立一个矩形框对象，与之连接的变量是 DDE 离散变量“电源开关”。下图是在组态王开发系统中的设计状态。运行时单击矩形对象，弹出右图所示输入对话框：

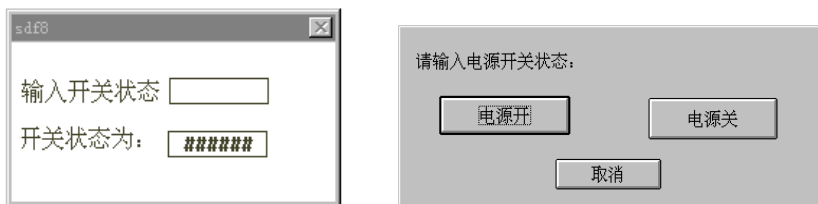


图7.52 离散值输入连接实例

在对话框中单击适当的按钮可以改变离散变量“电源开关”的值。

离散值输入连接的设置方法是：在“动画连接”对话框中单击“离散值输入”按钮，弹出对话框：

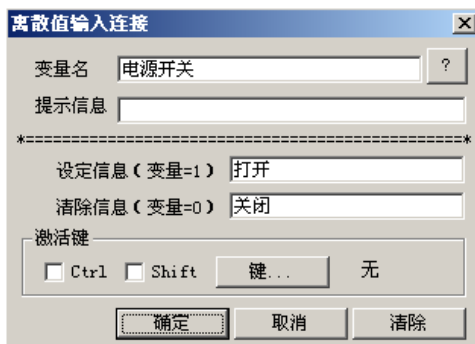


图7.53 离散值输入连接

对话框中各项设置的意义如下：

- ◇ **变量名：**要改变的离散类型变量的名称。单击右侧的“？”按钮可以查看已定义的变量和变量域。
- ◇ **提示信息：**运行时出现在弹出对话框上用于提示输入内容的字符串。
- ◇ **设置信息：**运行时出现在弹出对话框上第一个按钮上的文本内容，此按钮用于将离

散变量值设为 1。

- ✧ **清除信息：**运行时出现在弹出对话框上第二个按钮上的文本内容，此按钮用于将离散变量值设为 0。
- ✧ **激活键：**定义激活键，这些激活键可以是键盘上的单键也可以是组合键（Ctrl、Shift 和键盘单键的组合），在 Touchview 运行画面时可以用激活键随时弹出输入对话框，以便输入修改新的离散值。当“Ctrl”和“Shift”字符左边出现“✓”符号时，分别表示 Ctrl 和 Shift 键有效，单击“键”按钮，弹出如下所示的对话框：

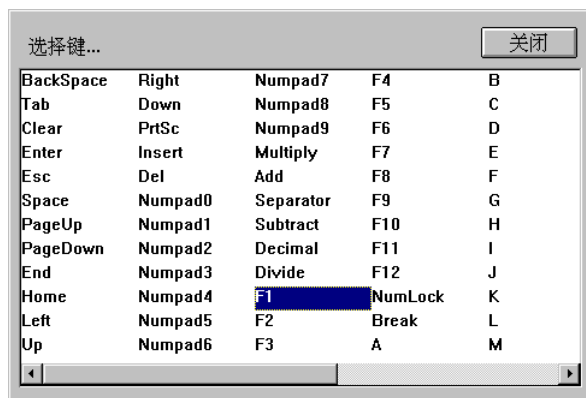


图7.54 定义激活键

在此对话框中可以选择一个键作为热键，再单击“关闭”按钮完成设置。

7.3.14 字符串输入连接

字符串输入连接是使被连接对象在运行时为触敏对象，用户可以在运行时改变数据库中的某个字符串类型变量的值。



图7.55 字符串输入连接实例



例如：

建立一个矩形框对象。使其能够输入内存字符串变量“记录信息”的值。左图是在组态王开发系统中的设计状态。运行时单击触敏对象，弹出输入对话框，如下图 7.56 所示。

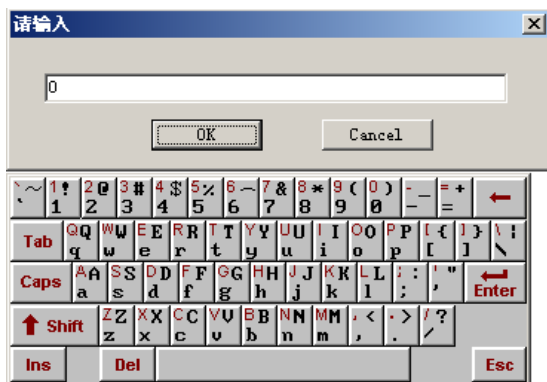


图7.56 字符串输入连接

“字符串输入”动画连接的设置方法是：选择连接对话框中的“字符串输入”按钮，弹出对话框：

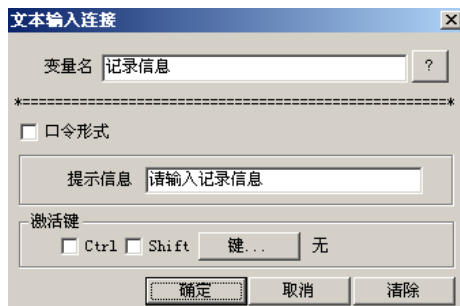


图7.57 字符串输入连接设置

对话框中各项设置的意义是：

- ◇ **变量名：**要改变的字符串类型变量的名称。单击“？”按钮可以查看已定义的变量和变量域。
- ◇ **提示信息：**运行时出现在弹出对话框上用于提示输入内容的字符串。
- ◇ **口令形式：**规定用户在向弹出对话框上的编辑框中键入字符串内容时，编辑框中的字符是否以口令形式（*****）显示。
- ◇ **激活键：**定义激活键,这些激活键可以是键盘上的单键也可以是组合键(Ctrl,Shift 和键盘单键的组合),在 Touchview 运行画面时可以用激活键随时弹出输入对话框,以便输入修改新的字符串值。当“Ctrl”和“Shift”字符左边出现“✓”符号时，分别表示 Ctrl 和 Shift 键有效，单击“键”按钮，弹出如下对话框：

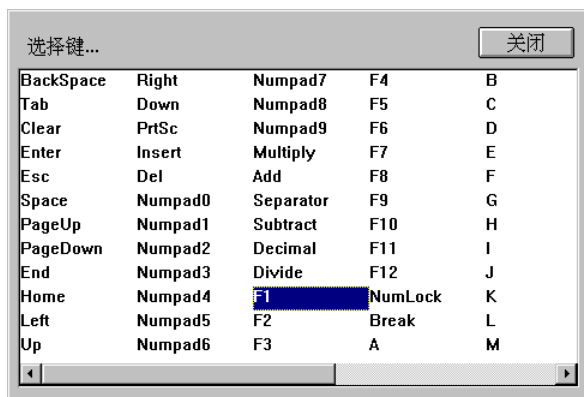


图7.58 定义激活键

在此对话框中可以选择一个键作为热键，再单击“关闭”按钮则完成设置。

7.3.15 闪烁连接

闪烁连接是使被连接对象在条件表达式的值为真时闪烁。闪烁效果易于引起注意，故常用于出现非正常状态时的报警。



图7.59 闪烁连接实例



例如：

建立一个表示报警状态的红色圆形对象，使其能够在变量“液位”的值大于180时闪烁。右图是在组态王开发系统中的设计状态。运行中当变量“液位”的值大于180时，红

色对象开始闪烁。闪烁连接的设置方法是：在“动画连接”对话框中单击“闪烁”按钮，弹出对话框：



图7.60 闪烁连接

对话框中各项设置的意义是：

- ◇ **条件表达式：** 输入闪烁的条件表达式，当此条件表达式的值为真时，图形对象开始闪烁。表达式的值为假时闪烁自动停止。单击“？”按钮可以查看已定义的变量名和变量域。
- ◇ **闪烁速度：** 规定闪烁的频率。

7.3.16 隐含连接

隐含连接是使被连接对象根据条件表达式的值而显示或隐含。本例中建立一个表示危险状态的文本对象“液位过高”，使其能够在变量“液位”的值大于180时显示出来。下图是在组态王开发系统中的设计状态。



图7.61 隐含连接实例

隐含连接的设置方法是：在“动画连接”对话框中单击“隐含”按钮，弹出对话框：



图7.62 隐含连接

对话框中各项设置的意义是：

- ✧ **条件表达式：**输入显示或隐含的条件表达式，单击“？”可以查看已定义的变量名和变量域。
- ✧ **表达式为真时：**规定当条件表达式值为 1 (TRUE) 时，被连接对象是显示还是隐含。当表达式的值为假时，定义了“显示”状态的对象自动隐含，定义了“隐含”状态的对象自动显示。

7.3.17 流动连接

流动连接用于设置立体管道内液体流线的流动状态。流动状态根据“流动条件”表达式的值确定。

在画面上绘制立体管道（立体管道的属性设置请参见“第四章 组态王开发环境 - 工程浏览器”）。鼠标双击该管道，在“动画连接”对话框中单击“流动”按钮，弹出管道流动连接对话框，如图 7.63 所示。

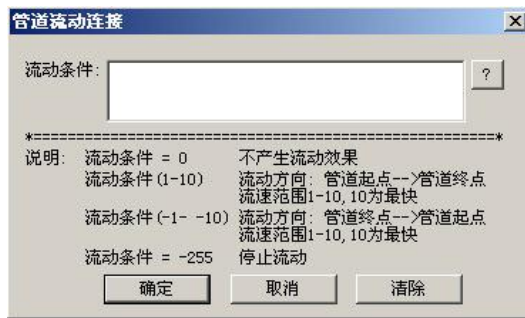


图7.63 流动连接

◇ “流动条件”：输入流动状态关联的组态王变量，应为整型变量。单击“？”可以选择已定义的变量名。管道流动的状态由关联的变量的值确定：

- 当变量值为 0 时，不产生流动效果，管道内不显示流线；
- 当变量值在(1,10)范围时，管道内液体流线的流动方向为管道起点至管道终点，流速为设定值，10 为速度的最大值；
- 当变量值为(-10, -1)时，管道内液体流线的流动方向为管道终点至管道起点，流速为设定值，-10 为速度的最大值；
- 当变量值为 0 时，停止流动，管道内显示静止的流线。

管道流动速度与组态王运行系统基准频率有关。当组态王运行系统的基准频率设置值大时，管道显示流动速度慢，否则快。

7.3.18 水平滑动杆输入连接

当有滑动杆输入连接的图形对象被鼠标拖动时，与之连接的变量的值将会被改变。当变量的值改变时，图形对象的位置也会发生变化。



例如：

建立一个用于改变变量“泵速”值的水平滑动杆。左图是设计状态，右图是在 Touchview 中的运行状态。

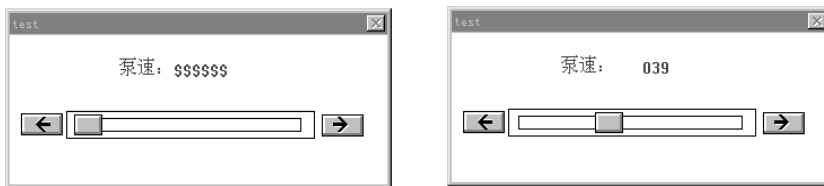


图7.64 水平滑动杆输入连接实例

水平滑动杆输入连接的方法是：在“动画连接”对话框中单击“水平滑动杆输入”按钮，弹出对话框：



图 7.65 水平滑动杆输入连接

对话框中各项设置的意义是：

- ◇ **变量名：**输入与图形对象相联系的变量，单击“？”可以查看已定义的变量名和变量域。
- ◇ **向左：**图形对象从设计位置向左移动的最大距离。
- ◇ **向右：**图形对象从设计位置向右移动的最大距离。
- ◇ **最左边：**图形对象在最左端时变量的值。
- ◇ **最右边：**图形对象在最右端时变量的值。

7.3.19 垂直滑动杆输入连接

垂直滑动杆输入连接与水平滑动杆输入连接类似，只是图形对象的移动方向不同。设置方法是：在“动画连接”对话框中单击“垂直滑动杆输入”按钮，弹出对话框：



图 7.66 垂直滑动杆输入连接

对话框中各项的意义解释如下：

- ◇ **变量名：**与产生滑动输入的图形对象相联系的变量。单击“？”按钮查看所有已定义的变量名和变量域。
- ◇ **向上：**图形对象从设计位置向上移动的最大距离。
- ◇ **向下：**图形对象从设计位置向下移动的最大距离。
- ◇ **最上边：**图形对象在最上端时变量的值。
- ◇ **最下边：**图形对象在最下端时变量的值。

7.3.20 动画连接命令语言

命令语言连接会使被连接对象在运行时成为触敏对象。当 Touchview 运行时，触敏对象周围出现反显的矩形框。命令语言有五种：“按下时”、“弹起时”、“按住时”、“鼠标进入”和“鼠标离开”，分别表示鼠标左键在触敏对象上按下、弹起、按住时、进入、离开

时，执行连接的命令语言程序。定义“按住时”的命令语言连接时，还可以指定按住鼠标后每隔多少毫秒执行一次命令语言，这个时间间隔在编辑框内输入。可以指定一个等价键，工程人员在键盘上用等价键代替鼠标，等价键的按下、弹起、按住三种状态分别等同于鼠标的按下、弹起、按住状态。单击任一种“命令语言连接”按钮，将弹出对话框用于输入命令语言连接程序。

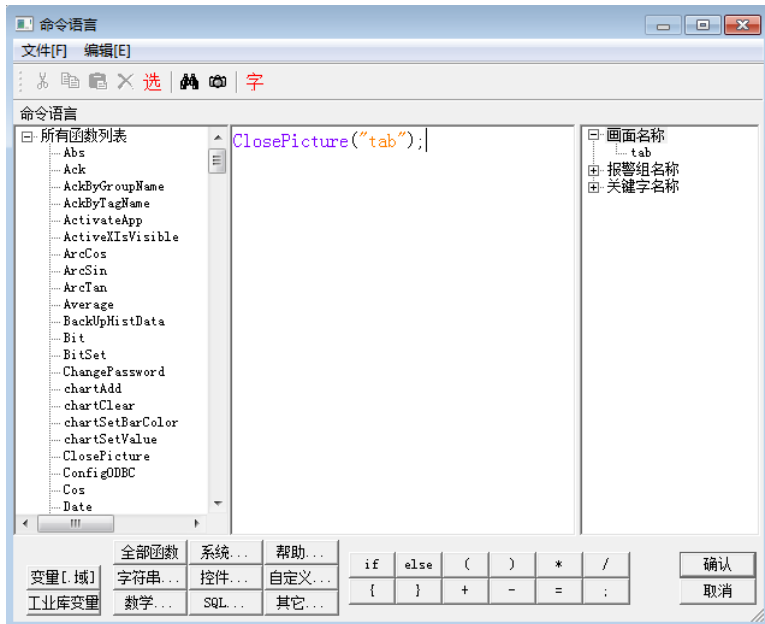


图7.67 命令语言

在对话框右边有一些能产生提示信息的按钮，可让用户选择已定义的变量名及域，系统预定义函数名，画面窗口名，报警组名，算符，关键字等。还提供剪切、复制、粘贴、复原等编辑手段，使用户可以从其它命令语言连接中复制已编好的命令语言程序。详细内容参见“第十章 命令语言程序”。

7.4 动画连接向导的使用

组态王提供可视化动画连接向导供用户使用。该向导的动画连接包括：水平移动、垂直移动、旋转、滑动杆水平输入、滑动杆垂直输入等五个部分。使用可视化动画连接向导可以简单、精确地定位图素动画的中心位置、移动起止位置和移动范围等。

7.4.1 水平移动动画连接向导

使用水平移动动画连接向导的步骤为：

- ◇ 首先在画面上绘制水平移动的图素，如圆角矩形。
- ◇ 选中该图素，选择菜单命令“编辑\水平移动向导”，或在该圆角矩形上单击右键，在弹出的快捷菜单上选择“动画连接向导\水平移动连接向导”命令，鼠标形状变为小“十”字型。
- ◇ 选择图素水平移动的起始位置，单击鼠标左键，鼠标形状变为向左的箭头，表示当前定义的是运行时图素由起始位置向左移动的距离，水平移动鼠标，箭头随之移动，并画出一条水平移动轨迹线。
- ◇ 当鼠标箭头向左移动到左边界后，单击鼠标左键，鼠标形状变为向右的箭头，表示当前定义的是运行时图素由起始位置向右移动的距离，水平移动鼠标，箭头随之移动，并画出一条移动轨迹线，当到达水平移动的右边界时，单击鼠标左键，弹出水平移动动画连接对话框。如下图 7.68 所示。

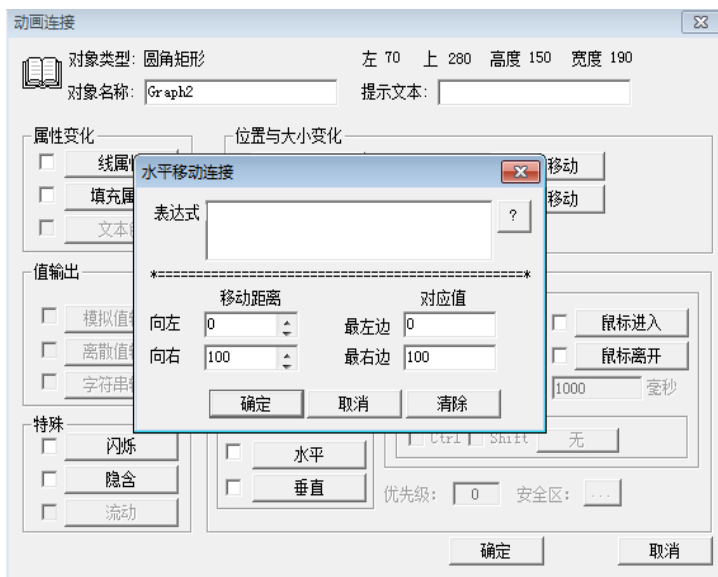


图7.68 水平移动动画连接

在“表达式”文本框中输入变量或单击“？”按钮选择变量。在“移动距离”的“向左”、“向右”文本框中的数据为利用向导建立动画连接产生的数据。用户可以按照需要再修改该项。单击“确定”完成动画连接。

7.4.2 垂直移动动画连接向导

使用垂直移动动画连接向导的步骤为：

- ✧ 首先在画面上绘制垂直移动的图素，如圆角矩形。
- ✧ 选中该图素，选择菜单命令“编辑\垂直移动向导”，或在该圆角矩形上单击右键，在弹出的快捷菜单上选择“动画连接向导\垂直移动连接向导”命令，鼠标形状变为小“十”字型。

- ◇ 选择图素垂直移动的起始位置，单击鼠标左键，鼠标形状变为向上的箭头，表示当前定义的是运行时图素由起始位置向上移动的距离，垂直移动鼠标，箭头随之移动，并画出一条垂直移动轨迹线。
- ◇ 当鼠标箭头向上移动到上边界后，单击鼠标左键，鼠标形状变为向下的箭头，表示当前定义的是运行时图素由起始位置向下移动的距离，垂直移动鼠标，箭头随之移动，并画出一条垂直移动轨迹线，当到达垂直移动的下边界时，单击鼠标左键，弹出垂直移动动画连接对话框。如下图 7.69 所示。

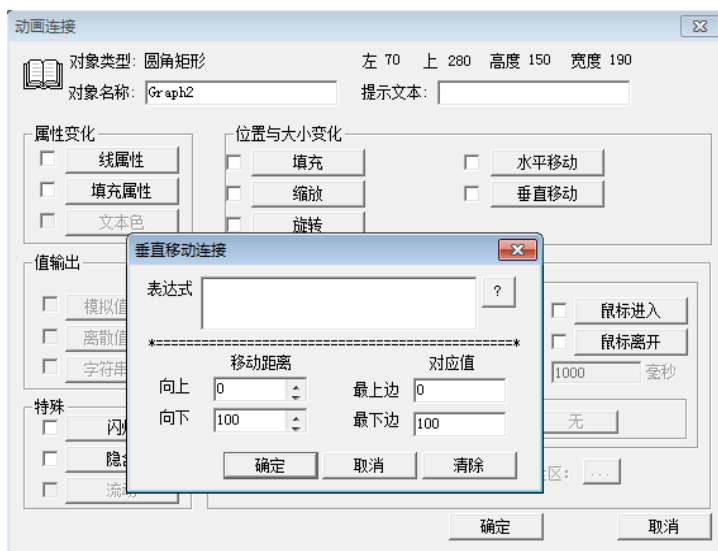


图 7.69 垂直移动动画连接

在“表达式”文本框中输入变量或单击“？”按钮选择变量。在“移动距离”的“向上”、“向下”文本框中的数据为利用向导建立动画连接产生的数据。用户可以按照

需要再修改该项。单击“确定”完成动画连接。

7.4.3 滑动杆输入动画连接向导

滑动杆的水平输入和垂直输入动画连接向导的使用与水平移动、垂直移动动画连接向导的使用方法相同。

7.4.4 旋转动画连接向导

使用旋转动画连接向导的步骤为：

- ◇ 首先在画面上绘制旋转动画的图素，如椭圆。
- ◇ 选中该图素，选择菜单命令“编辑\旋转向导”，或在该椭圆上单击右键，在弹出的快捷菜单上选择“动画连接向导\旋转连接向导”命令，鼠标形状变为小“十”字型。
- ◇ 选择图素旋转时的围绕中心，在画面上相应位置单击鼠标左键。随后鼠标形状变为逆时针方向的旋转箭头，表示现在定义的是图素逆时针旋转的起始位置和旋转角度。移动鼠标，环绕选定的中心，则一个图素形状的虚线框会随鼠标的移动而转动。
- ◇ 确定逆时针旋转的起始位置后，单击鼠标左键，鼠标形状变为顺时针方向的旋转箭头，表示现在定义的是图素顺时针旋转的起始位置和旋转角度，方法同逆时针定义。选定好顺时针的位置后，单击鼠标弹出旋转动画连接对话框。如下图 7.70 所示。

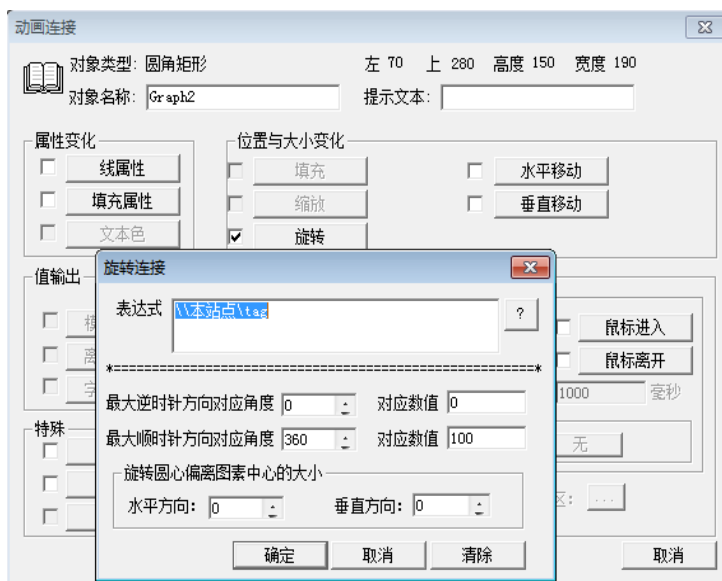


图7.70 旋转连接对话框

旋转连接动画向导很有力地解决用户在定义旋转图素时很难找到旋转中心的问题。

第八章 趋势曲线和其它曲线

- ☞ 教您如何使用历史趋势曲线及实时趋势曲线
- ☞ 教您如何使用温控曲线和超级 X-Y 曲线

8.1 曲线的一般介绍

组态王的实时数据和历史数据除了在画面中以值输出的方式和以报表形式显示外，还可以用曲线的形式显示。组态王的曲线有趋势曲线、温控曲线和超级 X-Y 曲线。

趋势分析是控制软件必不可少的功能，“组态王”对该功能提供了强有力的支持和简单的控制方法。趋势曲线有实时趋势曲线和历史趋势曲线两种。曲线外形类似于坐标纸，X 轴代表时间，Y 轴代表变量值。对于实时趋势曲线最多可显示四条曲线；而历史趋势曲线最多可显示十六条曲线，而一个画面中可定义数量不限的趋势曲线（实时趋势曲线或历史趋势曲线）。在趋势曲线中工程人员可以规定时间间距，数据的数值范围，网格分辨率，时间坐标数目，数值坐标数目，以及绘制曲线的“笔”的颜色属性。画面程序运行时，实时趋势曲线可以自动卷动，以快速反应变量随时间的变化；历史趋势曲线不能自动卷动，它一般与功能按钮一起工作，共同完成历史数据的查看工作。这些按钮可以完成翻页、设定时间参数、启动/停止记录、打印曲线图等复杂功能。

温控曲线反映出实际测量值按设定曲线变化的情况。在温控曲线中，纵轴代表温度值，

横轴对应时间的变化，同时将每一个温度采样点显示在曲线中。主要适用于温度控制，流量控制等等。

超级 X-Y 曲线主要是用曲线来显示两个变量之间的运行关系，例如电流—转速曲线等，支持多 Y 轴曲线。

8.2 实时趋势曲线

组态王提供两种形式的实时趋势曲线：工具箱中的组态王内置实时趋势曲线和实时趋势曲线 Active X 控件。

8.2.1 组态王内置实时趋势曲线

组态王内置实时趋势曲线绘制命令位于组态王工具箱中，最多支持 4 条曲线。

8.2.1.1 创建实时趋势曲线

在组态王开发系统中制作画面时，选择菜单“工具 \ 实时趋势曲线”项或单击工具箱中的“画实时趋势曲线”按钮，此时鼠标在画面中变为十字形，在画面中用鼠标画出一个矩形，实时趋势曲线就在这个矩形中绘出，如下图所示。

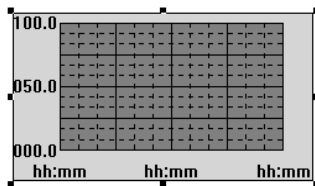


图 8.1 实时趋势曲线

实时趋势曲线对象的中间有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格左方和下方分别是 X 轴（时间轴）和 Y 轴（数值轴）的坐标标注。可以通过选中实时趋势曲线对象（周围出现 8 个小矩形）来移动位置或改变大小。在画面运行时实时趋势曲线对象由系统自动更新。

8.2.1.2 实时趋势曲线属性

用鼠标左键双击创建的实时趋势曲线，弹出实时趋势曲线属性对话框，如下图所示。

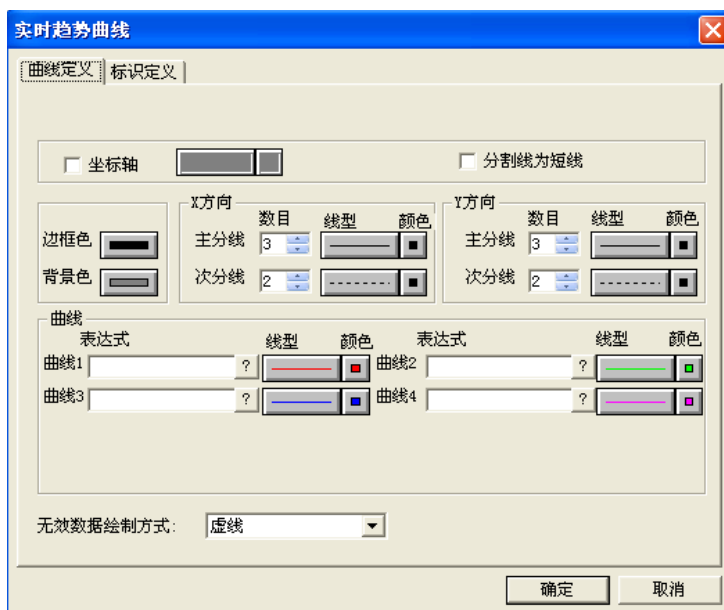


图 8.2 定义实时趋势曲线

属性对话框中各项含义如下：

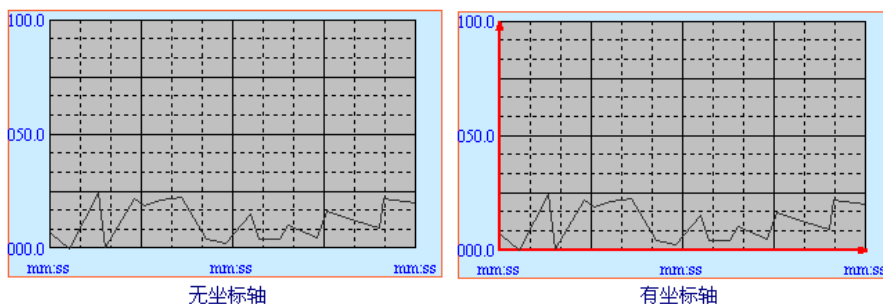
1、曲线定义属性卡片选项：

◇ **坐标轴**：选择曲线图表坐标轴的线形和颜色。选择“坐标轴”复选框后，坐标

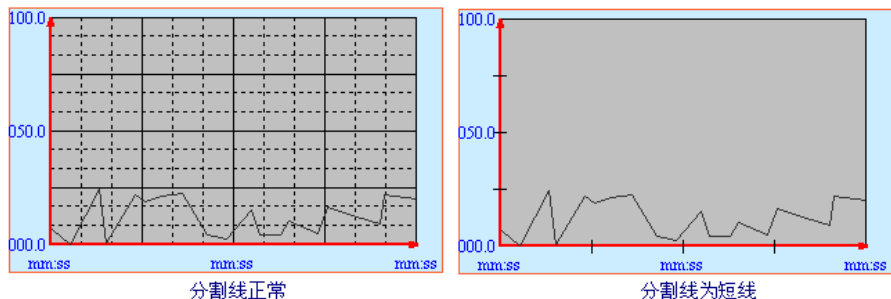
轴的线形和颜色选择按钮变为有效，通过点击线形按钮或颜色按钮，在弹出的列表中选择坐标轴的线形或颜色，如下图所示。



用户可以根据图表绘制需要，选择是否显示坐标轴，如下图所示，为不显示坐标轴和显示坐标轴的结果。



- ✧ **分割线为短线:** 选择分割线的类型。选中此项后在坐标轴上只有很短的主分割线，整个图纸区域接近空白状态，没有网格，同时下面的“次分割线”选择项变灰，图表上不显示次分割线。如下图所示为分割线正常显示和分割线为短线显示结果。



- ◇ **边框色、背景色：**分别规定绘图区域的边框和背景(底色)的颜色。按动这两个按钮的方法与坐标轴按钮类似，弹出的浮动对话框也与之大致相同。

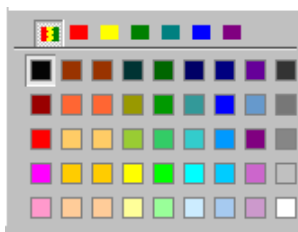


图 8.3 调色板

- ◇ **X方向、Y方向：**X方向和Y方向的主分割线将绘图区划分成矩形网格，次分割线将再次划分主分割线划分出来的小矩形。这两种线都可改变线型和颜色。分割线的数目可以通过小方框右边“加减”按钮增加或减小，也可通过编辑区直接输入。工程人员可以根据实时趋势曲线的大小决定分割线的数目，分割线最好与标识定义(标注)相对应。
- ◇ **曲线：**定义所绘的 1~4 条曲线Y坐标对应的表达式，实时趋势曲线可以实时计算表达式的值，所以它可以使用表达式。实时趋势曲线名的编辑框中可输入

有效的变量名或表达式，表达式中所用变量必需是数据库中已定义的变量。右边的“？”按钮可列出数据库中已定义的变量或变量域供选择。每条曲线可通过右边的线型和颜色按钮来改变线型和颜色。在定义曲线属性时，至少应定义一条曲线变量。

- ◇ **无效数据绘制方式：**在系统运行时对于采样到的无效数据（如变量质量戳≠192）的绘制方式选择。可以选择三种形式：虚线、不画线和实线。

2、标识定义属性卡片选项：

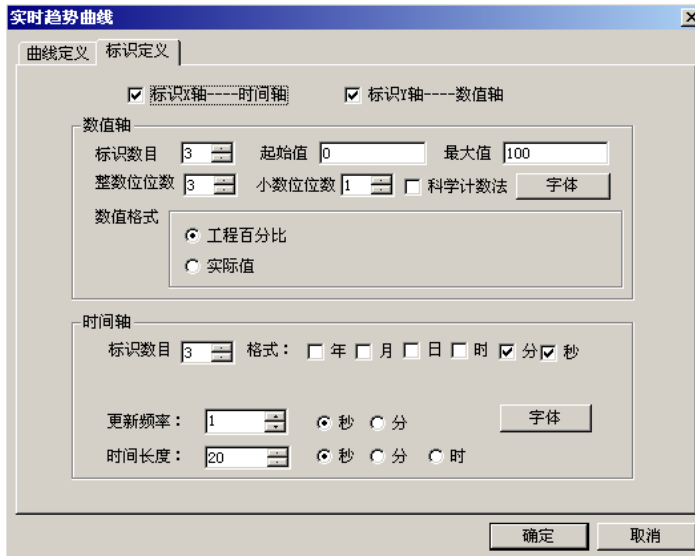


图 8.4 标识定义属性卡片

- ◇ **标识 X 轴—时间轴、标识 Y 轴—数值轴：**选择是否为 X 或 Y 轴加标识，即在绘图区域的外面用文字标注坐标的数值。如果此项选中，左边的检查框中有小叉

标记，同时下面定义相应标识的选择项也由无效变为有效。

◇ **数值轴(Y轴)定义区：**因为一个实时趋势曲线可以同时显示4个变量的变化，而各变量的数值范围可能相差很大，为使每个变量都能表现清楚，“组态王”中规定，变量在Y轴上以百分数表示，即以变量值与变量范围（最大值与最小值之差）的比值表示。所以Y轴的范围是0（0%）至1（100%）。

- **标识数目：**数值轴标识的数目，这些标识在数值轴上等间隔分布。
- **起始值：**曲线图表上纵轴显示的最小值。如果选择“数值格式”为“工程百分比”，规定数值轴起点对应的百分比值，最小为0。如果选择“数值格式”为“实际值”，则可输入变量的最小值。
- **最大值：**曲线图表上纵轴显示的最大值。如果选择“数值格式”为“工程百分比”，规定数值轴终点对应的百分比值，最大为100。如果选择“数值格式”为“实际值”，则可输入变量的最大值。
- **整数位数：**数值轴最少显示整数的位数。
- **小数位数：**数值轴最多显示小数点后面的位数。
- **科学计数法：**数值轴坐标值超过指定的整数和小数位数时用科学计数法显示。
- **字体：**规定数值轴标识所用的字体。可以弹出WINDOWS标准的字体选择对话框，相应的操作工程人员可参阅WINDOWS的操作手册。
- **数值格式：**
工程百分比：数值轴显示的数据是百分比形式。

实际值：数值轴显示的数据是该曲线的实际值。

◇ 时间轴定义区：

- **标识数目：**时间轴标识的数目，这些标识在数值轴上等间隔分布。在组态王开发系统中时间是以 yy:mm:dd:hh:mm:ss 的形式表示，在 TouchView 运行系统中，显示实际的时间。
- **格式：**时间轴标识的格式，选择显示哪些时间量。
- **更新频率：**图表采样和绘制曲线的频率。最小 1 秒。运行时不可修改。
- **时间长度：**时间轴所表示的时间跨度。可以根据需要选择时间单位——秒、分、时，最小跨度为 1 秒，每种类型单位最大值为 8000。
- **字体：**规定时间轴标识所用的字体。与数值轴的字体选择方法相同。

8.2.1.3 为实时趋势曲线建立“笔”

首先使用图素画出笔的形状(一般用多边形即可)，如下图所示，然后定义图素的垂直移动动画连接，可以通过动画连接向导选择实时趋势曲线绘图区域纵轴方向两个顶点，然后用对应的实时曲线变量所用的表达式定义垂直移动连接。

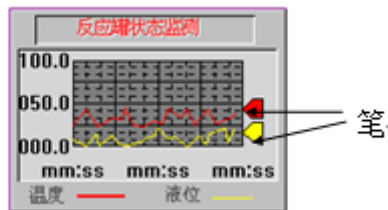


图 8.5 为实时趋势曲线建立“笔”

8.2.2 实时趋势曲线控件

组态王的实时趋势曲线控件具有以下特点：

- ◇ 通过 TCP/IP 获得实时数据，数据服务器可以是任何一台运行组态王的机器，而不需进行组态王网络配置。
- ◇ 最多可以显示 20 条曲线。
- ◇ 可以设置每条曲线的绘制方式，可以为每条曲线设定对照曲线。
- ◇ 可以移动曲线，显示一个采集周期内任意时间段的曲线。
- ◇ 可以保存曲线，加载曲线。
- ◇ 可以打印曲线。

8.2.2.1 创建实时趋势曲线控件

打开组态王画面，在工具箱中单击“插入通用控件”或选择菜单“编辑”下的“插入通用控件”命令，弹出“插入控件”对话框，在列表中选择“CkvrealTimeCurves Control”，单击“确定”按钮，对话框自动消失，鼠标箭头变为小“十”字型，在画面上选择控件的左上角，按下鼠标左键并拖动，画面上显示出一个虚线的矩形框，该矩形框为创建后的曲线的外框。当达到所需大小时，松开鼠标左键，则实时曲线控件创建成功，画面上显示出该曲线，如图 8.6 所示。

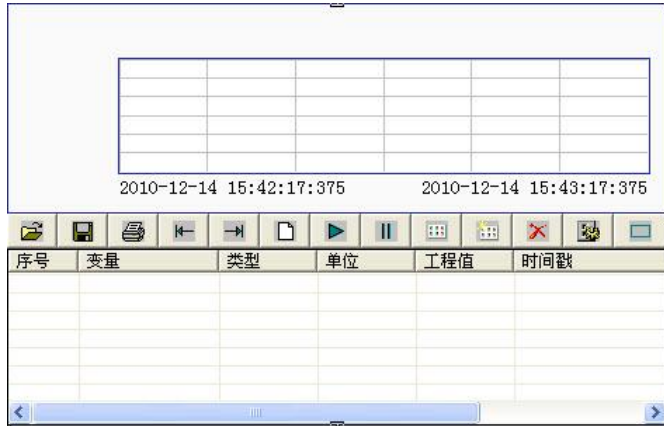


图 8.6 创建实时曲线控件

8.2.2.2 实时曲线控件的属性设置

实时曲线控件创建完成后，在控件上单击右键，在弹出的快捷菜单中选择“控件属性”命令，弹出实时曲线控件的属性设置对话框，如图 8.7 所示。



图 8.7 实时曲线控件属性

实时曲线控件包括两个属性页：“常规”属性页和“曲线”属性页。下面详细介绍属性页中每一项的具体的含义。

1、“常规”属性页：

如图 8.7 所示，常规属性页中包括：

◇ 绘制范围设置

- 距左边：实时曲线绘图区距离控件左边界的距离，设置范围 0—200，
- 距右边：实时曲线绘图区距离控件右边界的距离，设置范围 0—200。
- 距上边：实时曲线绘图区距离控件上边界的距离，设置范围 25—200。
- 距下边：实时曲线绘图区距离控件工具条显示区的距离，设置范围 25—200。

以上设置范围的单位为像素。

◇ 标注字体设置

点击“选择字体”按钮，设置标注字体。

◇ 时间跨度设置

- 显示时间跨度：曲线图表时间轴长度。设置范围是 1—100000，单位 s。
- 采集时间跨度：每次绘制一屏曲线总的时间长度。设置范围是 1—100000，单位 s。

采集时间跨度可以大于显示时间跨度。当绘制的曲线超出图表显示时，可以使用图表工具条上的左移、右移按钮移动查看曲线。

◇ 采集结束时，曲线的处理方式

到达采集结束时间，自动清空数据重新开始：采集结束时间即为一个采集周期（即时间跨度设置中的“采集时间跨度”）的结束时间，时间到，系统自动清空整个采集周期内的曲线数据，曲线重新开始绘制。到达采集结束时间，自动暂停：到达采集结束时间，系统暂停曲线数据采集，绘图区显示停止时刻的曲线。

◇ 服务端 IP

要绘制曲线的变量所在的数据服务器的 IP 地址，如果是本机变量，则输入本机 IP 地址。该控件的使用在组态王“单机”模式下即可。

2、“曲线”属性页：

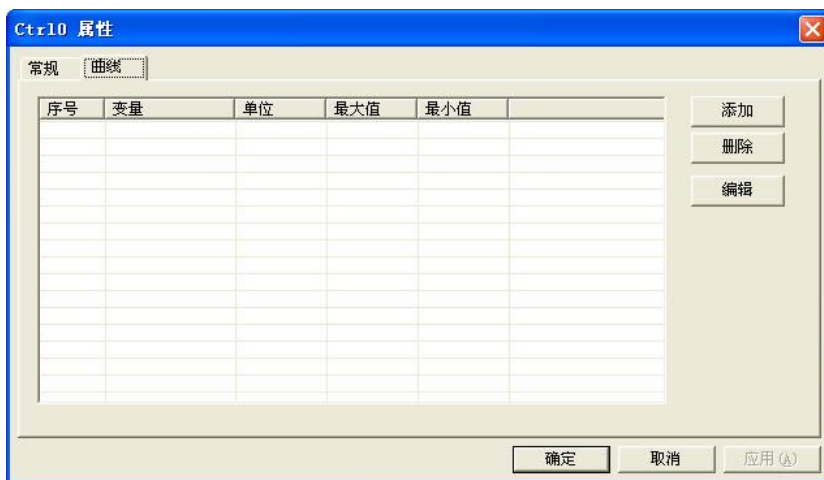


图 8.8 “曲线”属性页

在“曲线”属性页中进行曲线添加，编辑和删除的操作。点击“添加”按钮，显示“新增加曲线”对话框，如图 8.9 所示。



图 8.9 “新增加曲线”属性页

◇ 曲线绘点设置

● 绘点

选择曲线是否以绘点方式绘制，选中状态表示绘制数据点。绘点是指控件在绘制曲线时，在曲线上对每个数据点根据选择的点的颜色和线宽做标记点，如下图所示。



● 仿真时间戳

选择该项则曲线上增加描绘的数据点，仿真数据在曲线上的描点频率为本地运行系统基准频率（即仿真时间戳的变化频率）。该选项主要适用于数据长时间不变的变量曲线。

- **选择点颜色**

点击该按钮选择标记数据点的显示颜色。

- **点线宽**

标记数据点的大小，设置范围为 1—6。

- ◇ **曲线绘线设置**

- **绘线**

选择曲线是否以绘点方式绘制，选中状态表示绘线。

- **连线方式**

选择曲线的连线方式，包括模拟方式和阶梯方式。

- **选择线颜色**

点击该按钮选择曲线的显示颜色。

- **线宽**

曲线宽度，设置范围 1—6。

- ◇ **Y 轴最大值最小值**

- **最小值，最大值**

设置 Y 轴的最小值和最大值。

- **保留小数点位数**

Y 轴坐标显示的小数点位数。

◇ 曲线对应变量

- **变量名**

曲线关联的变量名。

- **变量单位**

为变量设置单位名称，可以为空。

◇ 对照曲线

可以为每条曲线设置对照曲线。

- **显示对照曲线**

选择是否显示曲线的对照曲线，选中状态表示显示。

- **选择线颜色**

点击该按钮选择对照曲线的显示颜色。

- **线宽**

对照曲线的宽度，设置范围 1—6。

- **编辑对照曲线**

点击该按钮，设置对照曲线。如图 8.10 所示。

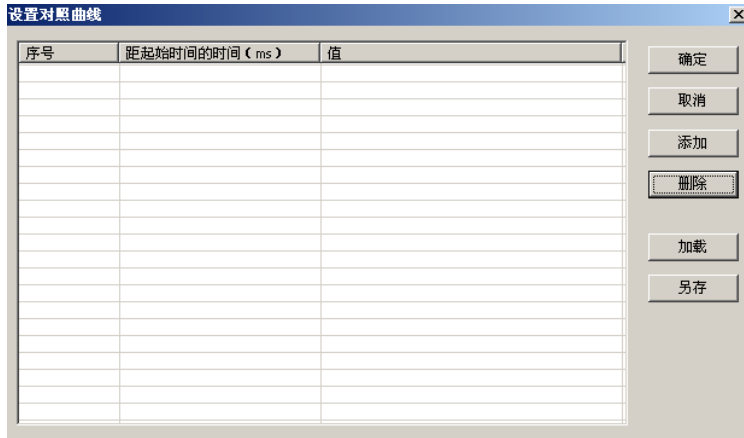


图 8.10 设置对照曲线

添加： 点击“添加”按钮，在对照曲线数据点列表中添加一数据点。设置该数据点距采集起始时间的的时间，和该数据点的值。如图 8.11 所示。

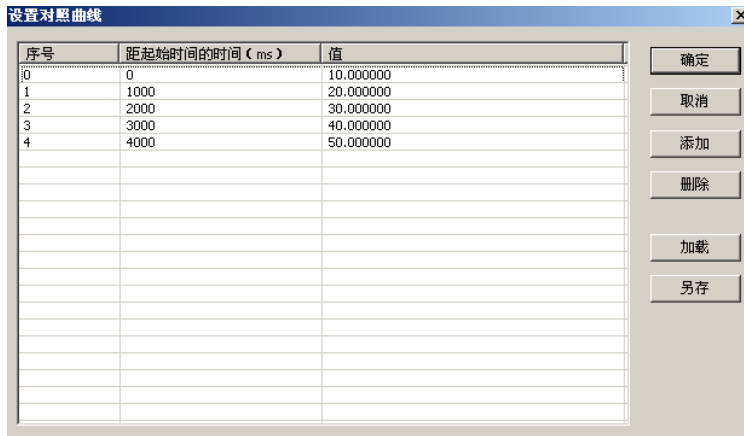


图 8.11 添加对照曲线的数据点

删除： 在对照曲线数据点列表中选择某个数据点或某些数据点，点击“删

除”按钮，删除选择的数据点。

另存：将设置的对照曲线数据保存到.CMP文件中。点击该按钮将显示“另存为”对话框。

加载：将保存过的.CMP文件数据加载到对照曲线数据点列表中。

8.2.2.3 运行时修改实时曲线属性

实时曲线属性定义完成后，进入组态王运行系统，运行系统的实时曲线如图 8.12 所示。

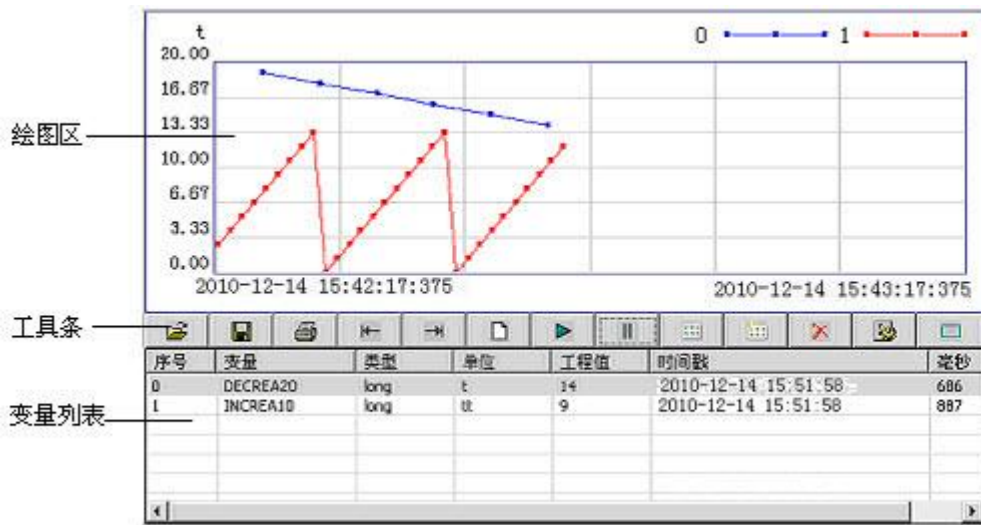


图 8.12 运行系统的实时曲线显示

◇ 绘图区

显示实时趋势曲线和它们的对照曲线。在绘图区最多可以显示 20 条实时曲线。

用鼠标左键点击绘图区不放，左右拖动鼠标，可以使曲线左右平移。

◇ 变量列表区

显示绘图区每条曲线关联的组态王变量信息。

绘图区的每条曲线都有自己的 Y 轴，在变量列表中选中哪个变量，绘图区就显示哪个变量曲线的 Y 轴。如图 8.11 所示，开发系统实时控件属性中设置变量列表中序号为 0 的变量曲线的 Y 轴最大值为 20，最小值为 0，序号为 1 的变量曲线的 Y 轴最大值为 15，最小值为 0。在变量列表中选中序号为 0 的变量，则绘图区的 Y 轴按照最大值为 20，最小值为 0 显示。

◇ 工具条

由具有不同功能的按钮组成，工具条的具体作用可以通过将鼠标放到按钮上时弹出的提示文本中看到。下面详细介绍每个按钮的作用。



：开始曲线绘制。组态王运行系统运行时，默认不绘制曲线。



：停止曲线绘制。





：设置曲线属性。在变量列表区选择某个曲线变量，点击该按钮，弹出“设置曲线”对话框，在该对话框中修改曲线属性。如图 8.13 所示。



图 8.13 设置曲线属性

该对话框中各项的设置方法与开发系统相同，参见 8.2.2.2 实时曲线控件的属性设置。

：添加新曲线。点击该按钮弹出如图 8.8 所示对话框。

：删除曲线。在变量列表区选择某个或某几个曲线变量，点击该按钮，弹出下面的对话框：

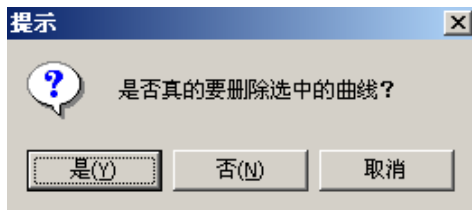


图 8.14 曲线删除确认

如果确认要删除曲线，点击“是(Y)”，所选曲线被删除。



：绘图设置。点击该按钮，弹出如图 8.15 所示对话框。

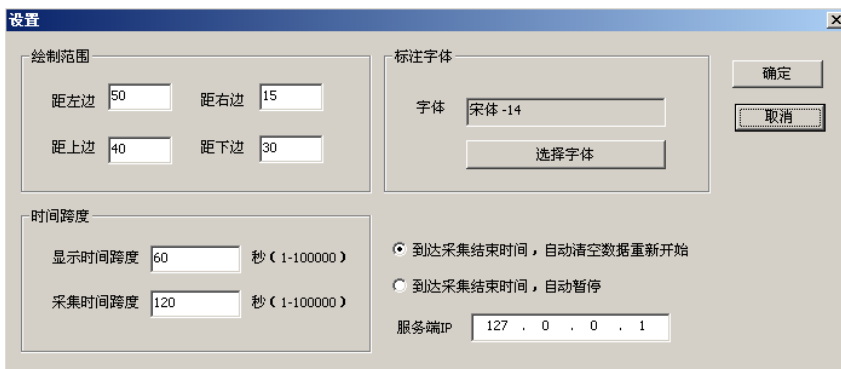


图 8.15 绘图设置

在该对话框中可以设置绘制范围，时间跨度等属性，该对话框中各项的设置方法与开发系统相同，参见 8.2.2.2 实时曲线控件的属性设置。



：清除绘图区所有数据。清除数据之前必须先暂停数据采集。



：显示一个采集周期内时间起点的的数据，在执行该按钮之前请先停止数据采集。



：显示一个采集周期内采集终止时的数据。



：隐藏/显示变量列表。



：打印曲线。



：保存曲线到.CRV 文件中。点击该按钮将显示“另存为”对话框。



：打开已保存的.CRV 文件, 将保存过的曲线加载到绘图区。

8.3 历史趋势曲线

组态王提供三种形式的历史趋势曲线：

- ◇ 第一种是从图库中调用已经定义好各功能按钮的历史趋势曲线，对于这种历史趋势曲线，用户只需要定义几个相关变量，适当调整曲线外观即可完成历史趋势曲线的复杂功能，这种形式使用简单方便；该曲线控件最多可以绘制 8 条曲线，但该曲线无法实现曲线打印功能。
- ◇ 第二种是调用历史趋势曲线控件，对于这种历史趋势曲线，功能很强大，使用比较简单。通过该控件，不但可以实现组态王历史数据的曲线绘制，还可以实现工业库中历史数据的曲线绘制、ODBC 数据库中记录数据的曲线绘制。在运行状态下，可以实现在线动态增加/删除曲线、曲线图表的无级缩放、曲线的动态比较、曲线的打印等等。
- ◇ 第三种是从工具箱中调用历史趋势曲线，对于这种历史趋势曲线，用户需要对曲线的各个操作按钮进行定义，即建立命令语言连接才能操作历史曲线，对于这种形式，用户使用时自主性较强，能做出个性化的历史趋势曲线；该曲线控件最多可以绘制 8 条曲线，该曲线无法实现曲线打印功能。

无论使用哪一种历史趋势曲线，都要进行相关配置，主要包括变量属性配置和历史数据文件存放位置配置。详细配置请见 8.3.1

8.3.1 与历史趋势曲线有关的其它必配置项

8.3.1.1 定义变量范围

由于历史趋势曲线数值轴显示的数据是以百分比来显示,因此对于要以曲线形式来显示的变量需要特别注意变量的范围。如果变量定义的范围很大,例如-999999~+999999,而实际变化范围很小,例如-0.0001~+0.0001,这样,曲线数据的百分比数值就会很小,在曲线图表上就会出现看不到该变量曲线的情况。关于变量范围的定义见图8.16。具体各项含义和使用注意事项请参见第五章“变量定义和管理”。

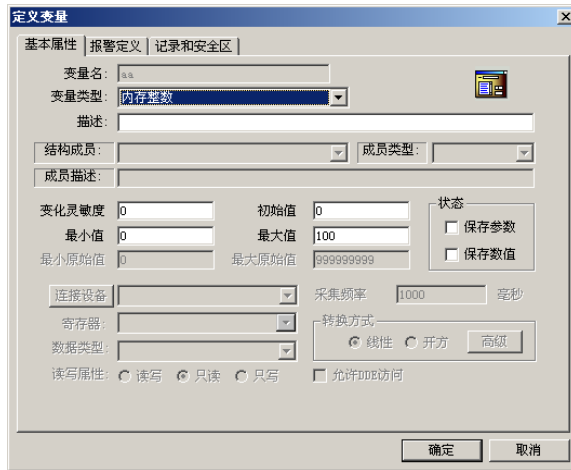


图 8.16 定义变量范围

8.3.1.2 对某变量作历史记录

对于要以历史趋势曲线形式显示的变量,都需要对变量作记录。在组态王工程浏览器中单击“数据库”项,再选择“数据词典”项,选中要作历史记录的变量,双击该变量,则

弹出“变量属性”对话框，如图8.17所示。

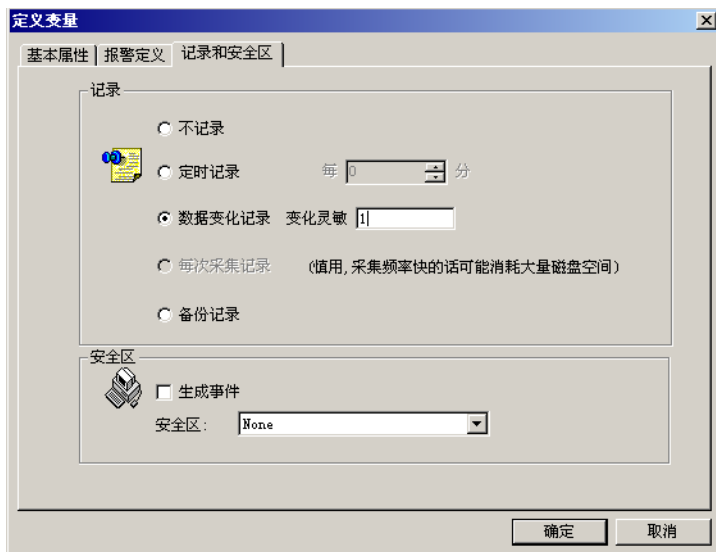
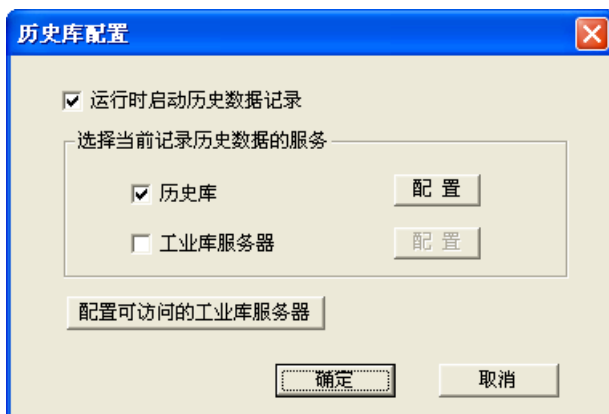


图 8.17 记录定义

选中“记录和安全区”选项卡片，选择变量记录的方式。变量记录的两种方式的具体含义和使用注意事项请参见第五章“变量定义和管理”。

8.3.1.3 定义历史库数据文件的存储目录

在组态王工程浏览器的菜单条上单击“配置”菜单，再从弹出的菜单命令中选择“历史数据记录”命令项，弹出的对话框，如下图所示。



选中“运行时启动历史数据记录”，并且点击“历史库”右边的“配置”按钮，弹出对话框，如图8.18所示。



图 8.18 定义历史库数据文件的存储目录

在此对话框中输入记录历史数据文件在磁盘上的存储路径和数据保存天数，也可进行分布式历史数据配置，使本机节点中的组态王能够访问远程计算机的历史数据。

8.3.1.4 重启历史数据记录

在组态王运行系统的菜单条上单击“特殊”菜单项，再从弹出的菜单命令中选择“重

启历史数据记录”，此选项用于重新启动历史数据记录。在没有空闲磁盘空间时，系统就自动停止历史数据记录。当发生此情况时，将显示信息框通知工程人员，工程人员将数据转移到其它地方后，空出磁盘空间，再选用此命令重启历史数据记录。

下面分别详细讲述三种方式的历史趋势曲线的使用方法。

8.3.2 通用历史趋势曲线

8.3.2.1 历史趋势曲线的定义

在组态王开发系统中制作画面时，选择菜单“图库\打开图库”项，弹出“图库管理器”，单击“图库管理器”中的“历史曲线”，在图库窗口内用鼠标左键双击历史曲线（如果图库窗口不可见，请按 F2 键激活它），然后图库窗口消失，鼠标在画面中变为直角符号“└”，鼠标移动到画面上适当位置，单击左键，历史曲线就复制到画面上了。如图 8.19 所示。拖动曲线图素四周的矩形柄，可以任意移动、缩放历史曲线。

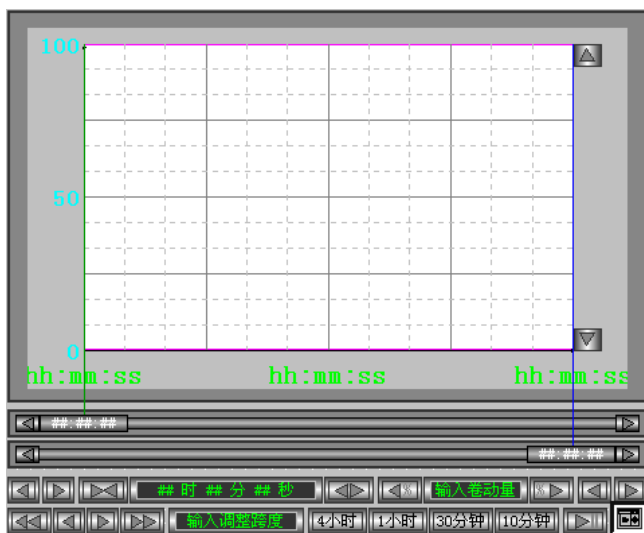


图 8.19 历史趋势曲线

历史趋势曲线对象的上方有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格左方和下方分别是 X 轴（时间轴）和 Y 轴（数值轴）的坐标标注。

曲线的下方是指示器和两排功能按钮。可以通过选中历史趋势曲线对象（周围出现 8 个小矩形）来移动位置或改变大小。通过定义历史趋势曲线的属性可以定义曲线、功能按钮的参数、改变趋势曲线的笔属性和填充属性等，笔属性是趋势曲线边框的颜色和线型，填充属性是边框和内部网格之间的背景颜色和填充模式。

8.3.2.2 历史趋势曲线对话框

生成历史趋势曲线对象后，在对象上双击鼠标左键，弹出“历史趋势曲线”对话框。历史趋势曲线对话框由三个属性卡片“曲线定义”、“坐标系”和“操作面板和安全属性”组成。如图 8.20 所示。

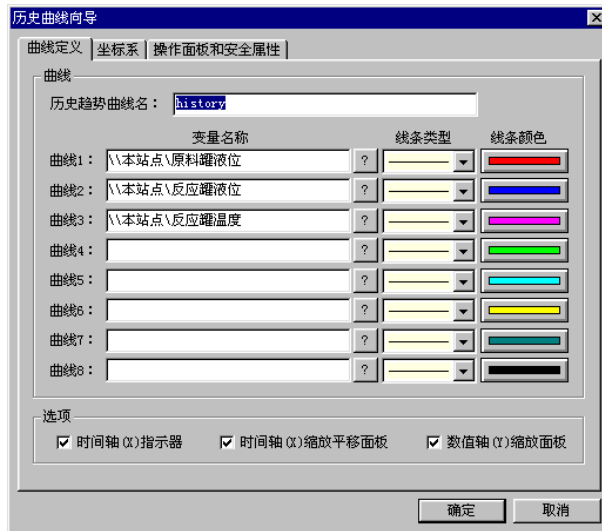


图 8.20 历史趋势曲线对话框

1、曲线定义属性卡片选项：

- ◇ **历史趋势曲线名：**定义历史趋势曲线在数据库中的变量名（区分大小写），引用历史趋势曲线的各个域和使用一些函数时需要此名称。
- ◇ **曲线 1~曲线 8：**定义历史趋势曲线绘制的 8 条曲线对应的数据变量名。数据变量名必须是在数据库中已定义的变量，不能使用表达式和域，并且定义变量时在“变量属性”对话框中选中了“是否记录”选择框，因为“组态王”只对这些变量作历史记录。关于变量的历史记录定义详细情况请参见第五章“变量的定义和管理”中的“变量的基本属性”一节。单击右边的“？”按钮可列出数据库中已定义的变量供选择。每条曲线可由右边的“线条类型”和“线条颜色”选择按钮分别选择线型和线条颜色。



图 8.21 线型选择对话框

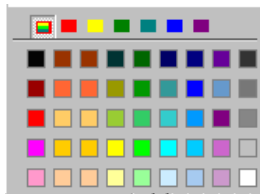



图 8.22 颜色选择对话框

 注意:

该颜色选择对话框为浮动式，用户选择颜色时需按住并移动鼠标到颜色区域，再单击所需要的颜色。

◇ **选项：** 定义历史趋势曲线是否需要显示时间指示器、时间轴缩放平移面板和 Y 轴缩放面板。这三个面板中包含对历史曲线进行操作的各种按钮。选中各个复选框时（复选框中出现“✓”号）表示需要显示该项。

2、坐标系属性卡片选项：



图 8.23 坐标系属性卡片

- ◇ **边框颜色、背景颜色：**分别规定网格区域的边框和背景颜色。按下按钮，弹出浮动调色板，选择所需的颜色，操作方法同曲线“线条颜色”一样。
- ◇ **绘制坐标轴：**选择是否在网格的底边和左边显示带箭头的坐标轴线。选中“绘制坐标轴”检查框（检查框中出现“✓”号）表示需要坐标轴线，同时下面的“轴线”按钮加亮，可选择轴线的颜色和线型。
- ◇ **分割线为短线：**选择分割线的类型。选中此项后在坐标轴上只有很短的主分割线，整个图纸区域接近空白状态，没有网格，同时下面的“次分割线”选择项变灰。
- ◇ **分割线：**X方向和Y方向的“主分割线”将绘图区划分成矩形网格，“次分割线”将再次划分主分割线划分成的小矩形。这两种线都可通过“属性”按钮选

择各自分割线的颜色和线型。分割线的数目可以通过小方框右边“加减”按钮增加或减小，也可通过编辑区直接输入。工程人员可以根据历史趋势曲线的大小决定分割线的数目，分割线最好与标识定义(标注)相对应。

- ◇ **标识 X 轴—时间轴、标识 Y 轴—数值轴：**选择是否为 X 或 Y 轴加标识，即在绘图区域的外面用文字标注坐标的数值。如果此项选中，左边的检查框中出现“√”号，同时下面定义相应标识的选择项也由灰变加亮。
- ◇ **数值轴(Y 轴)定义区：**因为一个历史趋势曲线可以同时显示 8 个变量的变化，而各变量的数值范围可能相差很大，为使每个变量都能表现清楚，“组态王”中规定，变量在 Y 轴上以百分数表示，即以变量值与变量范围（最大值与最小值之差）的比值表示。所以 Y 轴的范围是 0（0%）至 1（100%）。
 - **标识数目：**数值轴标识的数目，这些标识在数值轴上等间隔设置。
 - **起始值：**规定数值轴起点对应的百分比值，最小为 0。
 - **最大值：**规定数值轴终点对应的百分比值，最大为 100。
 - **字体：**规定数值轴标识所用的字体。可以弹出 WINDOWS 标准的字体选择对话框，相应的操作工程人员可参阅 WINDOWS 的操作手册。
- ◇ **时间轴(X 轴) 定义区：**
 - **标识数目：**时间轴标识的数目，这些标识在数值轴上等间隔。在组态王开发系统制作系统中时间是以 yy:mm:dd:hh:mm:ss 的形式表示，在 TouchView 运行系统中，显示实际的时间。
 - **格式：**时间轴标识的格式，选择显示哪些时间量。

- **时间长度：**时间轴所表示的时间范围。运行时通过定义命令语言连接来改变此值。
- **字体：**规定时间轴标识所用的字体。与数值轴的字体选择方法相同。

3、操作面板和安全属性卡片选项：



图 8.24 操作面板和安全属性卡片

- ◇ **操作面板关联变量：**定义 X 轴（时间轴）缩放平移的参数，即操作按钮对应的参数。包括调整跨度和卷动百分比。
 - **调整跨度：**历史趋势曲线可以向左或向右平移一个时间段，利用该变量来改变平移时间段的大小。该变量是一个整型变量，需要预先在数据词典中定义。
 - **卷动百分比：**历史趋势曲线的时间轴可以左移或右移一个时间百分比，

百分比是指移动量与趋势曲线当前时间轴长度的比值，利用该变量来改变该百分比的值大小。该变量是一个整型变量，需要预先在数据词典中定义。

对于调整跨度和卷动百分比这两个变量，用户只需要在数据词典中定义好，在历史曲线的操作按钮上已经建立好命令语言连接，至于其具体使用见8.3.2.3 历史曲线操作按钮。

8.3.2.3 历史趋势曲线操作按钮

因为画面运行时不自动更新历史趋势曲线图表，所以需要为历史趋势曲线建立操作按钮，时间轴缩放平移面板就是提供一系列建立好命令语言连接的操作按钮，完成查看功能。



图 8.25 操作按钮

操作按钮的详细功能说明如下：

1、时间轴单边卷动按钮：

其作用是单独改变使趋势曲线左端或右端的时间值。

◇ 第一排最前面两个按钮：



时间轴左端向左卷动

按下该按钮时可使时间轴左端向左移动，其中移动量可以通过第二排操作按钮“4小时”“1小时”“30分钟”“10分钟”来选择，或者通过“输入调整跨度”按钮（单位为秒）输入该移动量。例如，按下第二排的“10分钟”按钮，当按

一次该按钮时表示将时间轴左端左移 10 分钟，而右端保持不变。



时间轴左端向右卷动

按下该按钮时可使时间轴左端向右移动，操作方法同“时间轴左端向左卷动”类似，例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴左端右移10分钟，而右端保持不变。

◇ **第一排最后面两个按钮：**



时间轴右端向左卷动

按下该按钮时可使时间轴右端向左移动，操作方法同“时间轴左端向左卷动”类似，例如，按下第二排的“10 分钟”按钮，当按一次时表示将时间轴右端左移 10 分钟，而左端保持不变。



时间轴右端向右卷动

按下该按钮时可使时间轴右端向右移动，操作方法同“时间轴左端向左卷动”类似，例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴右端右移10分钟，而左端保持不变。

2、时间轴平动按钮：

其作用是使趋势曲线的左端和右端同时左移或右移。

◇ **第二排前面四个按钮：**



时间轴向左平移

按下该按钮时可使时间轴左右两端同时向左移动，其中移动量可以通过第二排操作按钮“4小时”“1小时”“30分钟”“10分钟”来选择，或者通过“输入调整跨度”按钮（单位为秒）输入该移动量。例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴左右两端同时左移10分钟。



时间轴向右平移

按下该按钮时可使时间轴左右端同时向右移动，操作方法同“时间轴向左平移”类似，例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴左右端同时右移10分钟。



时间轴向左平移两倍

按下该按钮时可使时间轴左右两端同时向左移动，其中移动量是选择或输入的移动量的两倍，例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴左右端同时左移20分钟。



时间轴向右平移两倍

按下该按钮时可使时间轴左右两端同时向右移动，其中移动量是选择或输入的移动量的两倍，例如，按下第二排的“10分钟”按钮，当按一次时表示将时间轴左右端同时右移20分钟。

3、时间轴百分比平移按钮：

其作用是使趋势曲线的时间轴左移或右移一个百分比，百分比是指移动量与趋势曲线当前时间轴长度的比值。比如移动前时间轴的范围是12:00~14:00，时间长度120分钟，

左移10%即12分钟后，时间轴变为11:48~13:48。

◇ 第一排第六——八个按钮

 百分比卷动量输入

按下该按钮弹出百分比卷动量输入对话框，百分比卷动量最小值0，最大值100。

 时间轴百分比左移

按下该按钮可将时间轴两端同时左移一个百分比，百分比量通过“卷动百分比”按钮输入，例如，输入10,表示卷动10%,当按一次时表示将时间轴两端同时左移10%。

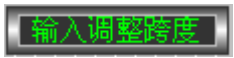
 时间轴百分比右移

按下该按钮可将时间轴两端同时右移一个百分比，百分比量通过“卷动百分比”按钮输入，例如，输入10,表示卷动10%,当按一次时表示将时间轴两端同时右移10%。

4、跨度调整和输入按钮：

选择或输入调整跨度量。

◇ 第二排第五——九个按钮：

 输入“调整跨度”按钮

按下该按钮弹出历史调整跨度输入对话框，输入调整跨度时间（以秒为单位），例如，输入7200，表示时间调整跨度是2小时。



按下该按钮时调整跨度即设置为4小时，按钮按下时呈白色。



按下该按钮时调整跨度即设置为1小时，按钮按下时呈白色。



按下该按钮时调整跨度即设置为30分钟，按钮按下时呈白色。



按下该按钮时调整跨度即设置为10分钟，按钮按下时呈白色。

5、时间轴缩放按钮：

建立时间轴上的缩放按钮是为了快速、细致地查看数据的变化。缩放按钮用于放大或缩小时间轴上的可见范围。

◇ 第一排第三-----五个按钮



时间轴量程显示

显示时间轴的量程。



缩小按钮

将时间轴的量程缩小到左右指示器之间的长度。若左右指示器已在窗口两端，则量程缩小一半。



放大按钮

将时间轴的量程增加一倍。

6、时间轴操作面板其它按钮：



时间更新按钮：将历史曲线时间轴的右端设置为当前时间，以查看最新数据。



参数设置按钮：在软件运行时设置记录参数，包括记录起始时间、记录长度等。在TouchView运行时可用鼠标单击该按钮，弹出对话框：



输入新参数对话框包含以下输入项：

- 起始时间：2001 年 5 月 15 日 8 时 19 分 15 秒
- 时间长度：0 日 15 时 50 分 0 秒
- 起始值：0
- 数值轴长度：100

底部有“确定(O)”和“取消(C)”按钮。

图 8.26 设置参数按钮

8.3.2.4 历史趋势曲线时间轴指示器

移动指示器，就可以查看整个曲线上变量的变化情况。移动指示器可以通过按钮，另外，为使用方便，指示器也可以作为一个滑动杆，指示器已经建立好命令语言连接，具体有以下几种移动方式：

✧ 左指示器向左移动

弹起或按住第一排指示器的左端按钮时，左指示器向左移动。按住时的执行频率是55毫秒。

◇ 左指示器向右移动

弹起或按住第一排指示器的右端按钮时，左指示器向右移动。按住时的执行频率是55毫秒。

◇ 右指示器向左移动

弹起或按住第二排指示器的左端按钮时，右指示器向左移动。按住时的执行频率是55毫秒。

◇ 右指示器向右移动

弹起或按住第二排指示器的右端按钮时，右指示器向右移动。按住时的执行频率是55毫秒。

8.3.3 历史趋势曲线控件

KVHTrend ActiveX Control 历史趋势曲线控件是组态王以 Active X 控件形式提供的绘制历史曲线和 ODBC 数据库曲线的功能性工具。该曲线具有以下特点：

- ◇ 可以连接组态王的历史库，也可以连接工业库服务器，还可以通过ODBC数据源连接到其它数据库上，如Access、SQLServer等。
- ◇ 连接组态王历史库或工业库服务器时，可以定义查询数据的时间间隔，如同在组态王中使用报表查询历史数据时使用查询间隔一样。
- ◇ 完全兼容了组态王原有历史曲线的功能。最多可同时绘制16条曲线。

- ◇ 可以在系统运行时动态增加、删除、隐藏曲线。还可以修改曲线属性。
- ◇ 曲线图表实现无级缩放。
- ◇ 可实现某条曲线在某个时间段上的曲线比较。
- ◇ 数值轴可以使用工程百分比标识，也可用曲线实际范围标识，二者之间自由切换。
- ◇ 可直接打印图表曲线。
- ◇ 可以自由选择曲线列表框中的显示内容。
- ◇ 可以选择移动游标时是否显示曲线数值。
- ◇ 可以在曲线中显示报警区域的背景色。

8.3.3.1 创建历史曲线控件

在组态王开发系统中新建画面，在工具箱中单击“插入通用控件”或选择菜单“编辑”下的“插入通用控件”命令，弹出“插入控件”对话框，在列表中选择“KVHTrend ActiveX Control”，单击“确定”按钮，对话框自动消失，鼠标箭头变为小“十”字型，在画面上选择控件的左上角，按下鼠标左键并拖动，画面上显示出一个虚线的矩形框，该矩形框为创建后的曲线的外框。当达到所需大小时，松开鼠标左键，则历史曲线控件创建成功，画面上显示出该曲线，如图 8.27 所示。

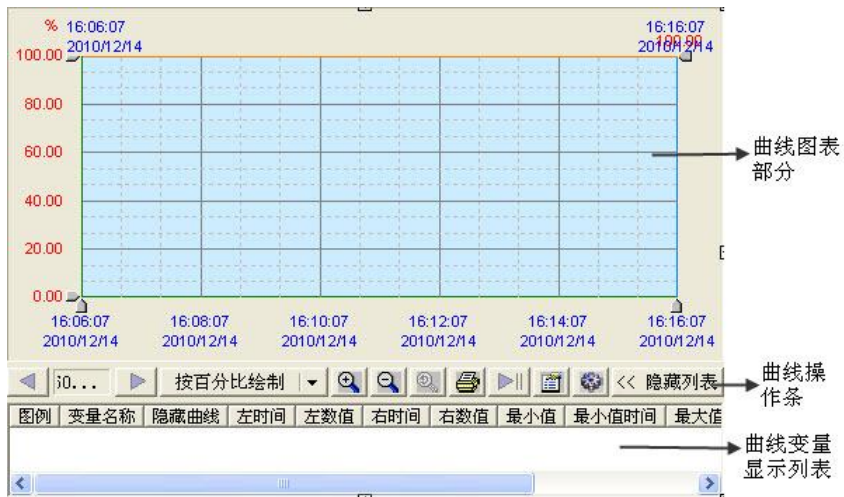


图8.27 历史曲线控件

8.3.3.2 设置历史曲线固有属性

历史曲线控件创建完成后，在控件上单击右键，在弹出的快捷菜单中选择“控件属性”命令，弹出历史曲线控件的固有属性对话框，如图 8.28 所示。

控件固有属性含有以下几个属性页：曲线、坐标系、预置打印选项、报警区域选项、游标配置选项。下面详细介绍每个属性页中的含义。



图8.28 历史曲线控件固有属性

1、曲线属性页

如图 8.28 所示，曲线属性页中下半部分为说明定义在绘制曲线时，历史数据的来源。曲线中数据的来源，可以是组态王历史库、工业库或其它通过 ODBC 连接的数据源。曲线属性页中上半部分“曲线”列表是定义曲线图表初始状态的曲线变量、绘制曲线的方式、是否进行曲线比较等：

◇ 历史库中添加：

从历史库中选择变量到曲线图表，并定义曲线绘制方式。单击“历史库中添加”

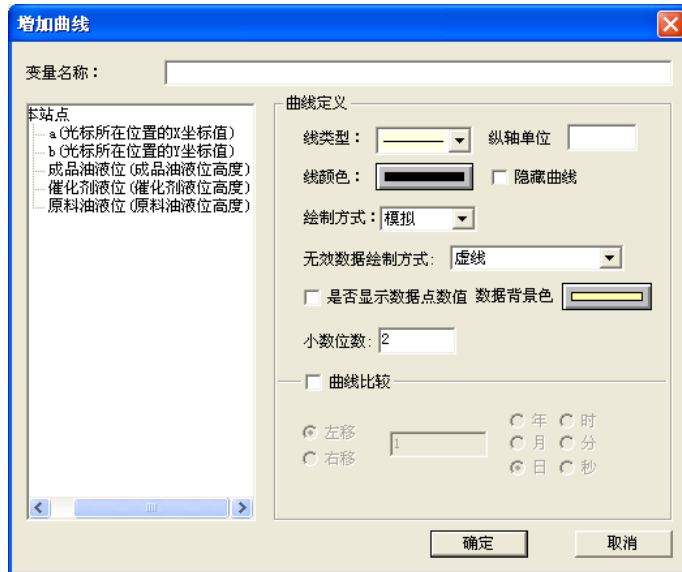
按钮，弹出如图 8.29 所示的对话框。



图 8.29 增加曲线

“增加曲线”对话框中各部分的含义为：

- **变量名称：**在“变量名称”文本框中输入要添加的变量的名称，或在左侧的列表框中选择，该列表框中仅会列出本工程中定义了历史记录属性的变量。单击鼠标，则选中的变量名称自动添加到“变量名称”文本框中，一次只能添加一个变量。变量名称后面的括号中会显示相应的变量描述，如图所示：



在添加、修改历史库变量对应的曲线时，可以按照变量 ID 或变量名称升序或降序排列，以方便进行变量的查找。

- **线类型：**单击“线类型”后的下拉列表框，选择当前曲线的线型。
- **线颜色：**单击“线颜色”后的按钮，在弹出的调色板中选择当前曲线的颜色。
- **绘制方式：**曲线的绘制方式有四种：模拟、阶梯、逻辑、棒图，可以任选一种。
- **无效数据绘制方式：**曲线在曲线变量关联的设备通讯失败，关联变量的质量戳为坏，运行系统退出的情况下显示的方式。分为：虚线，不画线，实线。

- **是否显示数值点数值，数据背景色：**如果选中，在绘图区显示曲线各数据点的数值，同时可以设置数据点值的背景色。
- **隐藏曲线：**控制运行时是否显示该曲线。在运行时，也可以通过曲线窗口下方的列表中的属性选择来控制显示或隐藏该曲线。
- **纵轴单位：**可以为每一条曲线设置不同的“纵轴单位”，如果不设置图 8.29 中的“纵轴单位”，那么曲线的纵轴单位按照图控件“坐标系”页面中“纵轴单位”的设置显示。
- **小数位数：**显示某变量的对应曲线时，设置该曲线数值显示的小数位数。仅当该变量是浮点型时，才起作用。不同的曲线可以设置不同的小数位数。
- **曲线比较：**通过设置曲线显示的两个不同时间，使曲线在绘制位置有一个时间轴上的平移，这样，一个变量名代表的两条曲线中，一个是显示与时间轴相同的时间的数据，另一个作比较的曲线显示有时间差的数据（如昨天），从而达到用两条曲线来实现曲线比较的目的。

选择完变量并配置完成后，单击“确定”，则曲线名称添加到“曲线列表”中。

如图 8.30 所示。

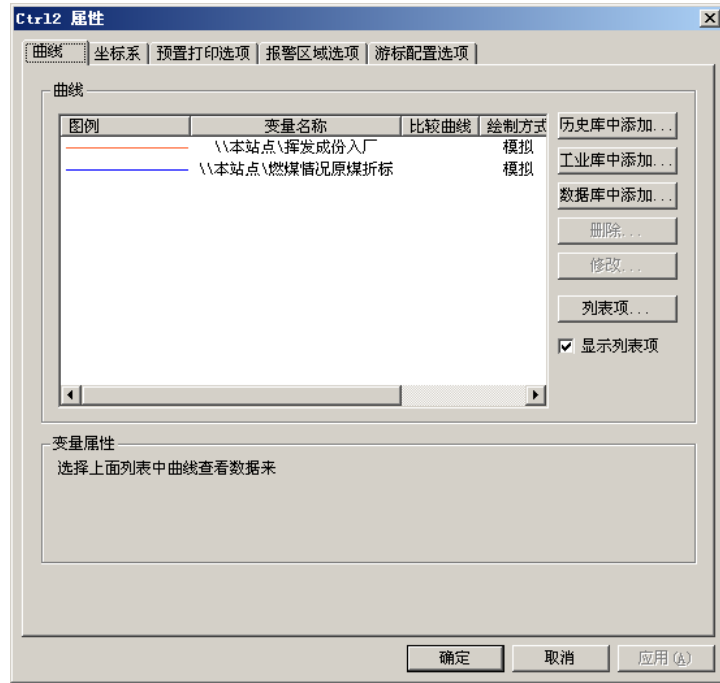


图8.30 增加历史库变量到曲线列表

运行后历史曲线如图所示。

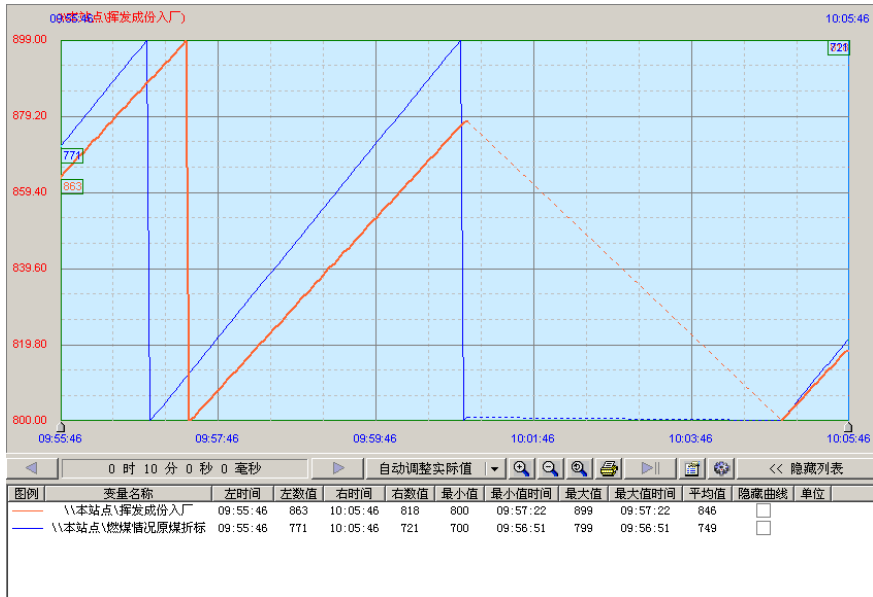


图8.31 显示历史库变量曲线

◇ 工业库中添加:

从工业库中选择变量到曲线图表，并定义曲线绘制方式。单击“工业库中添加”按钮，弹出如图 8.32 所示的对话框。



图8.32 设置工业库曲线

在“历史库配置”对话框中配置“可访问的工业库服务器”以后（参见 24.2 节），服务器名称一栏列出可访问的工业库，选择需要访问的工业库，点击“选择变量”按钮，弹出变量选择对话框，如图 8.33 所示。

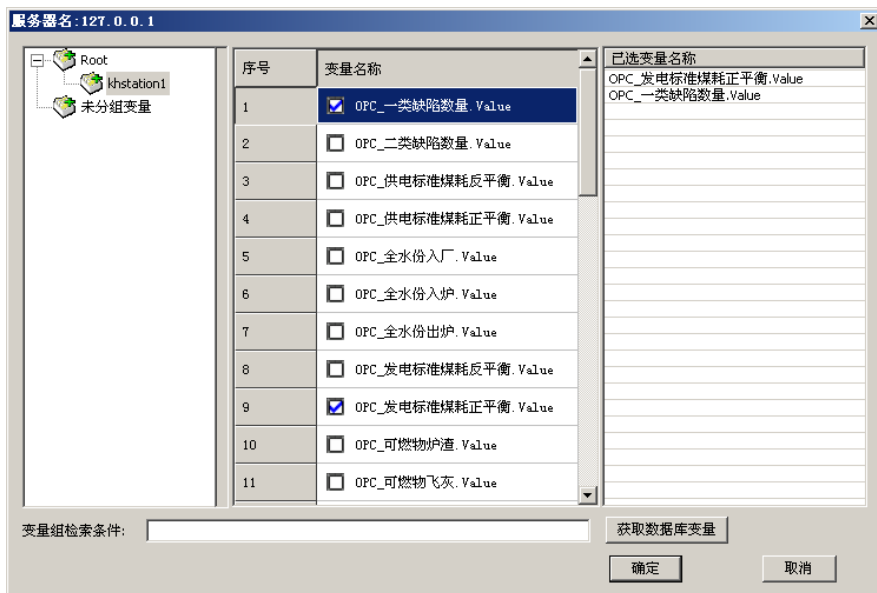


图 8.33 从工业库中选择变量

选中的工业库服务器中的所有的变量组名称显示在左列，用户可选定任意变量组，之后属于这个组的所有变量列于中间的页面。在变量过多的时候，用户可以通过设置“变量组检索条件”来缩小查找的范围，然后在需要的变量名前面的小方框中打上对勾，就可以在右面显示所选中的变量。单击“确定”，“变量名称”列出所选变量，如图 8.34 所示。



图 8.34 增加工业库变量到曲线列表

运行后历史曲线如图 8.35 所示。

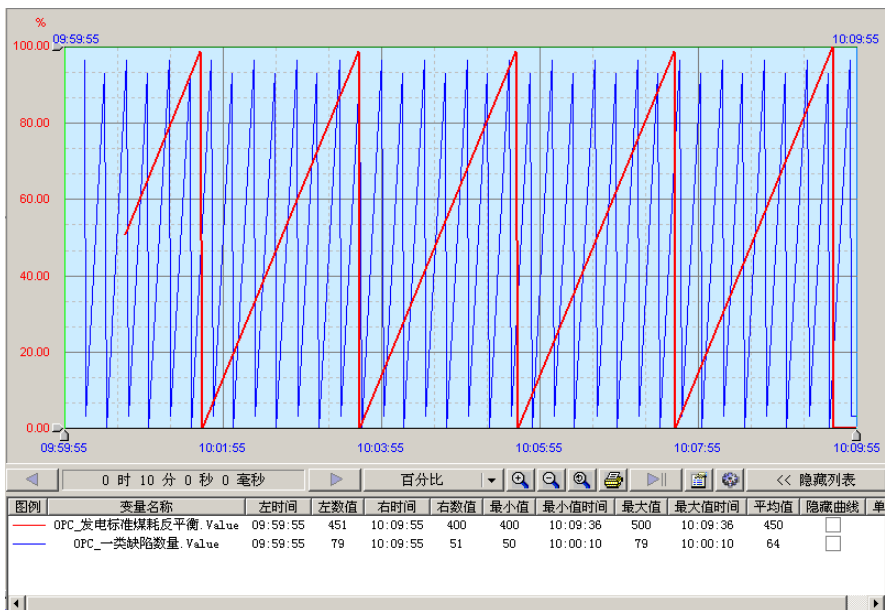


图8.35 显示工业库变量曲线

设置工业库曲线对话框中其它部分的含义同历史库增加曲线对话框中的含义。

◇ 数据库中添加：

从数据库中选择变量到曲线图表，并定义曲线绘制方式。单击“数据库中添加”按钮，弹出如图 8.36 所示的对话框。



图8.36 从数据库选择变量字段

● 数据源

若选择从数据库中添加变量，必须先配置数据源。具体配置方法如下：

- 1) 打开“控制面板”——》“管理工具”——》“数据源（ODBC）”。单击“用户 DSN”项，单击“添加”，弹出“创建新数据源”对话框。单击“用户 DSN”项，单击“添加”，弹出“创建新数据源”对话框。
- 2) 选择所需数据源的驱动如“Microsoft Access Driver (*.mdb)”，单击“完成”按钮。弹出“ODBC Microsoft Access 安装”对话框。
- 3) 在“数据源名”中定义一个数据源名称，用户可以自己设置，数据库“选择”中选择曲线要访问的数据所在的数据库，此数据库的表至少有三个字段：时间字段、数据字段、毫秒字段。单击“确定”，

新创建的数据源就添加到“用户 DSN”列表中。在配置好数据源以后，点击“数据源”按钮，在弹出的“Select Data Source”中选择上面在控制面板中定义的数据源。

- **表名称：**选择曲线使用的数据来自所选数据库的某一个表。
- **时间字段：**选择曲线数据对应的时间记录，日期/时间类型。
- **数据字段：**选择曲线对应的数据值，离散型、长整型或浮点型。
- **毫秒字段：**选择曲线数据对应的毫秒记录，长整型。
- **所在时区：**选择数据库中记录数据时所在的时区。需要用户定义数据来源所在的时区，缺省为当前操作系统设置的时区。如果时区选择错误的话，可能在图表的相应时间段上找不到曲线。
- **无效值：**每一条曲线都和表中一个表示其值的字段关联，这个字段的值在某一时刻可能是无效的，但表的结构决定了这个字段在一条记录中的值不能为空白，所以就有了无效值的定义。



例如

- 1) 当表中数值字段的值为 NULL 时表示该点数据无效，那么配置无效值时就可以为空。
- 2) 当表中数值字段的值为 0 时表示该点数据无效，那么配置无效值时就可以写“0”。
- 3) 当表中数值字段的值为“abcd”时表示该点数据无效，那么配置无效值时

就可以写“abcd”。

曲线设置如图 8.37 所示。



图8.37 数据库变量曲线设置

数据库变量曲线设置完成，点击“确定”，保存后运行，设置好参数以后，曲线的运行画面如图 8.38 所示。

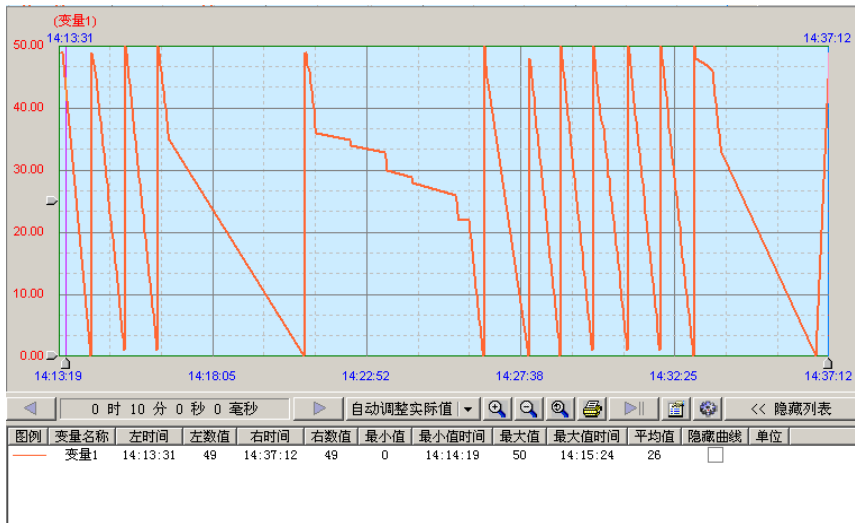


图8.38 显示数据库变量曲线

设置数据库曲线对话框中其它部分的含义与历史库增加曲线对话框中的含义相同。如上所述，可以增加多个变量到曲线列表中。选择已添加的曲线，则“删除”、“修改”按钮变为有效。

- **删除：**删除当前列表框中选中的曲线。选择删除时，系统提示删除确认对话框。除了用该方法可以对曲线进行删除操作之外，我们也可以通过控件方法来实现该功能，详见 8.3.3.4。
- **修改：**修改当前列表框中选中的曲线。选择修改时，系统弹出“修改曲线”对话框，用于对已定义的属性进行修改。
- **显示列表：**选中该项，在运行时，曲线窗口下方可以显示所有曲线的基本情况列表。在运行时也可以通过按钮控制是否要显示该列表。列表中

的内容可按图 8.39 中选择的内容显示，也可以自定义，但“图例”一项不可删除。单击“列表项”按钮，弹出列表项选择对话框，如下图所示。

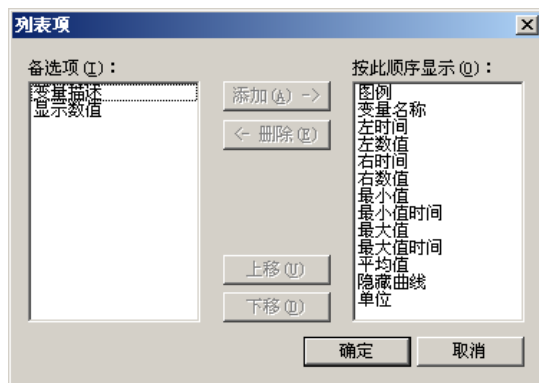


图 8.39 列表项选择对话框

左边列表框中不显示的项，右边列表框中为选中的需要显示的项。选择列表框中的项目，单击“添加”或“删除”，确定显示的项。单击“上移”、“下移”按钮，排列所选择的项的排列顺序。需要注意的是，“图例”一项的位置不可修改。

2、坐标系属性页：

单击“坐标系”标签，进入坐标系属性设置页，如图 8.40 所示。



图 8.40 坐标系属性页

- ◇ **边框颜色和背景颜色：**设置曲线图表的边框颜色和图表背景颜色。单击相应按钮，弹出浮动调色板，选择所需颜色。
- ◇ **绘制坐标轴选项：**是否在图表上绘制坐标轴。单击“轴线类型”列表框选择坐标轴线的线型；单击“轴线颜色”按钮，选择坐标轴线的颜色。绘制出的坐标轴为带箭头的表示 X、Y 方向的直线。
- ◇ **分割线：**定义时间轴、数值轴主次分割线的数目、线的类型、线的颜色等。如果选择了分割线“为短线”，则定义的主分割线变为坐标轴上的短线，曲线图表

不再是被分割线分割的网状结构，如图 8.41 所示。此时，次分割线不再起作用，其选项也变为灰色无效。

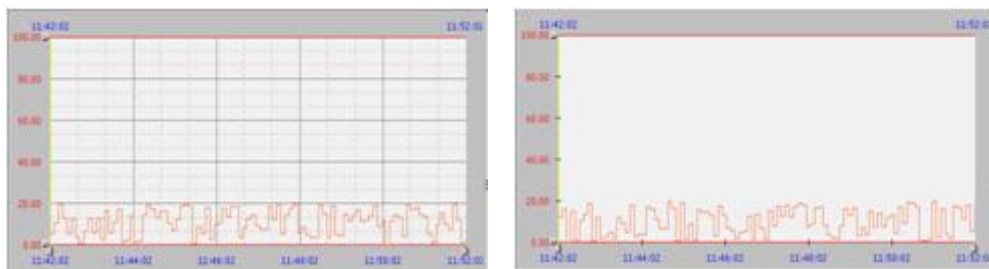


图 8.41 分割线与分割线为短线

◇ 标记数值 (Y) 轴:

- “标记数目”: 编辑框中定义数值轴上的标记的个数。
- “起始值”“最大值”: 规定数值轴起点, 终点对应的值, 当选择“按百分比绘制”时, “起始值”“最大值”表示的是百分比范围 (0~100); 当选择“实际值”时, “起始值”“最大值”表示的是实际数值。
- “纵轴单位”: 在定比例自动调整、实际值和自适应实际值显示模式时, 纵轴单位显示在 Y 轴的顶部。
- “显示变量名称”: 在 Y 轴顶部显示当前选择曲线的变量名称。
- “科学计数法显示”: Y 轴数值以科学计数法表示。
- “所有浮点数据统一设定小数点位数”: 选中并在后面设置小数位数, 则所有浮点类型数据按照设定小数位数显示, 在各曲线定义中设置的小数位数不再起作用。

- “按百分比绘制”：数值轴按照百分比显示，即以变量值与变量范围（最大值与最小值之差）的比值表示。当选择“按百分比绘制”时，数值轴按照“起始值”“最大值”设置的百分比显示。
- “定比例实际值”：曲线形状按百分比绘制，显示的数据值则是当前选中变量的实际值，Y 轴标识范围为曲线对应变量的量程（即在数据词典中设定的最大最小值）。
- “单一轴实际值”：与“定比例实际值”方式不同的是，“定比例实际值”按照各自的比例绘制，“单一轴实际值”则所有曲线均以实际值方式绘制在同一条 Y 轴上。通过双击可切换选择不同曲线，Y 轴数值会按当前曲线的量程进行标识。
- “自适应实际值”：数值轴按照查询时间段内曲线变量的最大最小值自动调整。也就是说在组态王运行系统中查询某时间段内的曲线数据时，每条曲线均可根据该时间段内的最大最小值自动调整，使所有曲线都能够完整的显示在同一绘图区内。在曲线控件的列表框中选中某条曲线时，曲线的数值轴坐标按照该曲线在查询时间段内的最大最小值显示。
- “字体”：单击“字体”按钮，弹出字体选择对话框，设置数值轴标记的字体及颜色等。


◇ 标记时间 (X) 轴：

- “标记数目”编辑框中定义时间轴上的标记的个数。
- 通过选择“格式”选项，选择时间轴显示的时间格式。

- “时间长度”编辑框定义初始显示时图表所显示的时间段的长度。
- 单击“字体”按钮，弹出字体、字型、字号选择对话框，选择数值轴标记的字体及颜色等。

◇ **游标显示：**如果选中，在绘图区显示左游标和右游标。

所有项定义完成后，单击“确定”返回。

 **注意：**

在曲线列表中可以通过双击选中曲线。

 **例：**

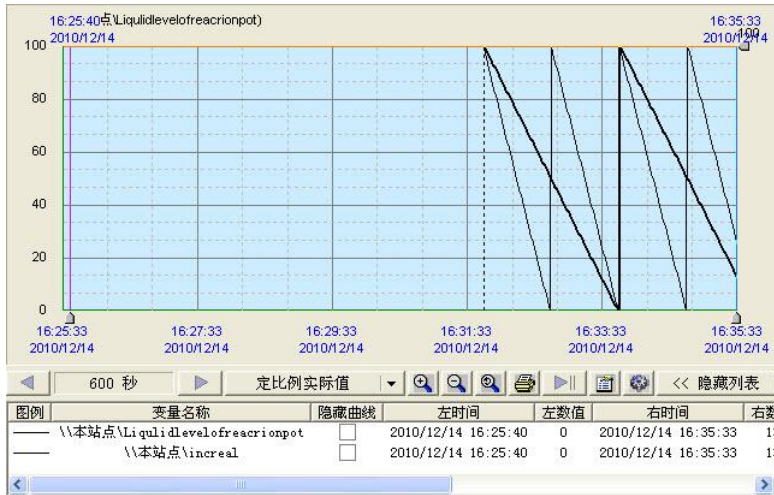
在变量词典中定义如下变量：

变量名	最小值	最大值
inreal	0	50
Liquidlevelofreactionpot	0	100

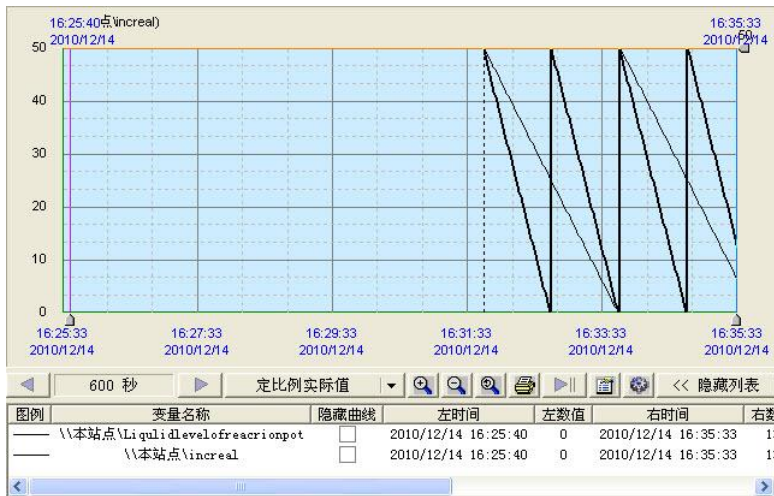
“定比例实际值”时：

选中变量“LiquidlevelofReactionpot”，曲线画面如下图所示：

第八章 趋势曲线和其它曲线

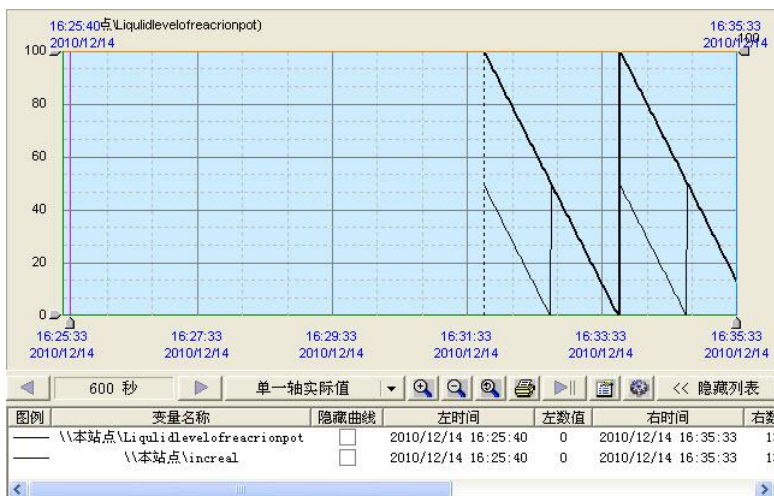


选中变量“increal”，曲线画面如下图所示：

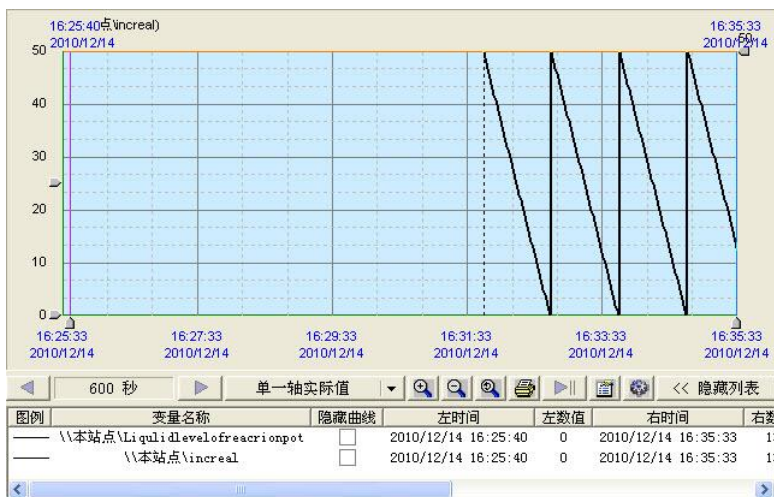


“单一轴实际值”时：

选中变量“LiquidlevelofReactionpot”，曲线画面如下图所示：



选中变量“increal”，曲线画面如下图所示：



3、预置打印选项属性页：

单击“预置打印选项”标签，进入预置打印选项属性页，如图 8.42 所示。

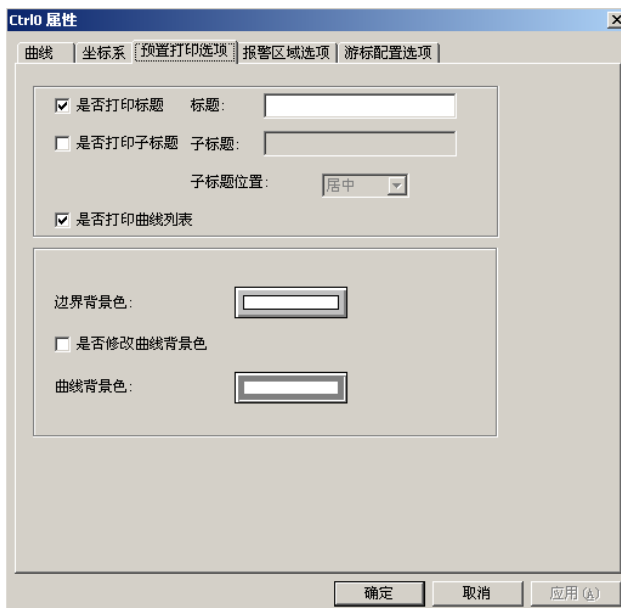


图 8.42 预置打印选项属性页

各选项含义如下：

- ◇ **是否打印标题：**选择此复选框后，标题的编辑框就变为可输入，在此编辑框可以输入要打印图表的标题。
- ◇ **是否打印曲线列表：**选择此复选框，打印曲线变量的列表。
- ◇ **边界背景色：**点击边界背景色按钮，可以为打印曲线选择不同的边界背景色。
- ◇ **是否修改曲线背景色：**选择此复选框时，曲线背景色按钮变为可选，点击此按钮可为打印曲线选择不同的背景色。

4、报警区域选项属性页

在曲线控件中显示报警区域的背景色。

单击报警区域选项按钮，进入报警区域选项属性页，如图 8.43 所示。

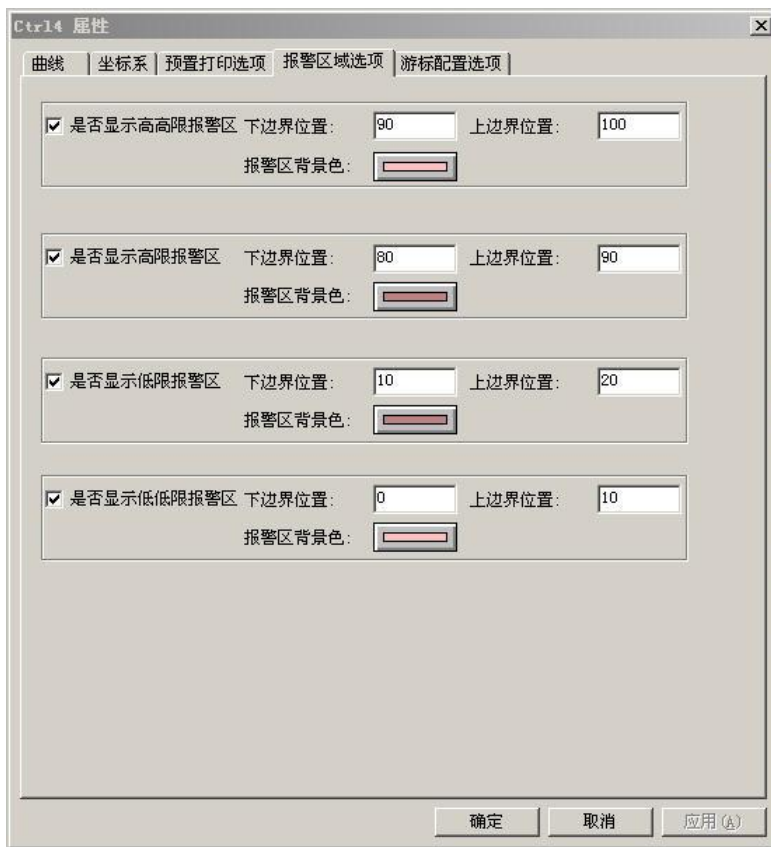


图 8.43 报警区域选项属性页

各选项含义如下：

- ◇ **是否显示高高限报警区：**选中此选项时，下边界位置、上边界位置以及报警区背景色选框被激活。在上、下边界编辑框输入高高报警区域范围的百分比值，然后选择该报警区曲线背景色。

- ◇ 是否显示高报警区：操作同上。
- ◇ 是否显示低报警区：操作同上。
- ◇ 是否显示低低报警区：操作同上。

定义后的结果如下图所示。

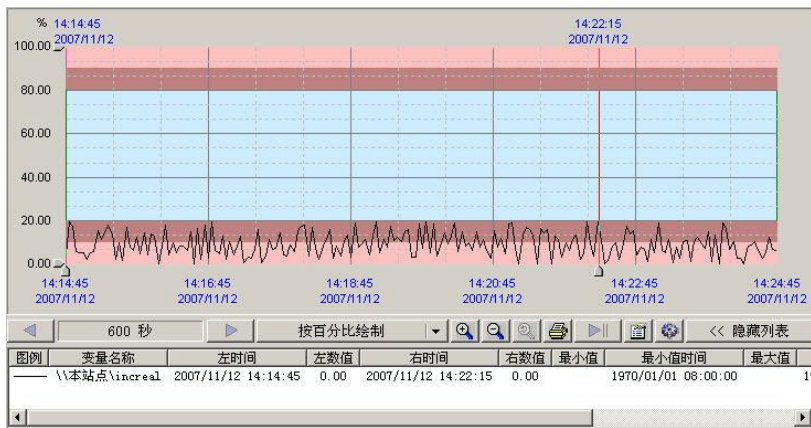


图 8.44 报警区域

5、游标配置选项属性页

当用户在使用历史趋势曲线进行历史数查询时，可能还需要知道某时间段的其它信息，这些信息可能没有在历史数据库中记录，比如操作员信息、班组信息、生产过程的批次信息等。历史曲线控件可以使用 ODBC 从任何第三方数据库中得到这些附加信息。

进入游标配置选项属性页，如图 8.45 所示。



图 8.45 游标配置选项

各选项含义如下：

- ✧ **左游标和右游标：**选中左游标复选框表示对左游标进行操作，选中右游标复选框表示对右游标进行操作。
- ✧ **左游标附加信息和右游标附加信息：**用于定义在游标上显示的附件信息。可以直接选中“左游标附加信息”复选框后在其后的编辑框中输入附加信息（最长 31 个字符），也可以从右边的“从数据库得到附加信息”来选择数据库的对应字段信息。这两种方法不可以同时使用。如果同时选择了编辑框输入和从数据库获得信息，则运行时只显示从数据库获得的信息。左游标与右游标的附加信息功能也不能同时使用。

如选中“左游标附加信息”复选框，在编辑框中输入“反应罐液位”，确定。保存画面，切换到运行系统，结果如下图所示。

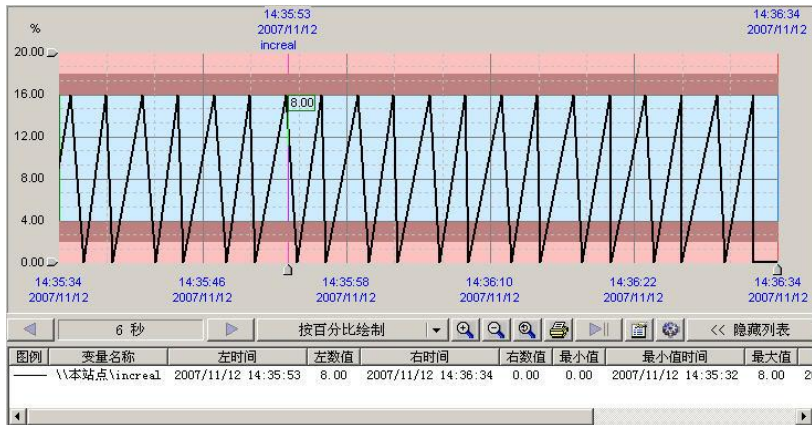


图 8.46 显示游标信息

使用从数据库获取附加信息，要首先建立一个支持 ODBC 访问的数据库，建立数据库表，其主要字段包括：

- 开始时间：日期/时间类型
- 结束时间：日期/时间类型
- 附加信息：字符串或文本型

如下图所示是以 ACCESS 数据库为例定义的附加信息表。

Aclass : 表			
	字段名称	数据类型	说明
	ID1	自动编号	
	开始时间	日期/时间	附加信息的开始时间
	结束时间	日期/时间	附加信息的结束时间
	附加信息	文本	附加信息的具体内容

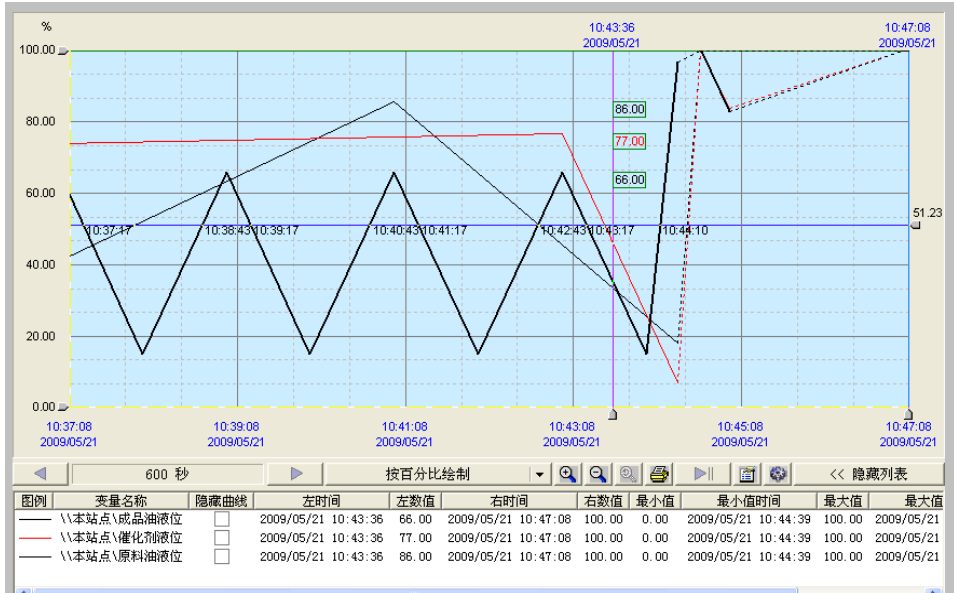
Aclass : 表				
	ID1	开始时间	结束时间	附加信息
	1	2007-11-12	2007-11-12 7:59:59	甲班
	2	2007-11-12 8:00:00	2007-11-12 15:59:59	乙班
	3	2007-11-12 16:00:00	2007-11-12 23:59:59	丙班
	4	2007-11-13	2007-11-13 7:59:59	丁班
	5	2007-11-13 8:00:00	2007-11-13 15:59:59	甲班
	6	2007-11-13 16:00:00	2007-11-13 23:59:59	乙班
	8	2007-11-14	2007-11-13 7:59:59	丁班

图 8.47 附加信息

◇ 曲线数值显示方式:


在曲线数值显示方式下有“从不显示数值”、“一直显示数值”以及“移动游标时显示数值”这三种类型可供选择，当选择其中某一种时，在系统运行后，就会在相应的游标轴处按选择的类型进行曲线数值显示。例如：在左游标和右游标的“曲线数值显示方式”下的选择框中都选中“一直显示数值”选项框。那么在运行系统中，随着左右游标的移动，游标轴上就会一直显示曲线的数值。

注意：移动历史趋势曲线画面上时间轴游标，在游标上的直线旁边会显示一些对应的数据轴的值，同理，移动数据轴上的游标，在其直线旁边会显示某条曲线上不同点对应的的时间值。注意：对于时间轴，可以通过小三角和它上面的直线来进行拖动；而对于数据轴，只能通过小三角来进行移动。如图所示：



- ◇ **数据源：**选择附加信息数据库的 ODBC 数据源名称。
- ◇ **用户名：**选择使用ODBC 数据源的用户名。
- ◇ **密码：**输入使用 ODBC 数据源的用户密码。
- ◇ **表名称：**选择曲线使用的数据来自所选数据库的哪一个表。如上图中的“附加信息表”。
- ◇ **开始时间：**选择数据库中附加信息的开始时间字段。
- ◇ **结束时间：**选择数据库中附加信息的结束时间字段。
- ◇ **附加信息字段：**选择数据库中记录的附加信息字段。
- ◇ **所在时区：**选择数据库记录时间所在的时区。为了统一时间基准，组态王在读取数据库的时间时是按照标准时间——格林威治时间——来读取的。所以需要

用户定义自己所在的时区。如果时区选择错误的话，可能在图表的相应时间段上找不到提示信息。

 例如：

工程师在数据库里建立开始时间为 2007-11-13 8:00 整结束时间为 2007-11-13 15:00 点整，在这段时间段的附加信息字段为早班，那么如下图 8.48 所示，当游标移动到这个时间段范围的任一个时间点时就会显示出早班这个附加信息字段。

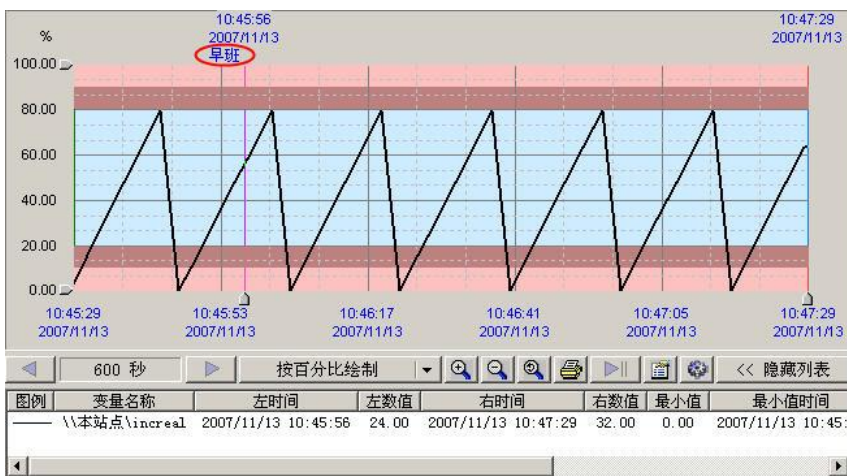


图 8.48 附加信息显示

 注意：

建立数据库里的字段时，开始时间和结束时间字段只能是日期/时间型，附加信息字段只能是字符型，如果定义为其它类型，将不会被调用。

- ◇ **曲线游标处数值显示风格：**定义游标在曲线图表上的风格。
 - **跟随曲线：**移动游标时，数值显示框的位置随曲线与游标的交叉点而变化。
 - **并列显示于上侧：**在移动游标时，数值显示框显示在曲线图表的顶部。
 - **并列显示于下侧：**在移动游标时，数值显示框显示在曲线图表的底部。
 - **显示变量名：**在数值显示框中显示数值的同时显示曲线对应的变量名称。

如下图所示。

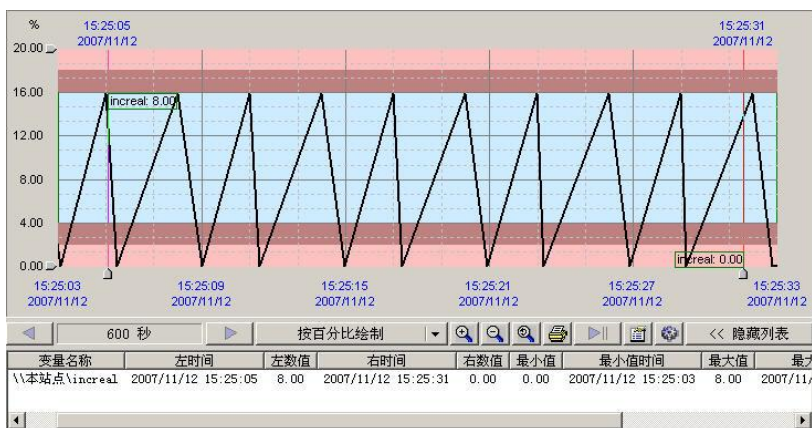


图8.49 显示变量名

- **背景透明：**选择游标数据显示框的背景是否透明。

8.3.3.3 设置历史曲线的动画连接属性

以上所述为设置历史曲线的固有属性，在使用该历史曲线时必定要使用到这些属性。由于该历史曲线以控件形式出现，因此，该曲线还具有控件的属性，即可以定义“属性”和“事件”。关于通用控件的“属性”和“事件”含义请见第十四章“控件”。该历史曲

线的具体“属性”和“事件”详述如下。

用鼠标选中并双击该控件，弹出“动画连接属性”设置对话框，如图 8.50 所示。

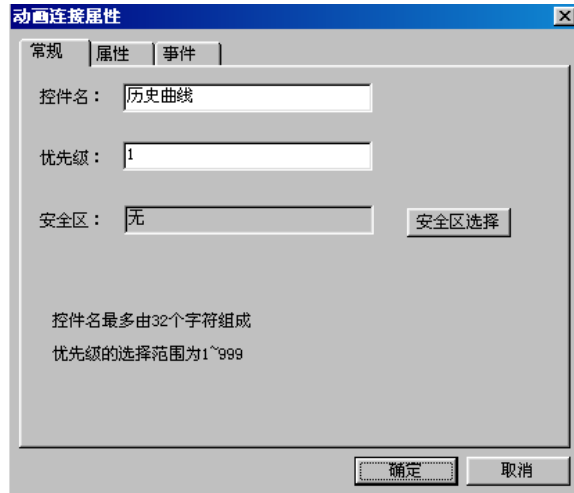


图 8.50 设置控件动画连接属性

动画连接属性共有 3 个属性页，下面一一介绍：

1、“常规”属性页：如图 8.42 所示。

- ◇ **控件名**：定义该控件在组态王中的标识名，如“历史曲线”，该标识名在组态王当前工程中应该唯一。
- ◇ **优先级、安全区**：定义控件的安全性。优先级与安全区含义请见第十六章“系统安全管理”。在运行时，当用户满足定义的权限时才能操作该历史曲线。

2、“属性”属性页：如图 8.51 所示。



图8.51 属性

定义控件属性与组态王变量相关联的关系。该历史曲线的详细属性及含义见 8.3.3.4 列表。

3、“事件”属性页：如图 8.52 所示



图8.52 事件

定义控件的事件函数。该历史曲线的详细事件及含义见 8.3.3.4 列表。

8.3.3.4 历史曲线属性、方法列表

历史曲线属性及含义列表

序号	名称	数据类型	含义
1	CurveUseKVHistData	Long (只读)	曲线历史数据来源的类型： 0：数据库；1：历史库；2： 工业库
2	CurveDSN	String	使用数据源名称
3	CurveTable	String	数据库的表名
4	CurveDateTimeField	String	数据库的时间字段名
5	CurveInvalidColor	OLE_COLOR	数据库的无效值

第八章 趋势曲线和其它曲线

6	CurveVarName	String	连接变量名
7	CurveColor	OLE_COLOR	曲线颜色
8	CurveLineType	Long	曲线线形 (0-9) 参照曲线配置对话框中的“线类型”下拉框
9	CurveTimeDelay	String	曲线比较时间差
10	CurveDrawType	Short	曲线绘制方式 (0-3) : 0: 模拟; 1: 阶梯; 2: 逻辑; 3: 棒图
11	CurveAlmColor	OLE_COLOR	报警显示颜色
12	CurveHideCurve	Bool	是否隐藏曲线
13	CurveMilliSecondField	String	毫秒字段名称
14	CurveDataField	String	数据字段名称
15	CurveInvalidValue	String	无效值字段名称
16	CurveUser	String	ODBC 数据源用户名
17	CurvePwd	String	ODBC 数据源密码
18	CurveShowDotDataVal	Bool	是否显示数据点的数值
19	CurveList	Bool	是否显示曲线列表
21	CurveUnit	String	各曲线数值的单位
21	CurveYMaxDefVal	float	变量定义的量程上限
22	CurveYMinDefVal	float	变量定义的量程下限
23	CurveDecimalNum	long	各曲线浮点类型数值显示的小数位数 (0-9)

24	CurveServerName	String	服务器名称（仅对工业库变量有意义）
25	CtrlBtn	Bool	是否显示控制按钮
26	BorderColor	OLE_COLOR	边界颜色
27	BakColor	OLE_COLOR	背景颜色
28	DrawXYAxis	Bool	是否绘制坐标轴
29	AxisLineType	Short	坐标轴线型
30	AxisColor	OLE_COLOR	坐标轴颜色
31	ShortDivLine	Bool	分割线是否为短线
32	MajXDivNum	Short	X 轴主分割线数
33	MinXDivNum	Short	X 轴副分割线数
34	MajYDivNum	Short	Y 轴主分割线数
35	MinYDivNum	Short	Y 轴副分割线数
36	MajXDivLineType	Short	X 轴主分割线线型
37	MinXDivLineType	Short	X 轴副分割线线型
38	MajYDivLineType	Short	Y 轴主分割线线型
39	MinYDivLineType	Short	Y 轴副分割线线型
40	MajXDivColor	OLE_COLOR	X 轴主分割线颜色
41	MinXDivColor	OLE_COLOR	X 轴副分割线颜色
42	MajYDivColor	OLE_COLOR	Y 轴主分割线颜色
43	MinYDivColor	OLE_COLOR	Y 轴副分割线颜色
44	YLabelNum	Short	Y 轴标记数
45	YminVal	Double	Y 轴标记最小值

第八章 趋势曲线和其它曲线

46	YmaxVal	Double	Y 轴标记最大值
47	YLabelFont	IFontDisp	Y 轴标记字体
48	XLabelNum	Short	X 轴标记数
49	Xwidth	Short	X 轴宽度
50	XLabelFont	IFontDisp	X 轴标记字体
51	XLabelFmtYear	Bool	X 轴标记格式一是否显示年
52	XlabelFmtMonth	Bool	X 轴标记格式一是否显示月
53	XlabelFmtDay	Bool	X 轴标记格式一是否显示日
54	XlabelFmtHour	Bool	X 轴标记格式一是否显示时
55	XlabelFmtMinute	Bool	X 轴标记格式一是否显示分
56	XlabelFmtSecond	Bool	X 轴标记格式一是否显示秒
57	XlabelFmtMilliSecond	Bool	X 轴标记格式一是否显示毫秒
58	XWidthUnit	Short	X 轴宽度单位：秒、分、时、日 0：秒；1：分钟；2：小时； 3：日；4：毫秒
59	XLabelColor	OLE_COLOR	X 轴标记颜色
60	YLabelColor	OLE_COLOR	Y 轴标记颜色
61	CurveNum	Long（只读）	返回当前曲线图表中的曲线数目
62	AdjustWidth	Long	时间轴调整宽度
63	AdjustWidthUnit	short	时间轴调整宽度单位：秒、

			分、时、日、毫秒
64	YMinReal	Double	Y 轴最小值
65	YMaxReal	Double	Y 轴最大值
66	SampleInterval	Long	采样间隔 秒
67	SampleIntervalMs	Long	采样间隔 毫秒
68	PrintTitle	String	打印标题
69	XLVernierShow	Bool	是否显示左游标
70	XRVernierShow	Bool	是否显示右游标
71	XLabelFmtMicroSecond	Bool	X 轴标记格式—是否显示微秒
72	SampleUS	Bool	是否微秒采样
73	bPrintTitle	Bool	是否打印标题
74	bPrintList	Bool	是否印曲线列表
75	YUNIT	String	纵轴单位
76	bHiHiAlarmArea	Bool	是否显示高高限报警区
77	HiHiAlarmAreaUp	Double	高高限报警区上边界位置
78	HiHiAlarmAreaDown	Double	高高限报警区下边界位置
79	bHiAlarmArea	Bool	是否显示高限报警区
80	HiAlarmAreaUp	Double	高限报警区上边界位置
81	HiAlarmAreaDown	Double	高限报警区下边界位置
82	bLoAlarmArea	Bool	是否显示低限报警区
83	LoAlarmAreaUp	Double	低限报警区上边界位置
84	LoAlarmAreaDown	Double	低限报警区下边界位置

85	bLoLoAlarmArea	Bool	是否显示低低限报警区
86	LoLoAlarmAreaUp	Double	低低限报警区上边界位置
87	LoLoAlarmAreaDown	Double	低低限报警区下边界位置
88	LVernierStr	String	左游标附加信息
89	RVernierStr	String	右游标附加信息
90	LVernierAppendStr	Bool	左游标是否显示附加信息
91	RVernierAppendStr	Bool	右游标是否显示附加信息
92	LShowVernierValType	Short	左游标数值显示方式 0. 不显时数值 1. 一直显示数值 2. 移动游标时显示数值
93	RShowVernierValType	Short	右游标数值显示方式 0. 不显时数值 1. 一直显示数值 2. 移动游标时显示数值
94	bPrintSubTitle	Bool	是否打印子标题
95	PrintSubTitle	String	打印子标题
96	SubTitlePos	Short	子标题位置
97	DataumTimeLBound	String	设定 X 轴左边界，界面同步更新 相对时间模式下，曲线的缩放、平移不能超出此范；绝对时间模式下，等同于设定

			查询时间左时间；不能超出当前时间，不能大于 DataumTimeRBound
98	DataumTimeRBound	String	设定 X 轴右边界，界面同步更新 相对时间模式下，曲线的缩放、平移不能超出此范围；绝对时间模式下，等同于设定查询时间右时间；不能超出当前时间
99	RelativeMode	Bool	设置时间轴显示模式，界面同步更新 True 相对时间模式 False 绝对时间模式，兼容老版本

历史曲线控件提供了很多控件方法，供用户在命令语言中调用。下面介绍历史曲线控件方法的用法。

1. CallSetAttrVal (STRING strAttrName, LONG lIndex, FLOAT fAttrVal)

功能：设置某条曲线的属性，该属性为的数值类型，包括 Bool、float，long、short、Double、OLE_COLOR 类型（OLE_COLOR 可以看作整数类型）。

此方法只对以“Curve…”开头的有关曲线的属性有效。

参数：strAttrName：曲线属性名称

lIndex: 曲线索引号

fAttrVal: 设置值, 实型



例: 新建历史趋势曲线控件 hisTrend, 定义按钮弹起时:

```
hisTrend.CallSetAttrVal("CurveDrawType",1,0);
```

把第一条曲线的属性 CurveDrawType 设置为 0, 即将第一条曲线的绘制方式设为按“模拟”方式绘制。

2. CallSetAttrStrVal (STRING strAttrName, LONG lIndex, STRING strAttrVal)

功能: 设置某条曲线的属性, 该属性为字符串类型。

此方法只对以“Curve…”开头的有关曲线的属性有效。

参数: strAttrName: 曲线属性名称

lIndex: 曲线索引号

strAttrVal: 设置值, 字符串型



例: 新建历史趋势曲线控件 hisTrend, 定义按钮弹起时:

```
hisTrend.CallSetAttrStrVal("CurveVarName",1,"Tag1");
```

把第一条曲线的属性 CurveVarName 设置为 tag1, 即将第一条曲线的的变量名称设置为 tag1。

3. AddCurveVarName (STRING szNewName)

功能: 添加历史曲线所查询的变量, 该变量数据来自组态王历史库。

参数: szNewName: 变量名



例：定义变量 Tag1，m 为历史曲线控件名称，定义按钮弹起时

```
m.AddCurveVarName("\\本站点\ Tag1");
```

把变量 Tag1 的历史曲线添加到历史曲线控件。

4. ChangeCurveVarName (LONG lIndex, STRING szNewName)

功能：改变历史曲线所连接的变量，该变量数据来自组态王历史库。

参数：lIndex：曲线索引号

szNewName：变量名



例：定义变量 V1，m 为历史曲线控件名称，定义按钮弹起时

```
m.ChangeCurveVarName(1, "\\本站点\v1");
```

把第一条曲线连接的变量改成 V1。

5. AddCurveVarNameForKHRDB (STRING StrVarName, STRING strServerName)

功能：添加历史曲线所查询的变量，该变量数据来自工业库。

参数：szNewName：变量名

strServerName：服务器名



例：定义变量 Tag1，m 为历史曲线控件名称，server1 为服务器名，定义

按钮弹起时

```
m.AddCurveVarNameForKHRDB("\\本站点\ Tag1", "server1");
```

把变量 server1 服务器上变量 Tag1 的历史曲线添加到历史曲线控件。

6. ChangeCurveVarNameForKHRDB (LONG lIndex, STRINGstr TagName, STRING

strServerName)

功能：改变历史曲线所连接的变量，该变量数据来自工业库。

参数：lIndex：曲线索引号

strTagName：变量名

strServerName：服务器名



例：定义变量 V1，server1 为服务器名，定义按钮弹起时

m.ChangeCurveVarNameForKHRDB(1, "\\本站点\V1", "server1");把第一条曲线连接的变量改成 server1 服务器上的变量 V1。

7. **AddCurveVarNameForODBC**(STRING szNewName,STRING szDSN, STRING szUser,STRING szPwd,STRING szTable,STRING szDateTimeField,STRING szMilliSecondField,STRING szDataField, STRING szInvalidValue)

功能：添加曲线查询的变量，该变量数据来自 ODBC 数据源。

参数：szNewName：所连接变量

szDSN：使用的数据源名

szUser：用户名

szPwd：密码

szTable：数据库的表名

szDateTimeField：表中对应的时间字段

szMilliSecondField：表中对应的毫秒字段

szDataField: 表中对应的数据字段

szInvalidValue: 表中对应的无效值



例: 定义变量 Tag1, m 为历史曲线控件名称, “历史曲线” 为数据源名称。

定义按钮弹起时

```
m.AddCurveVarNameForODBC("\\本站点\ Tag1", "历史曲线", "用户1", "abcd", "milli", "date", "ms", "data1", );
```

把变量来自 ODBC 数据库的变量 Tag1 的历史曲线添加到历史曲线控件。

8. **ChangeCurveVarNameForODBC**(LONG lIndex, STRING szNewName, STRING szDSN, STRING szUser, STRING szPwd, STRING szTable, STRING szDateTimeField, STRING szMilliSecondField, STRING szDataField, STRING szInvalidValue)

功能: 改变历史曲线所连接的变量, 该变量数据来自 ODBC 数据源。

参数: lIndex: 曲线索引号

szNewName: 所连接变量

szDSN: 使用的数据源名

szUser: 用户名

szPwd: 密码

szTable: 数据库的表名

szDateTimeField: 表中对应的时间字段

szMilliSecondField: 表中对应的毫秒字段

szDataField: 表中对应的数据字段

szInvalidValue: 表中对应的无效值



例: 定义变量 V2, m 为历史曲线控件名称, “历史曲线” 为数据源名称。定义按钮弹起时

```
m.ChangeCurveVarNameForODBC(2, "\\本站点\v2", "历史曲线", "用户1", "abcd", "milli", "date", "ms", "data1", );
```

把第二条曲线连接的变量改成 V2。

9. CurveRemove (LONG nIndex)

功能: 删除曲线。

参数: nIndex: 曲线索引号, 从 1 开始。

10. ChangeCurveColor (LONG lIndex, LONG lIndexColor)

功能: 改变一条历史趋势曲线的颜色。

参数: lIndex: 要改变的曲线的索引号

lIndexColor: 要改变为的颜色的编号, 取值范围 0 到 45, 对应调色板中的 45 种颜色。

返回值: Bool 型 为真表示改变成功, 为假表示失败。

11. CallGetAttrVal (STRING strAttrName, LONG lIndex)

功能: 获得曲线的属性名称

此方法只对以“Curve…”开头的属性值为数值型的曲线的属性有效。

参数：strAttrName：曲线的属性名称

lIndex：曲线索引号

返回值：字符串型

12. CallGetAttrStrVal (STRING strAttrName, LONG lIndex)

功能：获得某条曲线的属性名称

此方法只对以“Curve…”开头的属性值为字符串型的曲线的属性有效。

参数：strAttrName：曲线的属性名称

lIndex：曲线索引号

返回值：字符串型

13. GetIdFromVarName (LPCTSTR VarName)

功能：通过变量名称，获取变量在历史趋势曲线中的索引号

参数：VarName：变量的名称

返回值：整型 如果曲线中没有这个变量将返回-1



例：

曲线中按顺序添加了“温度”，“压力”两个变量，通过调用

GetIdFromVarName (“压力”);

将返回“压力”曲线的索引号：2

14. GetLeftVernierTimeValue ()

功能：获取左游标的时间值

参数：无

返回值：整型 自 1970 年 1 月 1 日 0 时起到获取的时间的秒数。

15. GetRightVernierTimeValue ()

功能：获取右游标的时间值

参数：无

返回值：整型 自 1970 年 1 月 1 日 0 时起到获取的时间的秒数。

16. GetSumCurveData (LONG GetTime)

功能：获取叠加曲线，某个时刻的值。

参数：GetTime：时间的年月日时分秒部分，将该时间用 HTConvertTime () 函数转换为自 1970 年 1 月 1 日 0 时到指定时间的秒数。

返回值：实型 获取到的历史数据，如果没有返回 0.0

例：

定义 ST 为要获取的历史数据的时间。

```
ST = HTConvretTime (2011, 12, 12, 0, 0, 0);
```

```
Value = GetSumCurveData (ST);
```

17. HTGetTimeStringAtScooter (LONG nScootNum, STRING szFormat)

功能：取得指定指示器处的时间，该时间为字符串格式

参数：nScootNum：指示器索引，1—左、2—右

szFormat：指定所需要的时间字符串格式，可为下列之一：

“Date”：以 Windows 控制面板相同格式显示日期

“Time”：以 Windows 控制面板相同格式显示时间

“DateTime”：以 Windows 控制面板相同格式显示日期时间

返回值：字符串型日期时间值

18. HTGetPenRealValue (LONG nPenNum, STRING szContent)

功能：取得指定曲线 Y 轴（数值轴）起点或终点的实际值

参数：nPenNum：曲线索引，1~16

szContent：指定所需要类型，可为下列之一：

“Start”：将 Y 轴下限值的工程百分比转换为实际值

“End”：将 Y 轴上限值的工程百分比转换为实际值

“Valid”：判断取得的值是否有效，0—无效，1—有效

返回值：实型，工程百分比对应的实际值



例：定义 PenVal 实数型变量，m 为历史曲线控件名称，定义按钮弹起时

```
PenVal= m.HTGetPenRealValue (1, " Start" );
```

取得第一条曲线在 Y 轴下限值的工程百分比处对应的实际值。

注：本方法不适用于离散型变量。

19. HTGetValue (LONG nPenNum, STRING szValType)

功能：取得指定曲线整个图表上（当前可见部分）的最大/最小/平均值

参数：nPenNum：曲线索引，1~16

szValType：指定所需要的计算类型，可为下列之一：

“AverageValue”：整个趋势的平均值

“MaxValue”：整个趋势的最大值

“MinValue”：整个趋势的最小值

“Valid”：判断取得的值是否有效，0—无效，1—有效

返回值：实型，曲线整个图表上（当前可见部分）的最大/最小/平均值



例：定义 PenVal 为实数型变量，m 为历史曲线控件名称，定义按钮弹起时

```
PenVal= m.HTGetValue (1, " MaxValue" );
```

取得第一条曲线在整个图表上的最大值。

注：本方法不适用于离散型变量。

20. HTGetValueAtScooter (LONG nPenNum, LONG nScootNum, STRING szValType)

功能：取得指定曲线在指示器位置处所要求的值

参数：nPenNum：曲线索引，1~16

nScootNum：指示器索引，1—左、2—右

szValType：指定所需要类型，可为下列之一：

“Value”：取得指示器处的值

“Valid”：判断取得的值是否有效，0—无效，1—有效

返回值：实型，指定曲线在指定指示器处的值



例：定义 PenVal 为实数型变量，m 为历史曲线控件名称，定义按钮弹起时

```
PenVal= m.HTGetValueAtScooter (1, 2, " Value" );
```

取得第一条曲线在右指示器处的值。

21. HTGetValueAtZone (LONG nPenNum, STRING szValType)

功能：取得指定曲线在左右指示器之间的最大/最小/平均值

参数：nPenNum：曲线索引，1~16

szValType：指定所需要的计算类型，可为下列之一：

“AverageValue”：平均值

“MaxValue”：最大值

“MinValue”：最小值

“Valid”：判断取得的值是否有效，0—无效，1—有效

返回值：实型，指定曲线在左右指示器之间的最大/最小/平均值



例：定义 PenVal 为实数型变量，m 为历史曲线控件名称，定义按钮弹起时

```
PenVal= m.HTGetValueAtZone (1, " MaxValue" );
```

取得第一条曲线的最大值。

注：本方法不适用于离散型变量。

22. HTScrollLeft ()

功能：将曲线的起始时间左移一个调整跨度

参数：无

返回值：无

23. HTScrollRight ()

功能：将曲线的起始时间右移一个调整跨度

参数：无

返回值：无

24. HTUpdateToCurrentTime ()

功能：将曲线的终止时间设为当前时间

参数：无

返回值：无

25. HTSetLeftScooterTime (LONG lTime, LONG sMilliSecond)

功能：设置曲线时间坐标起点

参数：lTime：时间的年月日时分秒部分，将该时间用 HTConvertTime () 函数转换为自 1970 年 1 月 1 日 0 时到指定时间的秒数

sMilliSecond：时间的毫秒部分

返回值：无



例：定义 ST 为起始时间，m 为历史曲线控件名称，定义按钮弹起时

```
ST=HTConvretTime(2001, 12, 1, 0, 0, 0);
```

```
m.HTSetLeftScooterTime(ST, 0);
```

设置该曲线的起始时间为 2001 年 12 月 1 日 0 时 0 分 0 秒，毫秒部分为 0。

26. HTZoomIn (STRING szLookString)

功能：通过增大时间轴和数值轴长度，使曲线图表缩小

参数: szLookString: 缩放类型, 可为下列之一:

“StartTime”: 保持起始时间与缩放前相等

“EndTime”: 保持终止时间与缩放前相等

“Center”: 保持中心时间与缩放前相等

返回值: 无



例: 缩小图表, 使中心时间保持不变, m 为历史曲线控件名称, 定义按钮

弹起时 `m.HTZoomIn(“Center”)`;

27. HTZoomOut (STRING szLookString)

功能: 通过缩短时间轴和数值轴长度, 使曲线图表放大

参数: szLookString: 缩放类型, 可为下列之一:

“StartTime”: 保持起始时间与缩放前相等

“EndTime”: 保持终止时间与缩放前相等

“Center”: 保持中心时间与缩放前相等

返回值: 无



例: 放大图表, 使起始时间保持不变, m 为历史曲线控件名称, 定义按钮

弹起时 `m.HTZoomIn(“StartTime”)`;

28. PrintCurve()

功能: 打印, 可进行打印配置的设置。

参数: 无

返回值：无

29. SetCurveSumFlag (STRING strAttrName, BOOL bSet)

功能：对某一曲线进行叠加与否的设置

参数：strAttrName：定义曲线变量的名字例如"\\本站点\datafloat2"

bSet：为 1 表示曲线叠加，为 0 为不叠加。

30. SetCurveSumEnd (LONG nType)

功能：对曲线进行叠加设置后，调用此函数才能生效。

参数：nType： 0: 模拟

1: 阶梯

2: 逻辑

3: 棒图

31. SetTimeParam (LONG lStartTime, LONG sStartms, LONG lXwidth, LONG sXwidthUnit)

功能：设置历史曲线时间坐标起点、时间轴长度

参数：lStartTime 时间年月日时分秒部分

sStartms 时间的毫秒部分

lXwidth 时间轴长度

sXwidthUnit 时间轴长度单位：0-秒 1-分 2-时 3-日 4-毫秒

返回值：无

注：组态王 6.60，sXwidthUnit 必须为 0，即秒，lXwidth 为转换为秒后的时间

长度。



例：m 为历史曲线控件名称，定义 ST 为起始时间，定义按钮弹起时：

```
ST=HTConvretTime(2002, 12, 16, 8, 0, 0);
```

```
m.SetTimeParam(ST, 0, 10, 1);
```

将历史曲线起始时间定义为 2002 年 12 月 16 日 8 时整，历史曲线时间轴长度为 10 分钟。

在组态王第二版中，上例应为：

```
ST=HTConvretTime(2002, 12, 16, 8, 0, 0);
```

```
m.SetTimeParam(ST, 0, 600, 0);
```

将历史曲线起始时间定义为 2002 年 12 月 16 日 8 时整，历史曲线时间轴长度为 10 分钟。

32. SetTimeParamDivided(LONG sStartYear, LONG sStartMonth, LONG sStartDay, LONG sStartHour, LONG sStartMinute, LONG sStartSecond, LONG sStartms, LONG lXwidth, LONG sXwidthUnit)

功能：设置历史曲线时间坐标起点、时间轴长度，可以直接输入时间参数，与 SetTimeParam 实现的功能相同。

参数：sStartYear 时间的年

sStartMonth 时间的月

sStartDay 时间的日

sStartHour	时间的时
sStartMinute	时间的分
sStartSecond	时间的秒
sStartms	时间的毫秒
lXwidth	时间轴长度
sXwidthUnit	时间轴长度单位: 0-秒

返回值: 无



例: m 为历史曲线控件名称, 定义按钮弹起时:

```
m. SetTimeParamDivided(2002, 12, 16, 8, 0, 0, 0, 600, 0)
```

将历史曲线起始时间定义为 2002 年 12 月 16 日 8 时整, 历史曲线时间轴长度为 10 分钟。

33. SetTimeParamString(String strStartTime, String strEndTime)

功能: 设置历史曲线时间坐标起点、终点, 可以直接输入以字符串形式表示的时间参数

参数: strStartTime 开始时间字符串
strEndTime 结束时间字符串



例: m 为历史曲线控件名称, 定义按钮弹起时:

```
m. SetTimeParamString("2004 年 9 月 20 日 13 时 0 分 0 秒 1 毫秒", "2004  
年 9 月 20 日 15 时 20 分 2 秒 999 毫秒")设置时间轴范围为:2004-9-20
```

13:00:00:001~2004-9-20 15:20:02:999;

34. ShowCurve(LONG lIndex, BOOL bShow)

功能：显示或隐藏指定的曲线

参数：lIndex 曲线索引，1~16

bShow 取 1 显示曲线，取 0 隐藏曲线



例：m 为历史趋势曲线控件名称，定义按钮弹起时：

m.ShowCurve (1,0) 隐藏曲线 1。

35. MutePrintCurve()

功能：打印，与控件打印按钮实现相同功能

参数：无

返回值：无

36. SetTimeDataumBound (LPCTSTR lpszTmLBound, LPCTSTR lpszTmRBound)

功能：相对时间模式下，一次设定 X 轴左、右边界；绝对时间模式下，设定查询左、右时间

参数：LPCTSTR lpszTmLBound 查询时间左时间

LPCTSTR lpszTmRBound 查询时间右时间

lpszTmLBound 不能大于 lpszTmRBound；调用方法时，界面同步更新

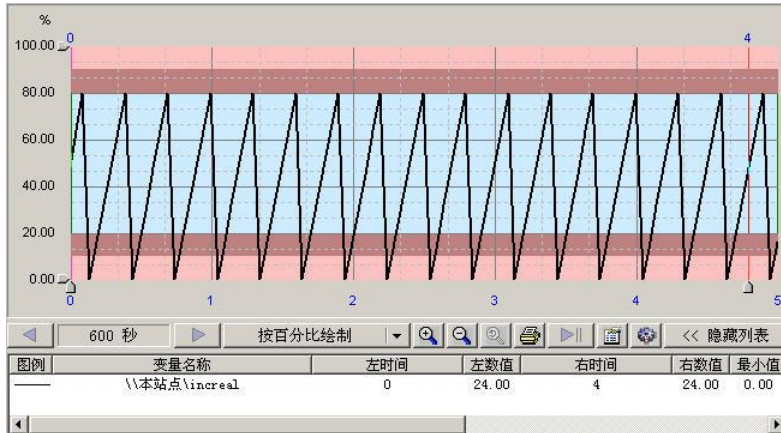


例：m 为历史曲线控件名称，定义按钮弹起时：

m.SetTimeDataumBound("2007 年 11 月 13 日 9 时 48 分 00 秒 000 毫秒",

“2007年11月13日9时53分00秒000毫秒”);

运行结果如图所示。(相对时间模式)



相对时间模式左右边界、单位只能通过脚本调用相关方法来设定，未提供界面输入支持。

相对时间模式下，在“输入新参数”界面，不能查询当前边界以外的参数。关于“输入新参数”，请参考“运行时修改历史曲线属性”一节。

37. `ExportToExcle(long IIdxCurve, LPCTSTR strStartTime, LPCTSTR strEndTime, LPCTSTR filepath, LPCTSTR tablename, long row, long col)`

功能：查询历史库，并将结果记录集导出到 Excel 表格，界面不刷新

参数：IIdxCurve 曲线序号，由 0 开始

strStartTime 查询起始时间

strEndTime 查询结束时间

filepath Excel 模板路径

tablename 工作表名

row 导出位置的起始行

col 导出位置的起始列

导出的数据共两列，第一列为时间，第二列为数值；相对时间模式下，时间为相对时间（本函数不能指定基准时间，而是自动取当前 DataumTimeLBound 和 XWidthUnit 属性，因此，如果相关属性设置不当，导出可能出现负值）；绝对时间模式下，时间为长时间类型；如果 filepath 文件不存在，本函数自动新建空文件；如果 tablename 不存在，本函数自动新建空工作表；此函数不具有保存功能，而是将自动启动 Excel 程序，并将数据导出到指定的工作表，用户可直接观看导出的数据。用户如果需要存盘，请手动操作 Excel。



例：m 为历史曲线控件名称，定义按钮弹起时：

```
m.ExportToExcle(0,"2007年7月28日14时40分00秒000毫秒",  
"2007年7月28日14时50分00秒000毫秒","c:\\test.xls","sheet2",5,5);
```

38. SetWhereForODBC (LONG lIndex, string szWhere)

功能：给历史趋势曲线增加设置 WHERE 条件的函数

参数：lIndex：曲线序号，起始值为 1

szWhere：SQL 条件语句，例如：[字段 1] = '字符串 1' And [字段 2] = 1

 注意:

用户在使用该历史曲线控件时, 请使用本文档中介绍的属性和方法, 其它的属性和方法请勿使用, 以防发生错误。

8.3.3.5 运行时修改历史曲线属性

历史曲线属性定义完成后, 进入组态王运行系统, 运行系统的历史曲线如图 8.53 所示。

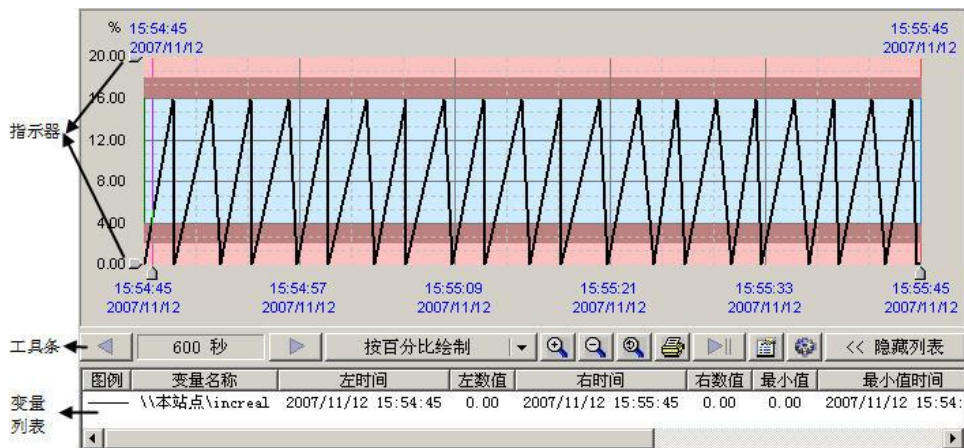


图 8.53 运行时的控件

◇ 数值轴指示器的使用:

拖动数值轴 (Y 轴) 指示器, 可以放大或缩小曲线在 Y 轴方向的长度, 一般情况下, 该指示器标记为当前图表中变量量程的百分比。

◇ 时间轴指示器的使用:

时间轴指示器所获得的时间字符串显示在时间指示器的顶部, 如图 8.53 所示。时间轴指示器可以配合函数等获得曲线某个时间点上的数据。

◇ 工具条的使用：

曲线图表的工具条是用来查看变量曲线详细情况的。工具条的具体作用可以通过将鼠标放到按钮上时弹出的提示文本看到。下面详细介绍每个按钮的作用。



图 8.54 图表工具条

● 调整跨度设置按钮：



图 8.55 修改调整跨度按钮

单击该按钮弹出如图 8.56 所示的对话框，修改当前跨度时间设定值。



图 8.56 修改调整跨度



在“单位”列表框中选择跨度的时间单位，有：日、时、分、秒、毫秒。在“跨度”编辑框中输入时间跨度的数值。

支持毫秒级数据的显示：

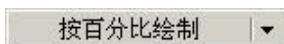
- A. 由于组态王历史库还不支持毫秒数据，因此真正支持到毫秒级目前还只限于 ODBC 数据库，等高速历史库完成后，组态王历史库才真正支持到毫秒级。
- B. 时间轴最短宽度为 10ms。

C. 放缩、移动都支持到毫秒。

在设置参数对话框中(运行时点击设置参数按钮弹出)中不能设置到毫秒级，要设置到毫秒级要使用命令语言。

单击按钮  使曲线图表向左移动一段指定的时间段。单击按钮  使曲线图表向右移动一段指定的时间段。

- **设置 Y 轴标记：**



鼠标单击按钮右侧的下拉箭头，弹出如图 8.57 所示的列表框。

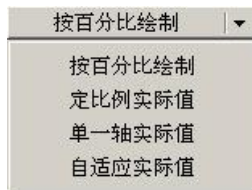



图 8.57 设置 Y 轴标记


选择列表中的“按百分比绘制”，曲线的 Y 轴按照百分比标记。选择列表中的“定比例实际值”，曲线绘制方式与按选择“百分比”方式类似，数值轴标识则根据当前选中曲线的对应变量的量程（在数据词典中定义的最大最小值）显示。选择“单一轴实际值”，曲线的 Y 轴按照实际值标记。选择自动调整实际值，曲线按照查询时间段内的最大最小值自动调整。数值轴标识按照当前曲线（在曲线变量列表中选中的曲线为当前曲线）的最大最小值显示。具体细节请参考“设置历史曲线固有属性”。

- **曲线图表无级缩放：**

放大按钮：  在曲线图表中选择一个矩形曲线区域，单击该按钮，或直接单



击该按钮，可以放大当前的曲线图表：

- A. 当在曲线区域选取了矩形区域时，时间轴最左/右端调整为矩形左/右边界所在的时间，数值轴标记最上/下端调整为矩形上/下边界所在数值，从而使曲线局部放大，左/右指示器位置分别置时间轴最左/右端。
- B. 当未选定矩形区域时，如左/右指示器不在时间轴最左/右端，时间轴最左/右端调整为左/右指示器所在的时间，数值轴不变，从而使曲线局部放大，左/右指示器位置分别置时间轴最左/右端。
- C. 当未选定矩形区域，左/右指示器在时间轴最左/右端时，时间轴宽度调整为原来的一半，保持中心位置不变，数值轴不变，从而使曲线局部放大，左/右指示器位置分别置时间轴最左/右端。

缩小按钮： 在曲线图表中选择一个矩形曲线区域，单击该按钮，或直接单击该按钮，可以缩小当前的曲线图表：

- A. 当在曲线区域选取了矩形区域时，矩形左/右边界所在的时间调整为时间轴最左/右端所在的时间，矩形上/下边界所在数值调整为数值轴最上/下端所在数值，从而使曲线局部缩小，左/右指示器位置分别置时间轴最左/右端。
- B. 当未选定矩形区域时，如左/右指示器不在时间轴最左/右端，左/右指示器所在的时间调整为时间轴最左/右端所在的时间，数值轴不变，从而使曲线局部缩小，左/右指示器位置分别置时间轴最左/右端。
- C. 当未选定矩形区域，左/右指示器在时间轴最左/右端时，时间轴宽度调整为原来的二倍，保持中心位置不变，数值轴不变，从而使曲线局部缩小，左/

右指示器位置分别置时间轴最左/右端。

- **回退按钮**：单击按钮将曲线图表返回到上一次查询时所获得的历史曲线。回退按钮只能回退一次，不能使用时显示为灰色。
- **打印曲线**：单击按钮弹出“打印属性”对话框，如图 8.58 所示。选择打印机，单击“属性”按钮，设置打印属性：纸张大小，打印方向等。可以将当前图表中显示的曲线及坐标系打印出来。

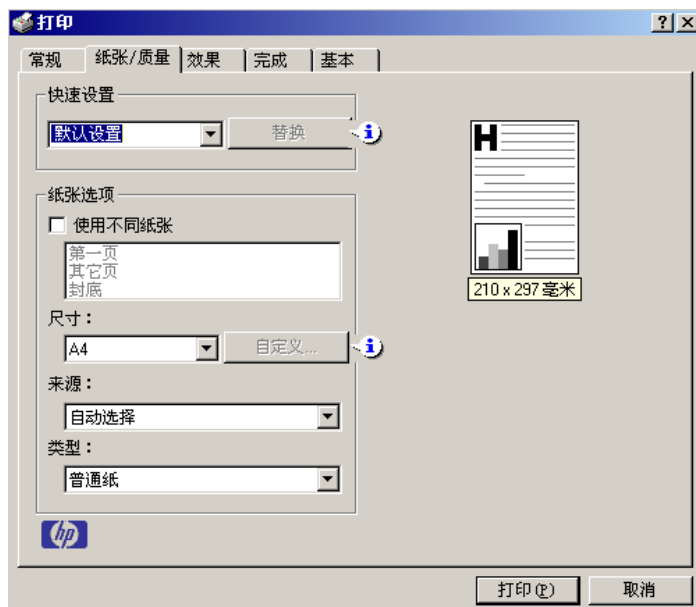




图 8.58 打印属性对话框

- **更新曲线图表终止时间为当前时间**：单击按钮将曲线图表的终止时间更新为当前时间。

- **设置参数：**单击按钮 弹出“输入新参数”对话框，如图 8.59 所示。修改时间轴的起止时间范围和数值轴范围，以及游标的显示方式

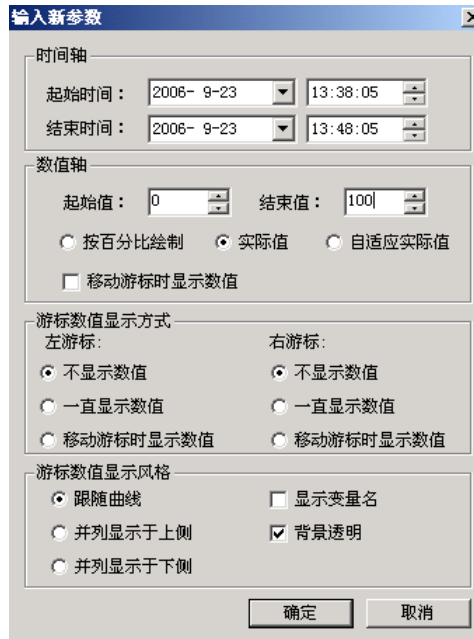



图 8.59 输入新参数

- **设置属性页：**单击“设置属性页”按钮, 弹出“曲线配置属性页”对话框，如图 8.60 所示。

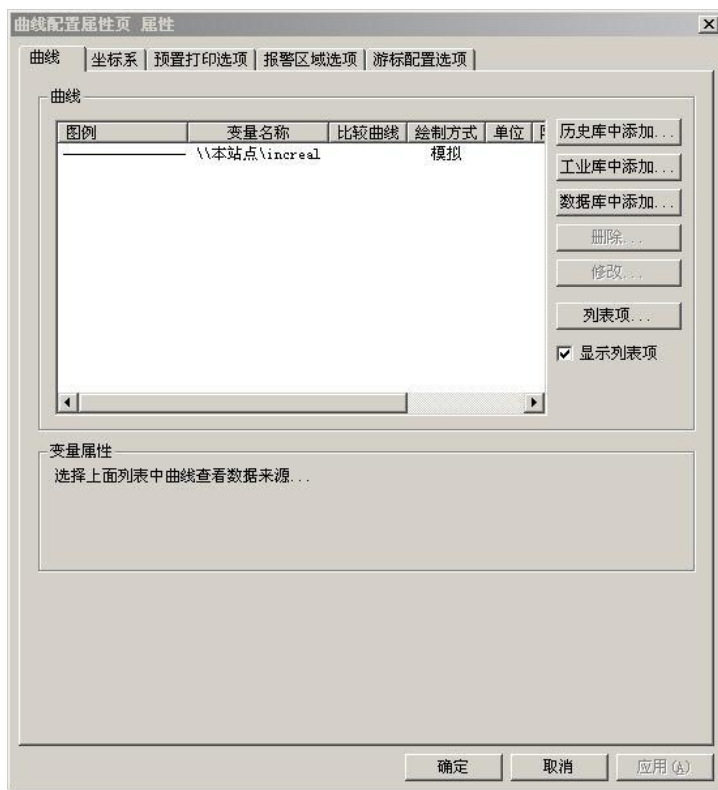


图 8.60 曲线配置属性页

当系统处于运行状态时，可以编辑历史趋势曲线及相关参数，其实现的功能与开发状态时的设置相同。

- **隐藏/显示变量列表：**单击按钮 **<< 隐藏列表** 或 **显示列表 >>** 可以隐藏/显示曲线变量列表。

曲线变量列表

曲线变量列表主要显示当前曲线图表中所显示的曲线及所对应的变量信息，显示的

信息内容由开发系统设置控件属性时，“曲线属性”页的列表项控制。在变量列表上单击右键或选中某条列表项，单击右键，弹出如图 8.61 所示的快捷菜单。

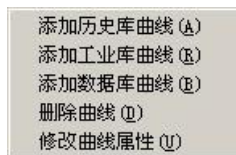


图 8.61 快捷菜单

添加历史库曲线：增加历史库曲线到当前曲线图表。

添加工业库曲线：增加工业库曲线到当前曲线图表。

添加数据库曲线：增加数据库曲线到当前曲线图表。

删除曲线：删除当前列表中选中的曲线。

修改曲线属性：修改当前选中的曲线的绘制属性。

8.3.4 个性化历史趋势曲线

8.3.4.1 历史趋势曲线的定义

在组态王开发系统中制作画面时，选择菜单“工具 \ 历史趋势曲线”项或单击工具箱中的“画历史趋势曲线”按钮，鼠标在画面中变为十字形。在画面中用鼠标画出一个矩形，历史趋势曲线就在这个矩形中绘出：

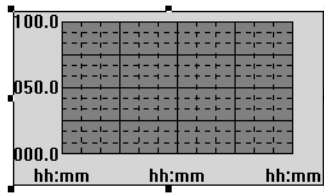


图 8.62 历史趋势曲线

历史趋势曲线对象的中间有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格左方和下方分别是 X 轴（时间轴）和 Y 轴（数值轴）的坐标标注。可以通过选中历史趋势曲线对象（周围出现 8 个小矩形）来移动位置或改变大小。通过调色板工具或相应的菜单命令可以改变趋势曲线的笔属性和填充属性，笔属性是趋势曲线边框的颜色和线型，填充属性是边框和内部网格之间的背景颜色和填充模式。工程人员有时见不到坐标的标注数字是因为背景颜色和字体颜色正好相同，这时需要修改字体或背景颜色。

8.3.4.2 历史趋势曲线对话框

生成历史趋势曲线对象的可见部分后，在对象上双击鼠标左键，弹出“历史趋势曲线”对话框。历史趋势曲线对话框由两个属性卡片“曲线定义”和“标识定义”组成。如图 8.63。



图 8.63 历史趋势曲线对话框

1、曲线定义属性卡片选项：

✦ **坐标轴：**选择是否在网格的底边和左边显示带箭头的坐标轴线。选中“坐标轴”检查框（检查框中出现“✓”号）表示需要坐标轴线，同时下面的“轴线”按钮加亮，可选择轴线的颜色和线型。按下线型按钮或颜色按钮出现如图 8.64 所示的线型和颜色选择对话框。

线型选择方法：在线型按钮上按下鼠标左键，弹出线型选择浮动菜单，按住鼠标左键，向下移动，选择的线型会在按钮上显示出来，当需要的线型出现在按钮上后，松开鼠标。

颜色选择方法：在颜色块按钮上按下鼠标左键，弹出颜色选择浮动菜单，拖动鼠标到颜色浮动菜单上后，松开鼠标左键，然后选择所需颜色。



图 8.64 线型和颜色选择对话框

- ◇ **分割线为短线：**选择分割线的类型。选中此项后在坐标轴上只有很短的主分割线，整个图纸区域接近空白状态，没有网格，同时下面的“次分割线”选择项变灰。
- ◇ **边框色、背景色：**分别规定网格区域的边框和背景颜色。按动这两个按钮的方法与坐标轴按钮类似，弹出的“颜色选择”浮动对话框和选择方法也与之大致相同。
- ◇ **X方向、Y方向：**X方向和Y方向的“主分割线”将绘图区划分成矩形网格，“次分割线”将再次划分主分割线划分成的小矩形。这两种线都可通过线型和颜色按钮选择各自分割线的颜色和线型。分割线的数目可以通过小方框右边“加减”按钮增加或减小，也可通过编辑区直接输入。工程人员可以根据历史趋势曲线的大小决定分割线的数目，分割线最好与标识定义(标注)相对应。
- ◇ **历史趋势曲线名：**定义历史趋势曲线在数据库中的变量名（区分大小写），引用历史趋势曲线的各个域和使用一些函数时需要此名称。
- ◇ **曲线 1~曲线 8：**定义历史趋势曲线绘制的 8 条曲线对应的数据变量名。数据变量名必须是在数据库中已定义的变量，不能使用表达式和域，并且定义变量时在“变量属性”对话框中选中了“是否记录”选择框，因为“组态王”只对这些变量作历史记录。单击右边的“？”按钮可列出数据库中已定义的变量或变量域供选择。每

条曲线可由右边的线型和颜色选择按钮分别选择颜色和线型。操作与“轴线”按钮类似。

- ◇ **无效数据绘制方式:** 曲线在曲线变量关联的设备通讯失败, 关联变量的质量戳为坏, 运行系统退出的情况下显示的方式。分为: 虚线, 不画线, 实线。

2、标识定义属性卡片选项:

在“历史趋势曲线”对话框的上部, 用鼠标单击“标识定义”属性卡片, 则历史趋势曲线对话框变为如8.65对话框。

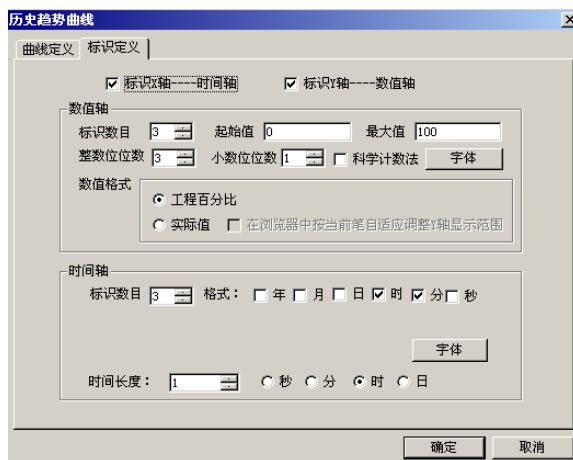


图 8.65 标识定义属性卡片

- ◇ **标识 X 轴—时间轴、标识 Y 轴—数值轴:** 选择是否为 X 或 Y 轴加标识, 即在绘图区域的外面用文字标注坐标的数值。如果此项选中, 左边的检查框中出现“√”号, 同时下面定义相应标识的选择项也由灰变加亮。
- ◇ **数值轴 (Y 轴) 定义区:**

- **标识数目：**数值轴标识的数目，这些标识在数值轴上等间隔
- **起始值，最大值：**规定数值轴起点，终点对应的值，当选择“按百分比绘制”时，“起始值”“最大值”表示的是百分比范围（0~100）；当选择“实际值”时，“起始值”“最大值”表示的是实际数值。
- **整数位位数：**数值轴显示整数的位数。
- **小数位位数：**数值轴显示小数点后面的位数。
- **科学计数法：**数值轴显示为科学计数型
- **字体：**规定数值轴标识所用的字体。可以弹出WINDOWS标准的字体选择对话框，相应的操作工程人员可参阅WINDOWS的操作手册。
- **数值格式：**

工程百分比：数值轴按照百分比显示，即以变量值与变量范围（最大值与最小值之差）的比值表示。当选择“按百分比绘制”时，数值轴按照“起始值”“最大值”设置的百分比显示。

实际值：数值轴按照实际值显示。当选择“实际值”时，数值轴按照“起始值”“最大值”的设置数值显示。

当选择实际值时，如果“在浏览器中按当前笔自适应调整 Y 轴显示范围”选项处于选中状态，那么用户进行该曲线的 WEB 发布时，IE 浏览器中的该曲线可以按照当前的曲线笔自适应调整 Y 轴显示范围，即按照曲线的最值设置 Y 轴的最大最小值。不选中“在浏览器中按当前笔自适应调整 Y 轴显示范围”，IE 上按实际值显示。

◇ 时间轴(X轴)定义区:

- **标识数目:** 时间轴标识的数目, 这些标识在数值轴上等间隔。在组态王开发系统制作系统中时间是以yy:mm:dd:hh:mm:ss的形式表示, 在TouchView运行系统中, 显示实际的时间。
- **格式:** 时间轴标识的格式, 选择显示哪些时间量。
- **时间长度:** 时间轴所表示的时间范围。运行时通过定义命令语言连接来改变此值。
- **字体:** 规定时间轴标识所用的字体。与数值轴的字体选择方法相同。

8.3.4.3 建立历史趋势曲线运行时的操作按钮

因为画面运行时不自动更新历史趋势曲线画面, 所以需要为历史趋势曲线建立操作按钮, 通过命令语言或使用函数改变历史趋势曲线变量的域, 可以完成查看、打印、换笔等功能。

历史趋势曲线变量有以下域:

ChartLength:	历史趋势曲线的时间长度, 长整型, 可读可写, 单位为秒。
ChartStart:	历史趋势曲线的起始时间, 长整型, 可读可写, 单位为秒。
ValueStart:	历史趋势曲线的纵轴起始值, 模拟型, 可读可写。
ValueSize:	历史趋势曲线的纵轴量程, 模拟型, 可读可写。

- ScooterPosLeft: 左指示器的位置, 模拟型, 可读可写。
- ScooterPosRight: 右指示器的位置, 模拟型, 可读可写。
- Pen1---Pen8: 历史趋势曲线显示的变量的 ID 号, 可读可写, 用于改变绘出曲线所用的变量。

如前文所述, 变量在历史趋势曲线的 Y 轴上表示的是一个百分比值, 属性 ValueStart 和 ValueSize 使用的也是百分比表示的值。比如 ValueStart 为 0.2, ValueSize 为 0.6 时, Y 轴上将只显示变量在最大值的 20%到 60%之间的变化。

属性 ScooterPosLeft 和 ScooterPosRight 的值范围在 0.0 和 1.0 之间变化, 其中 0.0 是历史趋势图表的最左边, 1.0 是历史趋势图表的最右边。

常用的按钮主要是定心与移动时间按钮和缩放按钮, 此外, 建立输出动画连接查看数据也经常使用。下图是历史趋势曲线的一个实际例子。

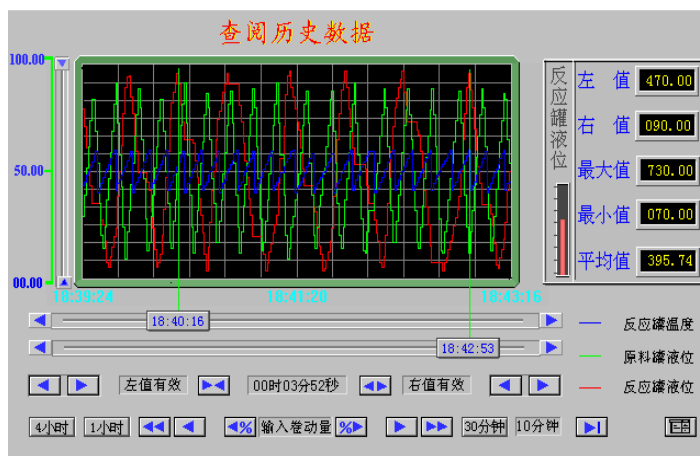


图 8.66 建好操作按钮的历史趋势曲线

◇ 建立定心与移动时间按钮:

- **单边卷动按钮：**其用途是单独改变趋势曲线左端或右端的时间值，命令语言连接程序如下：

时间轴左端向左卷动

```
history.ChartStart=history.ChartStart-3600;
```

```
history.ChartLength=history.ChartLength+3600;
```

其中“history”为历史趋势曲线名，本例将时间轴左端左移1小时，而右端保持不变。

时间轴左端向右卷动

```
history.ChartStart=history.ChartStart+3600;
```

```
history.ChartLength=history.ChartLength-3600;
```

本例将时间轴左端右移1小时，而右端保持不变。

时间轴右端向左卷动

```
history.ChartLength=history.ChartLength-3600;
```

本例将时间轴右端左移1小时，而左端保持不变。

时间轴右端向右卷动

```
history.ChartLength=history.ChartLength+3600;
```

本例将时间轴右端右移1小时，而左端保持不变。

- **时间轴平动按钮：**其作用是使趋势曲线的左端和右端同时左移或右移。

时间轴向左平移

```
history.ChartStart=history.ChartStart-3600;
```

本例按工程人员要求将时间轴左右两端同时左移 1 小时。

时间轴向右平移

```
history.ChartStart=history.ChartStart+3600;
```

本例按工程人员要求将时间轴左右两端同时右移1小时。

- **时间轴百分比平移按钮：**其作用是使趋势曲线的时间轴左移或右移一个百分比，百分比是指移动量与趋势曲线当前时间轴长度的比值。比如移动前时间轴的范围是12:00~14:00，时间长度120分钟，左移10%即12分钟后，时间轴变为11:48~13:48。

时间轴百分比左移

```
HTScrollLeft(history , 10 );
```

本例按工程人员要求将时间轴两端同时左移 10%。

时间轴百分比右移

```
HTScrollRight(history , 10 );
```

本例将时间轴两端同时右移10%。

- **建立指示器移动按钮：**

可以利用系统函数查看指示器处的变量值，再通过移动指示器，就可以查看整个曲线上变量的变化情况。移动指示器可以通过按钮，另外，为使用方便，指示器也可以作为一个滑动杆。建立指示器的方法如下：

单击工具箱内“历史趋势曲线”按钮，在画面上绘制历史趋势曲线对象。同时绘制指示器和移动指示器的按钮。指示器可以由矩形和一条竖直的直线两个图素构

成，可以把矩形部分制作得更美观，但需要把制作矩形部分的所有图素合成为一个复合图素。指示器上的文本对象“##:##:##”将用来显示指示器对应的时间。

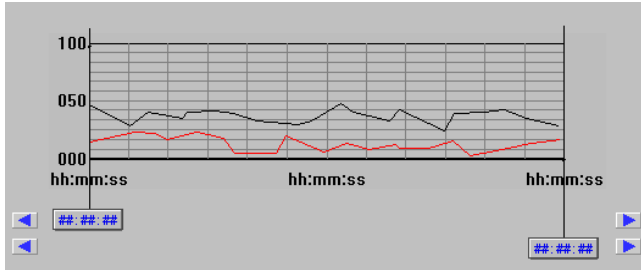


图 8.67 建历史趋势曲线的指示器按钮

为指示器的矩形部分建立滑动杆输入连接。为建立滑动杆连接，需要知道历史曲线窗口的宽度。在曲线窗口下方绘制一条和窗口等宽的直线，双击此直线对象，从弹出的“动画连接”对话框的第一行可以知道此直线的宽度，然后删除直线。假设曲线窗口宽度为390，曲线名为history，为左指示器的矩形部分建立“水平滑动杆输入”动画连接：



图 8.68 指示器建立滑动输入连接

为左指示器的竖直线和对象“##:##:##”建立“水平移动”动画连接：



图 8.69 指示器文本建立移动连接

还要为指示器的文本对象“##:##:##”建立“字符串输出”连接：

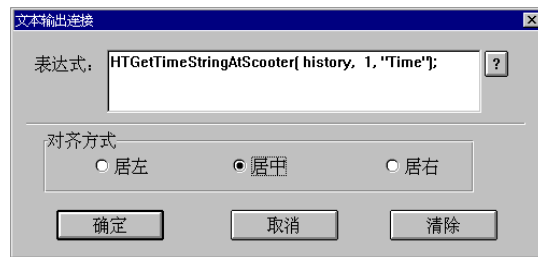


图 8.70 指示器文本建立输出

以上设置的是左指示器，右指示器的设置方法类似。

为指示器移动按钮建立命令语言连接。为使指示器能快速移动，建议设置鼠标“弹起时”和“按住时”两种命令语言连接：

左指示器向左移动

“弹起时”

history.ScooterPosLeft=history.ScooterPosLeft-0.002;

“按住时”：（执行频率 55 毫秒）

```
history.ScooterPosLeft=history.ScooterPosLeft-0.05;
```

左指示器向右移动

“弹起时”

```
history.ScooterPosLeft=history.ScooterPosLeft+0.002;
```

“按住时”：（执行频率 55 毫秒）

```
history.ScooterPosLeft=history.ScooterPosLeft+0.05;
```

右指示器向左移动

“弹起时”：

```
history.ScooterPosRight=history.ScooterPosRight-0.002;
```

“按住时”：（执行频率 55 毫秒）

```
history.ScooterPosRight=history.ScooterPosRight-0.05;
```

右指示器向右移动

“弹起时”：

```
history.ScooterPosRight=history.ScooterPosRight+0.002;
```

“按住时”：（执行频率 55 毫秒）

```
history.ScooterPosRight=history.ScooterPosRight+0.05;
```

✧ 建立缩放按钮：

建立时间轴上的缩放按钮是为了快速、细致地查看数据的变化。建立“缩放按钮”首先需要移动指示器。

- **时间轴的缩放按钮：**此按钮用于放大或缩小时间轴上的可见范围。缩放形式有三种，参见第十章“命令语言程序编写”中关于下列两个函数的说明。

缩小按钮

```
HTZoomIn(history , "Center" );
```

本例将时间轴的量程缩小到左右指示器之间的长度。若左右指示器已在窗口两端，则量程缩小一半。

放大按钮

```
HTZoomOut(history , "Center" );
```

本例将时间轴的量程增加一倍。

◇ 其它按钮

- **时间更新按钮：**将历史曲线时间轴的右端设置为当前时间，以查看最新数据。

命令语言连接程序：

```
HTUpdateToCurrentTime(history);
```

- **参数设置按钮：**在软件运行时设置记录参数，包括记录起始时间、记录长度等。

在TouchView运行时可用鼠标单击“参数设置”按钮，弹出对话框：



图 8.71 参数设置对话框

命令语言连接程序:

```
SetTrendPara( history );
```

◇ 建立输出动画连接

- 查看数值

- 查看变量名: 为文本对象建立“字符串输出”动画连接:

连接表达式: `HTGetPenName(history, 3);`

本例显示history曲线第三支笔对应的变量名。

- 查看指示器处的时间值: 为文本对象建立“字符串输出”动画连接:

连接表达式: `HTGetTimeStringAtScooter(history, 1, "Time");`

本例显示history曲线左指示器对应的的时间值。

- 查看指示器处的变量值

为文本对象建立“模拟值输出”动画连接:

连接表达式: `HTGetValueAtScooter(history, 1, 2, "Value");`

本例显示history曲线左指示器2号笔对应的变量值。

● 查看区间内的统计值

为文本对象建立“模拟值输出”动画连接：

连接表达式：HTGetValueAtZone(history, 1, “MaxValue”);

本例显示history曲线1号笔对应的变量在左右指示器之间的最大值。类似的可以求出变量在这个区间上的最小值和平均值。

◇ 设置数值轴：

当数值的变化比较小时，可以改变趋势曲线窗口内数值轴的范围来放大变量的变化幅度。比如右图将变量在0%~30%的变化放大到整个窗口大小。这种功能一般是通过“滑动杆输入”或按钮来改变曲线的ValueStart和ValueSize属性而实现的。

改变数值轴的起始值

给图素增加“垂直滑动杆输入连接”：



图 8.72 数值轴起始动画连接

改变数值轴的截止值

给图素增加“垂直滑动杆输入连接”：



图 8.73 数值轴截止动画连接

◇ 显示数值轴：

获取指定历史趋势曲线中的趋势笔所对应的实际值。

模拟值输出连接表达式：

```
HTGetPenRealValue(history, 1, "end");
```

```
HTGetPenRealValue(history, 1, "end")/2+HTGetPenRealValue(history, 1, "start")  
)/2HTGetPenRealValue(history, 1, "start")
```

分别对应着数值轴上的最大值, 最小值和中间值。

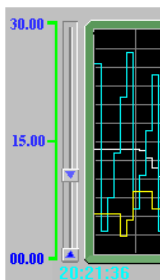


图 8.74 显示数值轴

8.3.5 访问远程历史数据库

组态王不仅能够读取本地计算机中的历史数据库,而且也能够访问远程计算机中的历史数据库,并通过本地机的**历史趋势曲线**观察远程计算机上的历史数据库中变量.访问远程历史数据库的操作方法详见第二十一章“网络功能”。

8.4 组态王内置温控曲线

8.4.1 在画面上放置温控曲线

温控曲线在组态王中以控件形式提供。其操作步骤如下：

- ◇ 单击工具箱中的“插入控件”按钮或选择菜单命令“编辑\插入控件”，则弹出“创建控件”对话框。
- ◇ 在“创建控件”对话框内选择“趋势曲线”下的“温控曲线”控件。
- ◇ 用鼠标左键单击“创建”按钮，鼠标变成十字形。然后在画面上画一个矩形框，温控曲线控件就放到画面上了。可以任意移动、缩放温控曲线控件，如

同处理一个单元一样。在画面上放置的温控曲线控件如图 8.75 所示。

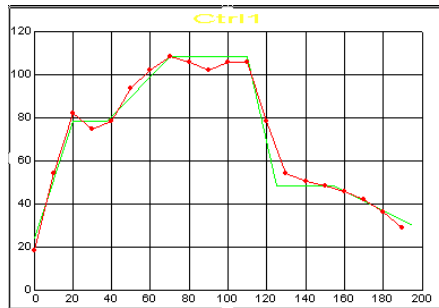


图 8.75 温控曲线

在温控曲线中，纵轴代表温度值，横轴对应时间的变化，同时将每一个温度采样点显示在曲线中，运行环境中还提供左右两个游标，当工程人员把游标放在某一个温度的采样点上时，该采样点的注释值就可以显示出来。

8.4.2 温控曲线属性设置

用鼠标双击温控曲线控件，则弹出温控曲线“属性页”对话框：



图 8.76 温控曲线属性设置

此对话框中各项的含义分别介绍如下：

- ◇ **名称：**温控曲线在组态王中的名称，由字母和数字组成。
- ◇ **访问权限：**温控曲线被操作的用户权限，为 1~999 间的整数。
- ◇ **曲线纵轴表示：**温控曲线纵轴所表示的测量量的名称。如输入“温度”，则纵轴（数值轴）坐标调整自动显示如下图所示。方便用户定义数值轴的名称。

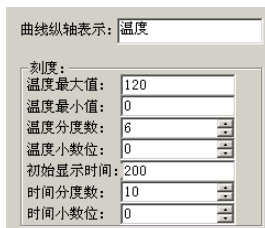


图 8.77 纵轴坐标调整

- ◇ **最大值：**用于设定温控曲线纵轴坐标的最大值，在温控曲线中，如纵轴代表温度变量，而纵轴坐标则代表温度的大小和高低。
- ◇ **最小值：**用于设定温控曲线纵轴坐标的最小值，最小值可以为 0 或负值。
- ◇ **分度数：**用于指定纵轴的最大坐标值和最小坐标值之间的等间隔数，通常默认值为 10 等份间隔。比如，如果纵轴的最大坐标值为 90，最小坐标值为 10，设定温度分度数为 20，则最小坐标值和最大坐标值之间有 20 等份，每一个等份代表的值为 4。
- ◇ **小数位：**用于设置纵轴坐标刻度值的有效小数位。比如，温控曲线纵轴显示要带两位小数时，则在“小数位”编辑框中输入 2 即可。
- ◇ **初始显示时间：**用于设定温控曲线横轴坐标的初始显示时间，同时也即是默认宽度，温控曲线自动滚动时宽度不变。在温控曲线中，横轴代表时间变量，而横轴坐标则代表时间的大小和高低。单位由绘制数据点的平均时间单位确定。
- ◇ **时间分度数：**用于设定横轴时间的分度值，此数越大，则时间分得越细。
- ◇ **时间小数位数：**用于设置横轴坐标刻度值的有效小数位。



注意：

温控曲线的时间轴单位依赖于添加曲线的基本时间单位，如：以秒为基本单位添加数据采集点，则曲线时间轴的单位为秒。

-
- ◇ **最大采集点数：**用于规定曲线上最多显示实时数据点的数目。
 - ◇ **设定曲线：**此按钮用于设置“设定曲线”的显示颜色。单击“设定曲线”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可供使用。设定曲线是由工程

人员事先设计的温度曲线。

- ◇ **实时曲线：**此按钮用于设置“实时曲线”的显示颜色。单击“实时曲线”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可供使用。实时曲线是从现场采集的实际温度变化曲线。
- ◇ **标注文字：**此按钮用于设置“标注文字”的显示颜色。单击“标注文字”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可供使用。标注文字是指游标上显示的文字或数值。
- ◇ **前景：**此按钮用于设置温控曲线纵坐标刻度值、网格线的显示颜色。单击“前景”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可供使用。
- ◇ **背景：**此按钮用于设置温控曲线的背景显示颜色。
- ◇ **游标：**此按钮用于设置“游标”的显示颜色。
- ◇ **字体：**用于设置刻度和游标的字符串字体。
- ◇ **显示操作条：**此选项用于显示/隐藏操作条，当此选项有效时，此选项前面有一个符号“√”。“初始状态”单选框由灰色变为正常色。
- ◇ **初始状态：**当“显示操作条”有效时，初始状态单选框由灰色变为正常色。此选项决定操作条显示时是按最大化还是最小化方式显示，选中某一种初始化状态，此选项前面有一个“●”的标记。如选中“最大化”，此时温控曲线如图 8.78 所示。
- ◇ **不要操作条背景色：**此选项用于控制操作条的背景色，当此选项有效时，此选项前面有一个符号“√”。此时操作条的背景色与界面背景色一样；当此选项无效时，此时操作条的背景色与温控曲线的背景色一样。

- ◇ **显示游标:** 此选项用于显示/隐藏游标, 当此选项有效时, 此选项前面有一个符号“✓”。“注释方式”单选框由灰色变为正常色。
- ◇ **注释方式:** 当“显示游标”有效时, “注释方式”单选框由灰色变为正常色。此选项决定游标在显示时是显示相关信息还是数值。选中某一种状态, 此选项前面有一个“•”的标记。
- ◇ **隐藏刻度值:** 此选项用于显示/隐藏纵轴坐标的刻度值, 当此选项有效时, 此选项前面有一个符号“✓”。
- ◇ **添加网格线:** 此选项用于显示/关闭网格线, 当此选项有效时, 此选项前面有一个符号“✓”, 网格线用于标识纵轴坐标刻度值的大小
- ◇ **标记采样点:** 此选项用于显示/关闭温度采样点, 当此选项有效时, 此选项前面有一个符号“✓”。同时, 温控曲线中出现采样点, 若此选项无效, 则温控曲线中不出现采样点。
- ◇ **显示边框:** 此选项用于显示/隐藏温控曲线的边框, 当此选项有效时, 此选项前面有一个符号“✓”。
- ◇ **设定方式:** 温控曲线的设定方式主要有升温保温模式和自由设定方式两种;
- ◇ **自动保存实时曲线:** 选择此项, 在系统运行时, 每经过一个保存间隔, 将自动保存当前画面图表上绘制的实时曲线。系统会自动在当前工程目录下以该控件名命名一个文件夹, 曲线数据文件被保存到该目录下。
- ◇ **保存间隔:** 自动保存实时曲线的时间间隔, 单位为秒。建议设为添加实时曲线数据点的整数倍。

- ◇ **自动载入实时曲线:** 选择此项, 在系统重新启动或打开画面时(原画面处于关闭状态)自动载入组态王上次运行时自动保存的实时曲线, 当增加一个采样实时值时, 时间偏移量从上次保存的最后一个采样值计算, 如果上次保存的实时曲线, 已达到设定曲线的时间, 则不载入。



例如:

1、 绘制温控曲线

在画面上, 画出温控曲线控件“tru”, 并对温控曲线进行属性定义, 如图 8.81 所示。

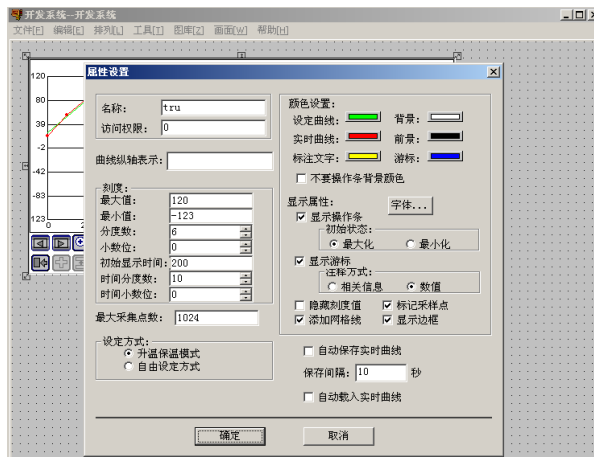


图 8.81 定义温控曲线属性

2、 调入设定温控曲线

设定的温控曲线存储为 csv 文件。例如, 此处设定的温控曲线 setsave.csv 文件(组态王演示工程中包含该文件)为:

SetData

20

39.000000

10.000000, 20.000000, 0

10.000000, 20.000000, 10

20.000000, 20.000000, 10

30.000000, 20.000000, 10

40.000000, 20.000000, 10

其中 20 表示曲线点数，39.000000 表示曲线第一点的位置，

10.000000, 20.000000, 0 表示第一段升温速率为 10，设定时间为 20，保温时间为 0

10.000000, 20.000000, 10 表示第二段升温速率为 10，设定时间为 20，保温时间为 10，以此类推



注意：

必须先生成该 csv 文件，数据是根据工程设备实际需要设定的。

在“画面属性”的“命令语言”中的“显示时”写上如下语句：

```
setchart=InfoAppDir()+"setsave.csv";
```

```
pvLoadData("tru", setchart, "SetValue");
```

表示从当前工程路径下调入设定的温控曲线 setsave.csv。

3、将采集来的数据生成实时曲线

利用函数 `PvAddNewRealPt` 可以在指定的温控曲线控件中增加一个采样实时值。如果需要在画面中一直绘制采集的数据，可以在“命令语言”的“存在时”写入如下语句：

```
IF(时间偏移量<200)
```

```
pvAddNewRealPt("tru",1, 水温,"RV_TIME");
```

其中 `tru` 为控件名称，1 表示相对前一采样点的时间偏移量，“水温”为从设备中采集来的数据，“RV_TIME”为注释性字符串。绘点的速度可以通过改变“存在时”的执行周期来调整。这样就可以对照看出采集的数据是否与原来设定的曲线相一致。

为适应用户的需求，亚控公司专门开发了温控专用解决方案，包括温控控制数据表和温控趋势曲线，具体内容请参见组态王 6.60 产品 CD “解决方案产品”目录中的内容。

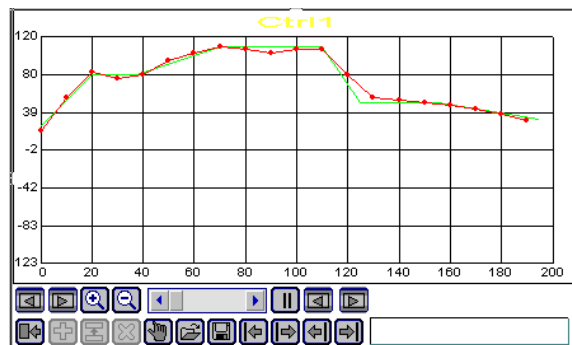



图 8.78 温控曲线操作条最大化

 注意：

当操作条按钮最大化显示时，若要显示出完整的操作条，温控曲线在界面上放置需足够大。

1、操作条

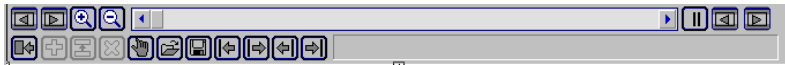





图 8.79 温控曲线操作条


 将 X 轴起点坐标向左移动。


 将 X 轴起点坐标向右移动。


 将 X 轴放大 2 倍


 将 X 轴缩小 2 倍


 左右移动曲线。


 运行系统运行时，当绘制的曲线超出当前画面时，画面自动向右卷滚。


 停止卷滚。


 将 X 轴起点坐标向左移动。

 将 X 轴起点坐标向右移动。

 隐藏或显示操作平台。

 从文件中装载设定曲线。

 保存设定曲线。

 调整坐标系。单击此按钮，弹出对话框：

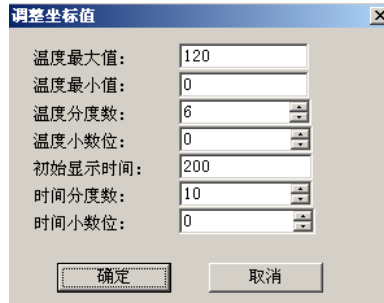






图 8.80 调整坐标系

温度最大值，温度最小值，温度分度数，小数位数用以重新确定纵轴—温度轴坐标；时间分度数，时间最大值，用以重新确定横轴—时间轴坐标。

 左游标左移。


 左游标右移。

 右游标左移。

 右游标右移。

以下三个按钮用于对选定的某段温控曲线进行操作，选定温控曲线的某段曲线，下面的三个按钮将变为有效。


2、插入设置段

 插入设置段。单击此按钮，显示如下：

增加段: 速率: 目标值: 保时: 段前 段后

- 增加段：用于显示温控曲线被选中的段号，从0开始，不可输入。
- 速率：用于设置该段温控曲线的斜率。
- 目标值：用于设置该段温控曲线最终要达到的值。
- 保时：用于设置该段温控曲线保持温度段的时间。
- 段前：用于决定将所设置温控曲线的某段插入到段前。
- 段后：用于决定将所设置温控曲线的某段插入到段后。
- 确认：保存设置，并将该设置插入到温控曲线的相应段。
- 取消：不保存设置。


3、修改设置段

 修改设置段。单击此按钮，显示如下：

修改段: 速率: 目标值: 保时:


确认：保存设置，并按该设置修改温控曲线的相应段。

其它选项同上所示。


 删除设置段。将选定的温控曲线的某段删除。


4、曲线的缩放卷滚操作：



 左边界左移:时间轴左边界坐标左移初始显示时间/时间分度数个时间单位。

 左边界右移:时间轴左边界坐标右移初始显示时间/时间分度数个时间单位。


 放大曲线


 缩小曲线




曲线卷滚操作: 点击左按钮, 时间轴向左平移当前显示时间宽度时间分度数 2 个时间单位。点击游标左侧, 时间轴向左平移当前显示时间宽度时间分度数 2 个时间单位。拖动游标, 快速卷动。

 停止自动卷滚。

 开始自动卷滚。

 右边界左移:时间轴右边界坐标左移初始显示时间/时间分度数个时间单位。

 右边界右移:时间轴右边界坐标右移初始显示时间/时间分度数个时间单位。

 注意:

若选择“升温保温模式”, 不可以在温控曲线上添加设定; 若选择“自由设定方式”则可以在温控曲线上直接添加设定。

8.5 超级 X-Y 曲线控件

超级 XY 曲线控件是组态王以 Active X 控件形式提供的 XY 曲线,该曲线控件可以同时显示 16 条曲线和每条曲线对应的 Y 轴。与组态王内置的 XY 曲线相比,功能更强大,使用更方便。

关于Active X控件的概念和使用方法,请参见本手册第十四章“控件”。这里只介绍本控件的使用方法。

8.5.1 创建超级 X-Y 曲线

在组态王开发系统中新建画面,在工具箱中单击“插入通用控件”或选择菜单“编辑”下的“插入通用控件”命令,弹出“插入控件”对话框,在列表中选择“KvChartXY ActiveX Control”,单击“确定”按钮,对话框自动消失,鼠标箭头变为小“十”字型,在画面上选择控件的左上角,按下鼠标左键并拖动,画面上显示出一个虚线的矩形框,该矩形框为创建后的曲线的外框。当达到所需大小时,松开鼠标左键,则历史曲线控件创建成功,画面上显示出该曲线,如图8.83所示。

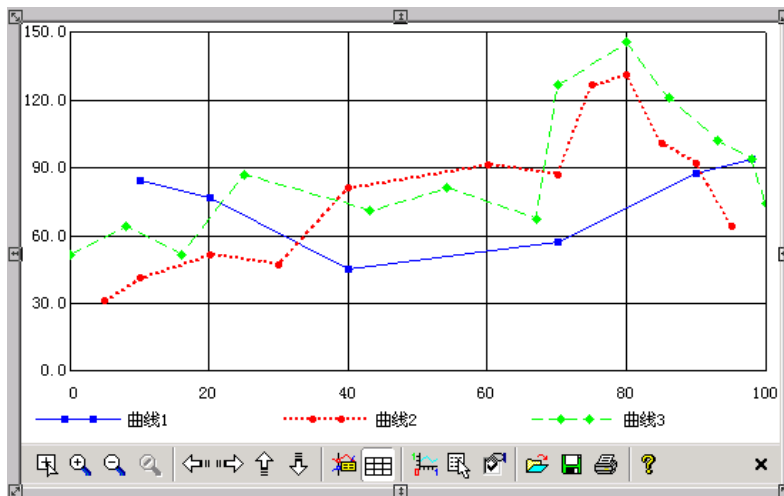


图8.83 创建后的超级XY曲线

8.5.2 设置超级 X-Y 曲线的固有属性

选中画面上创建后的控件，单击鼠标右键，在弹出的快捷菜单中选择“控件属性”，系统弹出曲线的固有属性对话框，如图8.84所示。

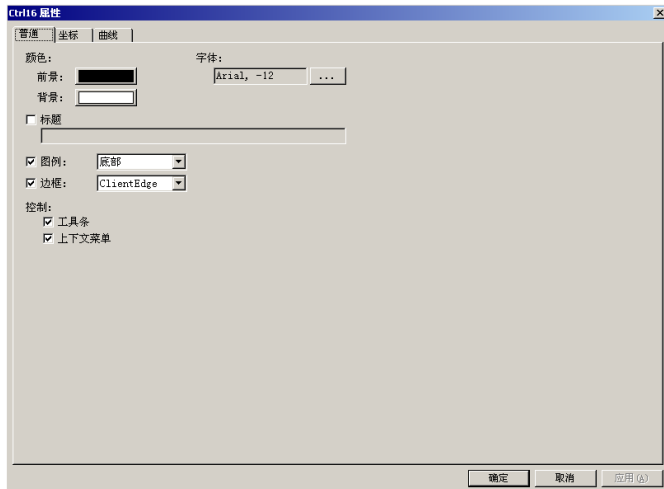


图8.84 超级XY曲线的固有属性页—普通

1、普通属性页各项配置：

- ◇ **颜色：** 设置曲线背景颜色，曲线图表上字符标识和分割线颜色（前景颜色）。直接单击颜色按钮，从弹出的颜色菜单中选择需要的颜色。
- ◇ **字体：** 选择曲线图表上标识字符的字体属性，单击字体按钮，在弹出的对话框中选择字体属性。
- ◇ **标题：** 为曲线图表添加一个标题。如图8.85所示。

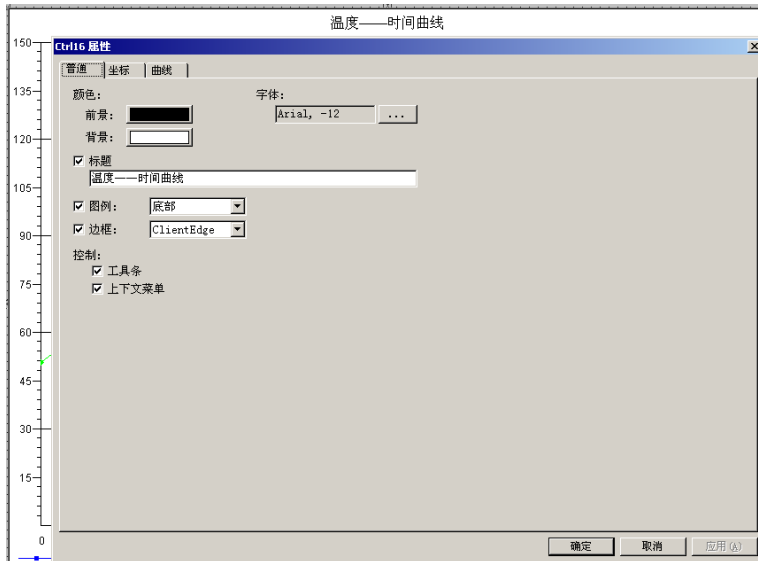


图8.85 设置曲线标题

- ◇ **图例**：选择图例在曲线图表上显示的位置。有底部和顶部两个选项。
- ◇ **边框**：选择曲线边框的样式：ThinBorder—单线边框；ClientEdge—客户 边界；StaticEdge—静态边界。
- ◇ **控制**：选择是否选择图表上提供的工具条。

2、坐标属性页各项配置：



图8.86 超级XY曲线的固有属性页—坐标

- ◇ **X轴信息：**定义定义X轴的标题，最值，小数位数和网格数。
 - 显示X轴：选中以后，在组态王画面显示出X轴。
 - X轴标题：在选择显示X轴以后，此项才可以被激活并进行修改，选中此项以后，组态王画面会显示出X轴的名称。
 - 最大值，最小值：X轴坐标轴的范围。
 - 网格数：按照设置的数值在绘图区X轴方向等分。范围：1~100。
 - 小数位：显示X轴的数据的小数位。范围：0~10。
- ◇ **Y轴信息列表：**超级XY曲线支持多条Y轴显示，最多可设置16曲线，在列表选中要设置的Y轴，设置后，要点击“更新Y轴信息”按钮，设置的相应信息才会显示在这个

列表中。

- Y轴信息主要是用来对多条Y轴进行设置。
- 显示Y轴：选中以后,在组态王画面显示出Y轴。
- Y轴标题：在选择显示Y轴以后,此项才可以被激活并进行修改,选中此项以后,组态王画面会显示出Y轴的名称。
- 最大值,最小值：Y轴坐标轴的范围。
- 刻度数：按照设置的数值将Y轴等分。范围：1~100。
- 小数位：显示X轴的数据的小数位。范围：0~10。
- 网格数：按照设置的数值将X轴等分。范围：1~100。
- 在曲线画图区水平位置：曲线画图区(控件中曲线数据画图区的区域)左边、右边：可以将Y轴设置在画图区的左边或右边。
- 刻度在Y轴的水平方向：同左边、右边相联系,确定Y轴的位置。
- 刻度在Y轴的水平方向：指Y轴刻度显示的位置。
- 更新Y轴信息:将Y轴的设置更新在Y轴信息列表中,每次更改设置后,只有更新Y轴信息,设置才会生效。

◇ **网格线**：选中此项,在绘图区显示网格。

3、曲线属性页各项配置：

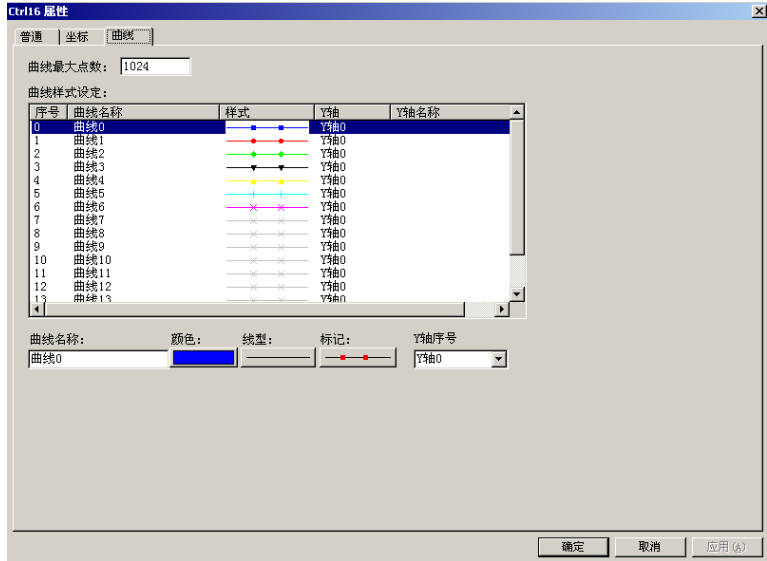


图8.87 超级XY曲线的固有属性页—曲线

- ✧ **曲线最大点数:** 定义每条曲线绘制时最多描述的数据点的个数，如1024，则表明每条曲线可以最多添加1024个数据点，点数的多少将影响机器的运行速度，请选择适当的点数。
- ✧ **曲线样式设定:** 定义曲线绘制的颜色、线型和Y轴名称。首先在列表选中要修改的曲线，在对话框底部的编辑框和颜色、线形选择按钮上显示出当前曲线的设置，可以进行修改。
- ✧ **曲线名称:** 在曲线样式设定列表中，选中一个曲线，该文本框显示选中曲线名称，此时可以对该曲线进行颜色线型等修改。
- ✧ **颜色、线型、标记:** 设置曲线显示的外形。

- ◇ **Y轴序号**：由于超级XY曲线是多Y轴的，该选项是将坐标页设置的Y轴名称显示在列表中Y轴名称项下。

控件属性定义完成后，单击“应用”按钮可以查看定义的效果。如果确定定义完成，单击“确定”按钮关闭对话框。

8.5.3 设置超级 X-Y 曲线的动画连接属性

在使用该控件之前，需要定义控件的动画连接属性。鼠标双击控件，弹出控件的动画连接属性对话框，如图8.88所示。

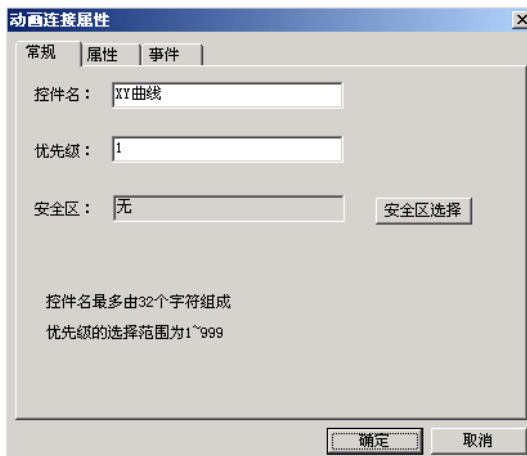


图8.88 超级XY曲线的动画连接属性

在“常规”属性页中的“控件名”对话框中输入控件名称。定义控件的操作优先级和安全区。控件属性页中的属性与8.6.2节中介绍的控件固有属性的含义相同，不再一一介绍。

8.5.4 超级 X-Y 曲线的使用

超级XY曲线提供了丰富的控件方法供用户调用，另外在控件界面上提供了功能全面的工具条供操作使用。如图8.89所示，为运行状态下的超级XY曲线。

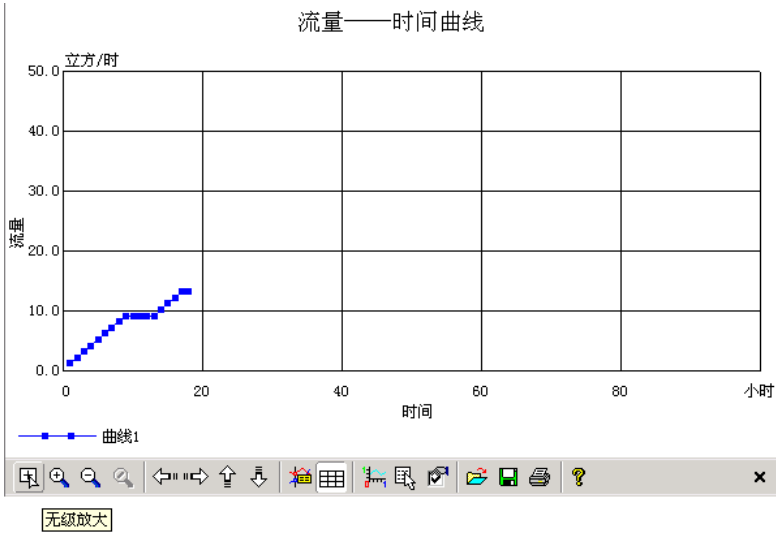








图8.89 运行时的超级XY曲线

操作人员可以利用曲线工具条功能对曲线进行属性修改、无法缩放、移动、保存、打印等操作。

- ◇  无级放大：单击该按钮，然后在图表上按下鼠标左键，拖动并选择一个矩形区域，当松开鼠标时，所选择的区域被自动放大。以选择区域的四边为轴坐标。
- ◇  放大两倍：以图表的X、Y轴中心为准，将曲线放大两倍，曲线图表向两边和上下扩展。

- ◇  缩小两倍：以图表的X、Y轴中心为准，将曲线缩小两倍，曲线图表由两边和上下向中心压缩。
- ◇  原始大小：将曲线图表恢复到原始定义的大小。
- ◇  移动：点击一次该按钮，曲线按照按钮的方向分别会向左、右、上或下移动一个最小单位刻度。
- ◇  显示数值：当该按钮有效时，在图表上将显示一个指示器，如图8.90所示。移动指示器，右上角会显示当前指示器的坐标和曲线的数值。

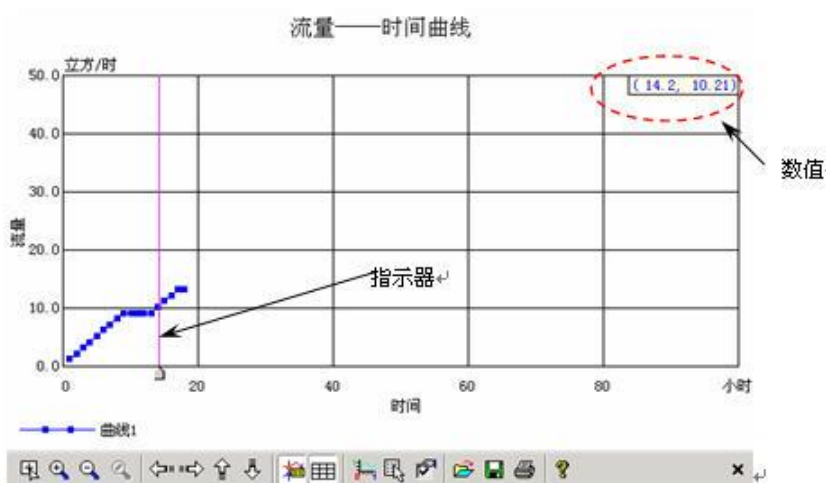






图8.90 超级XY曲线——显示数值

- ◇  显示网格：是否显示图表中的分割线。
- ◇  调整坐标：单击该按钮，弹出调整坐标对话框，可以修改图表的坐标配置。

- ◇  **设置曲线**: 单击该按钮, 弹出设置曲线对话框, 可以修改曲线的属性配置, 如线的颜色、线形等。
- ◇  **选项**: 单击该按钮, 弹出如图8.91所示的选项对话框, 可以修改曲线的配置选项。

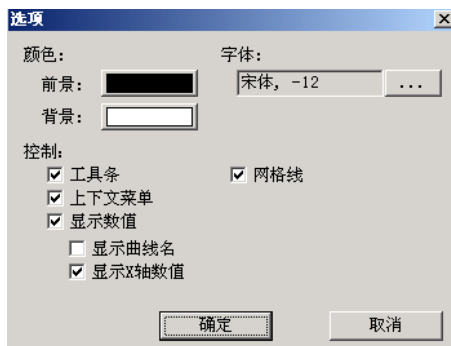






图8.91 选项对话框

- ◇  **从文件中加载**: 从文件中加载一个已经存在的曲线文件, 绘制出曲线来。
- ◇  **保存曲线到文件**: 将当前绘制的曲线保存成文件。
- ◇  **打印**: 打印曲线。点击该按钮时, 弹出打印属性对话框, 可以设置页面属性、选择打印机等。
- ◇  **关闭**: 关闭曲线工具条。

控件方法介绍:

1. void AddNewPoint(float x, float y, long nIndex)

给指定曲线添加一个数据点, 可以在程序开始时定义要显示的曲线

参数: x - 设置数据点的 x 轴坐标值

y - 设置数据点的 y 轴坐标值

nIndex - 给出 X-Y 轴曲线控件中的曲线索引号, 取值范围 0-7

返回值: 无

2. void Clear(long nIndex)

清除一条曲线数据。

参数: nIndex – 同 1

返回值: 无

3. void ClearAll()

清除所有曲线数据。

参数: 无

返回值: 无

4. float GetValueAt(float x, long nIndex)

返回对应于 X 轴点值的 Y 值点插值。

参数: x – X 轴坐标值

nIndex - 同 1

返回值: 相应于 x 值的 Y 轴的值。

5. long GetDataCount(long nIndex)

返回某一曲线点的数目。

参数: nIndex – 同 1

返回值： 返回曲线索引号为 nIndex 的数据点数目。

6. float GetValueX(long nDataIndex, long nCurveIndex)

返回曲线中设定点的 X 轴值。

参数： nDataIndex – 数据点的序号

nCurveIndex – 同 1

返回值： 返回序号为 nDataIndex 的数据点的 X 轴的值

7. float GetValueY(long nDataIndex, long nCurveIndex)

返回曲线中设定点的 Y 轴值。

参数： nDataIndex – 数据点的序号

nCurveIndex – 同 1

返回值： 返回序号为 nDataIndex 的数据点的 Y 轴的值

8. void ZoomIn()

绘图区放大一倍。

参数： 无

返回值： 无

9. void ZoomOut()

绘图区缩小一倍。

参数： 无

返回值： 无

10. void ZoomResume()

恢复原大小。

参数：无

返回值：无

11. void SetClearCurves(BOOL bClear)

功能：主要配合函数 LoadFromFile()进行使用，可以设置在载入其它曲线前是否清除以前的曲线。

参数：BOOL bClear

bClear=0 时不清除，bClear=1 时清除。不进行设置默认为清除

返回值：无

12. void SetXAxesRange(float XMax, float XMin)

设置 X 轴的最大最小值。

参数： Xmax – X 轴的最大值

Xmin – X 轴的最小值

返回值：无

13. void SetYAxesRange(float YMax, float YMin)

设置 Y 轴的最大最小值。

参数： Ymax – Y 轴的最大值

Ymin – Y 轴的最小值

返回值：无

注：该函数只使用于 6.51 之前的版本（单纵轴的控制用），6.51 之后的版本将不

在使用该函数，而使用 `SetIndexYAxesRange` 函数

14. void SetIndexYAxesRange(float YMax, float YMin, long nIndex)

设置 Y 轴的最大最小值。

参数: YMax – Y 轴的最大值

YMin – Y 轴的最小值

nIndex – Y 轴的序号(范围:0-15)

返回值: 无

15. void SetXGrids(long nGrids)

设置 X 轴的分度数。

参数: nGrids – 分度数

返回值: 无

16. void SetYGrids(long nGrids)

设置 Y 轴的分度数。

参数: nGrids – 分度数

返回值: 无

17. int SaveToFile(string szFileName)

保存数据到文件。如果函数提供的保存路径不存在，则自动建立该路径。

参数: szFileName – 文件名

返回值: 返回错误代码。如果成功则返回 0。否则:

-1 – 表示无法打开文件

-2 — 表示文件名无效

-3 — 表示写入异常

18. int LoadFromFile(string szFileName)

从文件中加载数据。

参数: szFileName – 文件名

返回值: 返回错误代码。如果成功则返回 0。否则

-1 — 表示文件无法载入

-2 — 表示文件名无效

-3 — 表示文件格式不正确

-4 — 表示读入错误

19. long LoadHttpFile(CString szHttpName)

功能: 该方法是用来加载网络站点上的 CSV 文件, 并进行曲线显示。

参数: STRING szHttpName: 网络站点上的文件地址

返回值: 返回错误代码。如果成功则返回 0。否则

-1 — 表示文件无法载入

-2 — 表示文件名无效

-3 — 表示文件格式不正确

-4 — 表示读入错误



注意:

通过多次使用 `Long LoadHttpFile(CString szHttpName)` 即可达到加载多个 CSV 文件的目的。

20. `long SetCurvesMapToYAxisIndex(long nCurves, long nYAxisIndex)`

设置指定的曲线同 Y 轴的对应关系

参数: `nCurves` – 曲线序号(范围:0-15)

`nYAxisIndex` – Y 轴序号(范围:0-15)

返回值:

0: 成功

-1: 输入的曲线不再范围内,

-2: 输入的 Y 轴序号不再范围内

21. `long SetCurvesLineStyle(long nCurvesIndex, long nLineStyle)`

设置指定的曲线的线形

参数: `nCurvesIndex` – 曲线序号(范围:0-15)

`nLineStyle` – 线形序号(0-11)

返回值:

0: 成功,0 - 11 的各个线形的序号

-1: 输入的曲线不再范围内

-2: 输入的线形序号不再范围内

22. long SetCurvesColorStyle(long nCurvesIndex, long nColorRed, long nColorGreen, long nColorBlue)

设置指定的曲线的颜色，采用 RGB 格式设置颜色

参数：nCurvesIndex – 曲线序号(范围:0-15)

nColorRed – 红色素值 (0-255)

nColorGreen – 绿色素值 (0-255)

nColorBlue – 兰色素值 (0-255)

返回值：

0: 成功,

-1: 输入的曲线不再范围内

-2: 输入的色素值不再范围内

23. long SetCurvesMarkStyle(long nCurvesIndex, long nMarkStyle)

设置指定的曲线的标记

参数：nCurvesIndex – 曲线序号(范围:0-15)

nMarkStyle – 标注类型(0-8)

返回值：

0: 成功

-1: 输入的曲线不再范围内

-2: 输入的标记类记不再范围内

24. long SetCurvesName(long nCurvesIndex, string strName)

设置指定的曲线的名称

参数: nCurvesIndex – 曲线序号(范围:0-15)

strName – 曲线名称(32 个字节的长度)

返回值:

0: 成功

-1: 输入的曲线不再范围内

-2: 输入的曲线名称长度越界

25. long SetCurvesVisible(long nCurvesIndex, long bVisible)

设置指定的曲线的是否显示

参数: nCurvesIndex – 曲线序号(范围:0-15)

bVisible – 0: 不显示 1: 显示 只能输入 0, 1 值,

返回值:

0: 成功

-1: 输入的曲线不再范围内

-2: 输入的显示值越界

26. long SetIndexYAxexShow(long nIndex, long bShow)

设置指定的 Y 轴是否显示

参数: nIndex – Y 轴序号(范围:0-15)

bShow – 0: 不显示 1: 显示 只能输入 0, 1 值,

返回值:

0: 成功

-1: 输入的曲线不再范围内

-2: 输入的显示值越界

27. long SetXAxesShowTimeFormat(long nShowTime)

设置 X 轴用时间格式显示 (“时: 分: 秒”), 并且最小值为: 0

参数: nShowTime – 是否显示时间 0: 正常按数据显示, 1: 显示时间

返回值 : 0: 成功

1: 输入的数据越界

28. bool SetChartProperty(string szName, string szProperty)

设置 XY 控件的属性。

参数: szName – 属性名

szProperty – 属性值。

29. void PrintChartXY(string szFileName)

打印指定文件。

参数: szFileName—文件名

返回值: 无

第九章 报警和事件系统

- ☞ 介绍报警和事件的相关知识
- ☞ 教您如何配置报警和报警窗口
- ☞ 教您如何使用报警窗口

为保证工业现场安全生产，报警和事件的产生和记录是必不可少的。“组态王”提供了强有力的报警和事件系统，并且操作方法简单。

9.1 关于报警和事件

报警是指当系统中某些量的值超过了所规定的界限时，系统自动产生相应警告信息，表明该量的值已经超限，提醒操作人员。如炼油厂的油品储罐，如果往罐中输油时，如果没有规定油位的上限，系统就产生不了报警，无法有效提醒操作人员，则有可能会造成“冒罐”，形成危险。有了报警，就可以提示操作人员注意。报警允许操作人员应答。

事件是指用户对系统的行为、动作。如修改了某个变量的值，用户的登录、注销，站点的启动、退出等。事件不需要操作人员应答。

组态王中报警和事件的处理方法是：当报警和事件发生时，组态王把这些信息存于内存中的缓冲区中（缓冲区大小可以设置—见本章 9.5.1.1 节），报警和事件在缓冲区中是

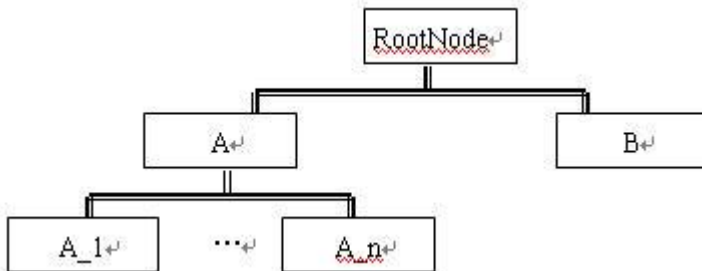
以先进先出的队列形式存储，所以只有最近的报警和事件在内存中。当缓冲区达到指定数目或记录定时时间到时，系统自动将报警和事件信息进记录。报警的记录可以是文本文件、开放式数据库或打印机。另外，用户可以从人机界面提供的报警窗中查看报警和事件信息。

9.2 报警组

9.2.1 报警组概述

往往在监控系统中，为了方便查看、记录和区别，要将变量产生的报警信息归到不同的组中，即使变量的报警信息属于某个规定的报警组。组态王中提供报警组的功能，组态王中最多可以定义 512 个节点的报警组。

报警组是按树状组织的结构，缺省时只有一个根节点，缺省名为 RootNode（可以改成其它名字）。可以通过报警组定义对话框为这个结构加入多个节点和子节点。这类似于树状的目录结构，每个子节点报警组下所属的变量，属于该报警组的同时，属于其上一级父节点报警组。如在上述缺省 RootNode 报警组下添加一个报警组“A”，则属于报警组“A”的变量同时属于“RootNode”报警组。原理图如下图所示。



通过报警组名可以按组处理变量的报警事件，如报警窗口可以按组显示报警事件，记录报警事件也可按组进行，还可以按组对报警事件进行报警确认。

定义报警组后，组态王会按照定义报警组的先后顺序为每一个报警组设定一个 ID 号，在引用变量的报警组域时，系统显示的都是报警组的 ID 号，而不是报警组名称（组态王提供获取报警组名称的函数 `GetGroupName()`）。每个报警组的 ID 号是固定的，当删除某个报警组后，其它的报警组 ID 都不会发生变化，新增加的报警组也不会再占用这个 ID 号。

9.2.2 报警组定义

在组态王工程浏览器的目录树中选择“数据库\报警组”，如图 9.1 所示：

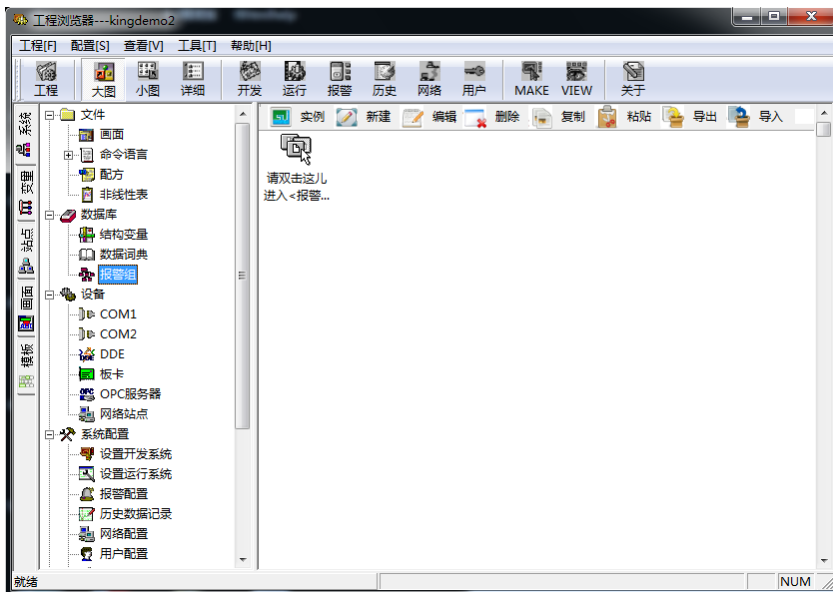


图9.1 报警组定义

双击右侧的“请双击这儿进入<报警组>对话框...”。弹出报警组定义对话框，如图9.2所示：

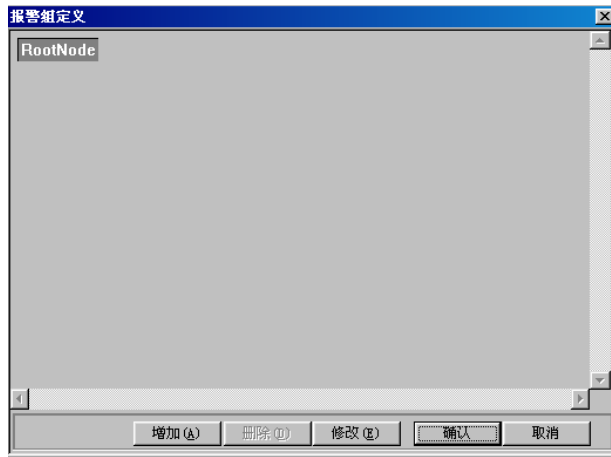
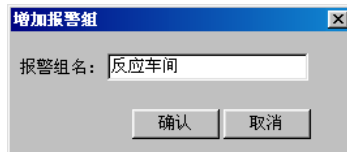


图9.2 报警组定义对话框

对话框中各按钮的作用是：

- ✧ **<增加>按钮**：在当前选择的报警组节点下增加一个报警组节点。
- 如选中图9.2中的“RootNode”报警组，单击<增加>按钮，弹出“增加报警组”对话框，如图9.3所示，在弹出的对话框中输入“反应车间”：



9.3 增加报警组对话框

确定后，在“RootNode”报警组下，会出现一个“反应车间”报警组节点。选

中“RootNode”报警组，单击<增加>按钮，在弹出的增加报警组对话框中输入“炼钢车间”，确定后，在“RootNode”报警组下，会再出现一个“炼钢车间”报警组节点。

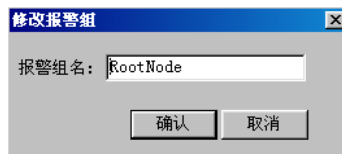
- 选中“反应车间”报警组，单击<增加>按钮，在弹出的增加报警组对话框中输入“液位”，则在“反应车间”报警组下，会出现一个“液位”报警组节点。最终增加结果如图 9.4 所示：



9.4 增加报警组示例

◇ <修改>按钮：修改当前选择的报警组的名称。

选中图 9.2 中的“RootNode”报警组，单击<修改>按钮，弹出如图 9.5 所示的“修改报警组”对话框：



9.5 修改报警组

对话框的编辑框中自动显示原报警组的名称,将编辑框中的内容修改为“企业集团”,然后确定。则原“RootNode”报警组名称变为了“企业集团”。如图 9.6 所示:



9.6 增加和修改后的报警组

- ◇ **<删除>按钮:** 删除当前选择的报警组。

在对话框中选择一个不再需要的报警组,单击<删除>按钮,弹出删除确认对话框,确认后删除当前选择的报警组;如果一个报警组下还包含子报警组,则删除时系统会提示该报警组有子节点,如果确认删除时,该报警组下的子报警组节点也会被删除。

- ◇ **<确认>按钮:** 保存当前修改内容,关闭对话框。
- ◇ **<取消>按钮:** 不保存修改,关闭对话框。

 **注意:**

这里需要注意的是:根报警组(RootNode)只可以修改名称但不可删除。

9.3 如何定义变量的报警属性

在使用报警功能前,必须先要对变量的报警属性进行定义。组态王的变量中模拟型(包括整型和实型)变量和离散型变量可以定义报警属性。下面一一介绍。

9.3.1 通用报警属性功能介绍

在组态王工程浏览器“数据库/数据词典”中新建一个变量或选择一个原有变量双击它,在弹出的“定义变量”对话框上选择“报警定义”属性页,如图9.7所示。



9.7 通用报警属性

报警属性页可以分为以下几个部分:

- ◆ **报警组名和优先级选项:** 单击“报警组名”标签后的按钮,会弹出“选择报警组”对话框,在该对话框中将列出所有已定义的报警组,选择其一,确认后,则该变量的报警信息就属于当前选中的报警组。如图9.7中选择“反应车间”,则当前定义

的变量就属于反应车间报警组，这样在报警记录和查看时直接选择要记录或查看的报警组为“反应车间”，则可以看到所有属于“反应车间”的报警信息。

优先级主要是指报警的级别，主要有利于操作人员区别报警的紧急程度。报警优先级的范围为 1~999，1 为最高，999 最低。在图 9.7 的优先级编辑框中输入当前变量的报警优先级。

- ◇ **模拟量报警定义区域：**如果当前的变量为模拟量，则这些选项是有效的，具体定义方法在 9.3.2 节中详细介绍。
- ◇ **开关量报警定义区域：**如果当前的变量为离散量，则这些选项是有效的，具体定义方法在 9.3.3 节中详细介绍。
- ◇ **报警的扩展域的定义：**报警的扩展域共有两个，主要是对报警的补充说明、解释。在报警产生时的报警窗中可以看到。

在介绍报警类型之前，先介绍关于报警的三个概念：

1. 报警产生：变量值的变化超出了定义的正常范围，处于报警区域
2. 报警确认：对报警的应答，表示已经知道有该报警，或已处理过了，报警进行确认后，报警状态并不消失
3. 报警恢复：变量的值恢复到定义的正常范围，不再处于报警区域

9.3.2 模拟型变量的报警类型

模拟型变量主要是指整型变量和实型变量，包括内存型和 IO 型的。模拟型变量的报警类型主要有三种：越限报警、偏差报警和变化率报警。对于越限报警和偏差报警可以定

义报警延时和报警死区。

9.3.2.1 越限报警

模拟量的值在跨越规定的高低报警限时产生的报警。越限报警的报警限共有四个：低低限、低限、高限、高高限。其原理图如图 9.8 所示。



9.8 越限报警原理图

在变量值发生变化时，如果跨越某一个限值，立即发生越限报警，某个时刻，对于一个变量，只可能越一种限，因此只产生一种越限报警，例如：如果变量的值超过高高限，就会产生高高限报警，而不会产生高限报警。另外，如果两次越限，就得看这两次越的限是否是同一种类型，如果是，就不再产生新报警，也不表示该报警已经恢复；如果不是，则先恢复原来的报警，再产生新报警。越限报警产生和恢复的算法为：

- ◇ 大于低低限时恢复低低限，小于等于低低限时产生报警
- ◇ 大于低限时恢复低限，小于等于低限时报警产生报警
- ◇ 大于等于高限时报警，小于高限时恢复高限
- ◇ 大于等于高高限时报警，小于高高限时恢复高高限

越限类型的报警可以定义其中一种，任意几种或全部类型。如图 9.9 中所示为越限报警定义。有“界限值”和“报警文本”两列。



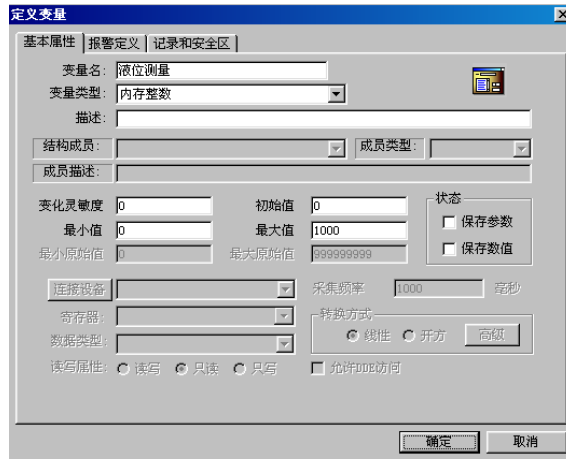
9.9 越限报警定义

界限值列中选择要定义的越限类型，则后面的界限值和报警文本编辑框变为有效。在界限值中输入该类型报警越限值，定义界限值时应该：最小值 \leq 低低限值 $<$ 低限 $<$ 高限 $<$ 高高限 \leq 最大值。在报警文本中输入关于该类型报警的说明文字，报警文本不超过 15 个字符。

💡 例如：要对液位变量设定报警限值，要求液位的高高报警值=900，高报警值=750，

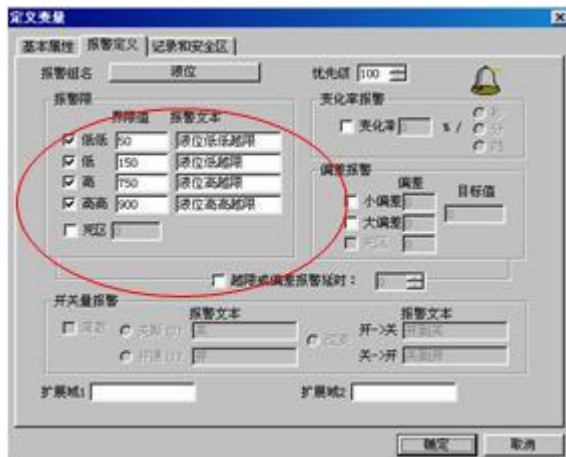
低报警值= 150，低低报警值= 50，一般可按如下步骤进行：

1. 在数据词典中新建内存整型变量，在变量的基本属性中设置变量名称“液位测量”，变量类型选择“内存整型”（一般为 I0 变量，这里定义内存型，只为说明操作方法），定义其最小值为 0，最大值为 1000。定义后的基本属性如图 9.10 所示：





9.10 越限报警变量基本属性定义

2. 选择变量定义对话框的“报警属性”页，如图 9.11 所示。选择“报警限”项目中的“低低”，后面的“界限值”和“报警文本”编辑框变为有效，在界限值中输入 50，在报警文本编辑框中输入“液位低低报警”。依次类推，分别选择其它几个项目，输入如图 9.11 所示的界限值和报警文本。



9.10 越限报警变量越限属性定义

3. 定义报警组和优先级。单击“报警组名称”后的按钮，在弹出的“选择报警组”对话框中选择 9.2 节中定义的报警组“液位”，在“优先级”编辑框中输入优先级为 100。
4. 单击“确定”，完成报警定义。
5. 新建一个画面，在工具箱中选择“报警窗口”按钮，在画面上创建报警窗，双击报警窗口，在报警窗口名编辑框中输入“越限报警窗”，选择“历史报警窗”选项。报警窗的具体配置使用参见 9.5.1 节。然后确定。
6. 在工具箱上选择按钮，在画面上添加一个文本，双击该文本，定义该文本的动画连接，在模拟值输入中选择变量“液位测量”并确定。在模拟值输入中同样选择该变量，“值范围”最大值为 1000，最小值为 0。建立完动画连接后，保存当前画面。
7. 在画面上单击右键，在弹出的菜单中选择“切换到 View”，进入组态王运行系统。打开刚才的画面。如图 9.12 所示。

8. 在画面上液位测试变量输入值 5，报警窗口中出现一条报警信息，如图 9.13 所示。然后分别输入 100、146、800、900。则会产生一系列的报警，在报警窗中显示出来。如图 9.14 所示。可以看到，当数据小于等于 50 时，产生低低超限报警；当数据大于 50 小于等于 150 时，恢复低低限报警，产生低限超限报警；当数据大于 150 小于 750 时，恢复低限报警，此时该变量没有报警；当数据大于等于 750 小于 900 时，产生高限超限报警；当数据大于等于 900 时，恢复高限报警，产生高高限超限报警。反之，当数据逐步减小时，在相应的区域也会产生相应的报警和恢复。



9.12 超限报警画面 1



9.13 超限报警画面 2

事件日期	事件时间	报警日期	报警时间	报警标志	报警类型	报警值/限值	报警值/限值	报警值	报警等级	报警标志	扩展码
—	—	02/11/22	13:59:52.000	液位报警	液位报警故障	3.0	—	30.0	100	报警	
02/11/22	13:59:58.240	02/11/22	13:59:52.000	液位报警	液位报警故障	3.0	—	348.0	100	报警	
—	—	02/11/22	13:59:58.240	液位报警	液位报警故障	148.0	—	350.0	100	报警	
02/11/22	13:20:17.290	02/11/22	13:20:18.141	液位报警	液位报警故障	148.0	—	360.0	100	报警	
—	—	02/11/22	13:20:23.990	液位报警	液位报警故障	800.0	—	370.0	100	报警	
02/11/22	13:29:33.130	02/11/22	13:29:23.990	液位报警	液位报警故障	800.0	—	360.0	100	报警	
—	—	02/11/22	13:29:33.130	液位报警	液位报警故障	900.0	—	360.0	100	报警	

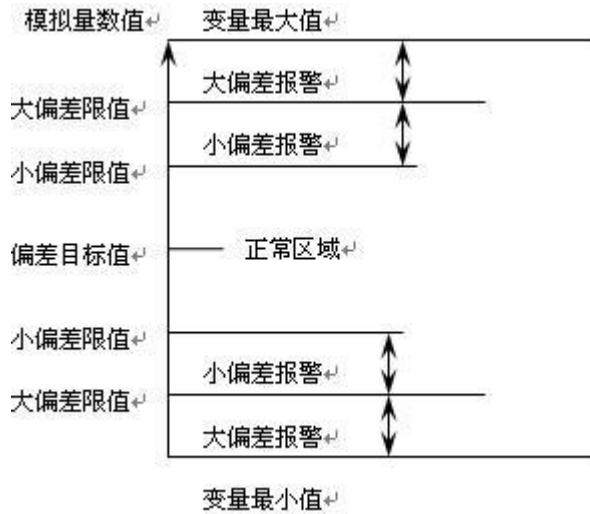
液位测量 900

9.14 越限报警画面3

9.3.2.2 偏差报警

模拟量的值相对目标值上下波动超过指定的变化范围时产生的报警。偏差报警可以分为小偏差和大偏差报警两种。当波动的数值超出大小偏差范围时，分别产生大偏差报警和小偏差报警，其原理图如图 9.15 所示。偏差报警限的计算方法为：

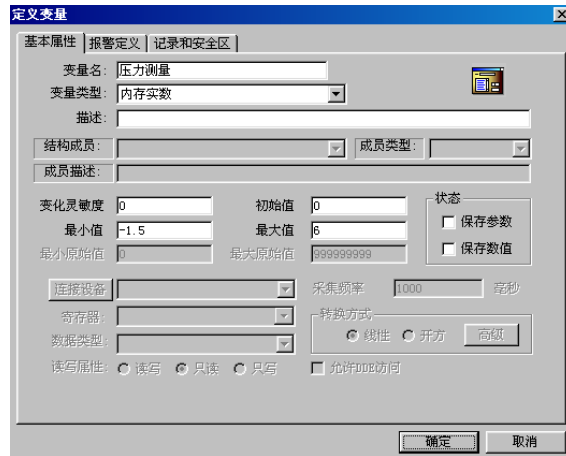
- ◇ 小偏差报警限=偏差目标值±定义的小偏差
- ◇ 大偏差报警限=偏差目标值±定义的大偏差
- ◇ 大于等于小偏差报警限时，产生小偏差报警
- ◇ 大于等于大偏差报警限时，产生大偏差报警
- ◇ 小于等于小偏差报警限时，产生小偏差报警
- ◇ 小于等于大偏差报警限时，产生大偏差报警
- ◇ 偏差报警在使用时可以按照需要定义一种偏差报警或两种都使用。
- ◇ 变量变化的过程中，如果跨越某个界限值，则立刻会产生报警，而同一时刻，不会产生两种类型的偏差报警。



9.15 偏差报警原理图

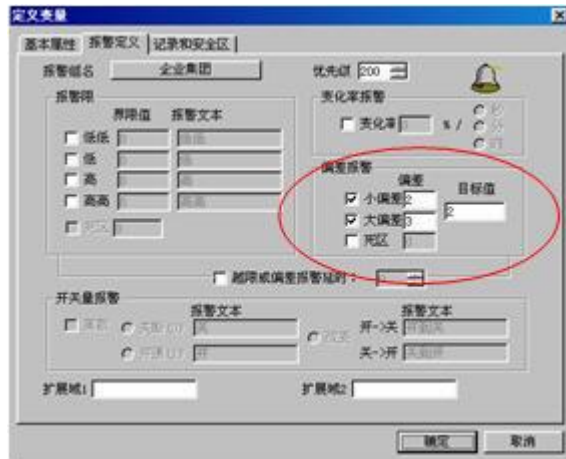
例如：某一工序中要求压力在一定的范围内，不能太大，也不能太小，这是可以定义偏差报警来确定压力的值是否在要求的范围内。

1. 在数据词典中新建内存实型变量，在变量的基本属性中设置变量名称“压力测量”，变量类型选择“内存实型”（一般为 I0 变量，这里定义内存型，只为说明操作方法），定义其最小值为-1.5，最大值为 6。定义后的基本属性如图 9.16 所示。



9.16 偏差报警基本属性定义

2. 选择定义变量对话框的“报警定义”属性页。选择“偏差报警”组中的小偏差和大偏差选项，则小偏差和大偏差的限值编辑框变为有效。在“偏差目标值”编辑框中输入目标值 2；在小偏差限值中输入值 2，在大偏差限值中输入值 3。
3. 选择相应的报警组和优先级，如图 9.17 所示。定义完成后，确定关闭对话框。
4. 在 9.3.2.1 节建立的画面中在创建一个文本，定义动画连接，模拟值输出、值输入连接的变量为“压力测量”，值输入定义值输入范围为-1.5~6。保存画面。
5. 修改变量的值，数据值增加：当数据变化到 4 (2+2) 时，产生小偏差报警；变化到 5 (2+3) 时，恢复小偏差报警，产生大偏差报警。数据值减小：当数据小于 5 时，恢复大偏差报警，产生小偏差报警；当数据小于 4 时，恢复小偏差报警，没有报警；当数据小于 0 (2-2) 时，产生小偏差报警；当数据小于-1 (2-3) 时，恢复小偏差报警，产生大偏差报警。



9.17 偏差报警报警属性定义

9.3.2.3 变化率报警

变化率报警是指模拟量的值在一段时间内产生的变化速度超过了指定的数值而产生的报警，即变量变化太快时产生的报警。系统运行过程中，每当变量发生一次变化，系统都会自动计算变量变化的速度，以确定是否产生报警。变化率报警的类型以时间为单位分为三种：%x/秒、%x/分、%x/时。变化率报警的计算公式如下：

$$\left(\left(\text{变量的当前值} - \text{变量上一次变化的值} \right) \times 100 \right) / \left(\left(\text{变量本次变化的时间} - \text{变量上一次变化的时间} \right) \times \left(\text{变量的最大值} - \text{变量的最小值} \right) \times \left(\text{报警类型单位对应的值} \right) \right)$$

其中报警类型单位对应的值定义为：如果报警类型为秒，则该值为 1；如果报警类型为分，则该值为 60；如果报警类型为时，则该值为 3600。取计算结果的整数部分的绝对值作为结果，若计算结果大于等于报警极限值，则立即产生报警。变化率小于报警极限值时，报

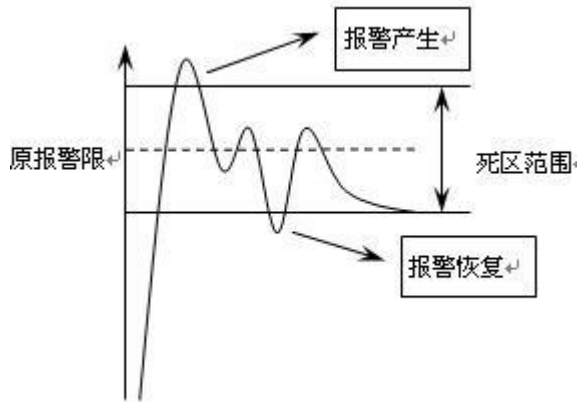
警恢复。变化率报警定义如图 9.18 所示。选择变化率选项，在编辑框中输入报警极限值，选择报警类型的单位。



9.18 变化率报警定义

9.3.2.4 报警延时和死区

对于越限和偏差报警，可以定义报警死区和报警延时。报警死区的原理图如图 9.19 所示。报警死区的作用是为了防止变量值在报警限上下频繁波动时，产生许多不真实的报警，在原报警限上下增加一个报警限的阈值，使原报警限界线变为一条报警限带，当变量的值在报警限带范围内变化时，不会产生和恢复报警，而一旦超出该范围时，才产生报警信息。这样对消除波动信号的无效报警有积极的作用。



9.19 报警死区原理图

例如：针对 9.3.2.1 节中的例子，对“液位测量”变量的越限报警进行报警死区的定义，原要求为液位的高高报警值=900，高报警值=750，低报警值= 150，低低报警值= 50。现在对报警限增加死区，死区值为 5。操作步骤如下：

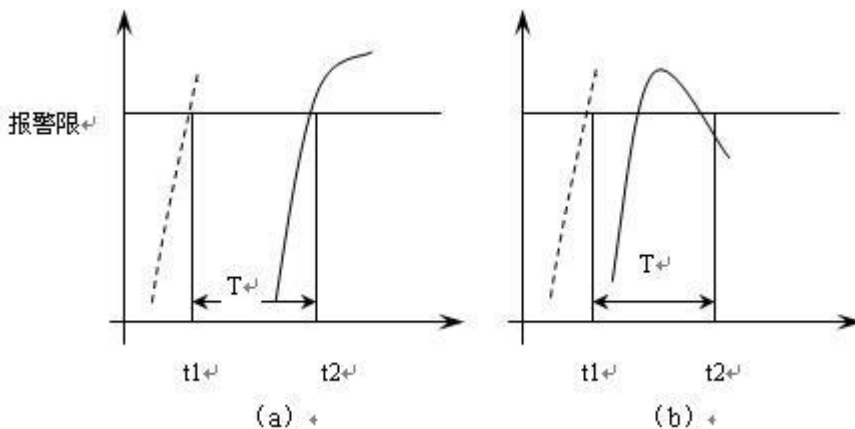
1. 在组态王的数据词典中重新定义变量“液位测量”的报警属性，如图 9.20 所示，选择报警限中“死区”选项，在编辑框中输入死区值 5，单击“确定”关闭对话框。
2. 切换到组态王运行系统，修改“液位测量”变量的值，当数据变化时，产生报警的界限值为：



9.20 越限死区值定义

	低低限	低限	高限	高高限
报警值	≤ 45	≤ 145	≥ 755	≥ 905
恢复值	> 55	> 155	< 745	< 895

对于偏差报警死区的定义和使用与越限报警大致相同，这里不在讲述。报警延时是对系统当前产生的报警信息并不提供显示和记录，而是进行延时，在延时时间到后，如果该报警不存在了，表明该报警可能是一个误报警，不用理会，系统自动清除；如果延时到后，该报警还存在，表明这是一个真实的报警，系统将其添加到报警缓冲区中，进行显示和记录。如果定时期间，有新的报警产生，则重新开始定时。报警延时原理图如图 9.21 所示。图中虚线表示变量刚产生报警时的变量曲线，实线表示延时后的变量曲线。变量数据变化在 t_1 时刻产生报警，在图 (a) 中，变量的值经过延时时间 T 后，依然在报警限之上，这是系统产生报警；在图 (b) 中，变量的值经过延时时间 T 后，已经恢复到了报警限之下（类似于毛刺信号），系统不再产生这个报警。报警延时的时间单位为秒。组态王中，同一个变量的越限和偏差报警使用同一个报警延时时间。



9.21 报警延时原理图

例如：

针对 9.3.2.2 节中的例子，对“压力测量”变量的偏差报警进行报警延时的定义。要求报警要延时 2 秒，防止误报警。

1. 在组态王数据词典中重新定义“压力测量”变量的报警属性，如图 9.22 所示，选中“越限或偏差报警延时”选项，在其后面的编辑框中输入延时时间：2。单击“确定”按钮，关闭对话框。
2. 切换到组态王运行系统，修改变量的值，使压力测量=4.5，这是系统不会立刻产生报警，保持该值，则 2 秒后，系统产生小偏差报警。再次修改变量的值，使压力测量=3.5，这是系统不会立刻恢复报警，保持该值，则 2 秒后，系统恢复小偏差报警。
3. 修改变量的值，使压力测量=5，这是系统不会立刻产生报警，在 2 秒内修改变量的值使压力测量=3，则 2 秒后，系统不会再产生该报警。



9.22 报警延时定义

9.3.3 离散型变量的报警类型

离散量有两种状态：1、0。离散型变量的报警有三种状态：

- ◇ 1 状态报警：变量的值由 0 变为 1 时产生报警
- ◇ 0 状态报警：变量的值由 1 变为 0 时产生报警
- ◇ 状态变化报警：变量的值有 0 变为 1 或由 1 变为 0 为都产生报警

离散量的报警属性定义如图 9.23 所示。在报警属性页中报警组名、优先级和扩展域的定义与模拟量定义相同。在“开关量报警”组内选择“离散”选项，三种类型的选项变为有效。定义时，三种报警类型只能选择一种。选择完成后，在报警文本中输入不多于 15 个字符的类型说明。



9.23 离散型变量的报警属性定义

9.4 事件类型及使用的方法

事件是不需要用户来应答的。组态王中根据操作对象和方式等的不同，事件分为以下几类：

- ◇ 操作事件：用户对变量的值或变量其它域的值进行修改
- ◇ 登录事件：用户登录到系统，或从系统中退出登录
- ◇ 工作站事件：单机或网络站点上组态王运行系统的启动和退出
- ◇ 应用程序事件：来自 DDE 或 OPC 的变量的数据发生了变化

事件在组态王运行系统中人机界面的输出显示是通过历史报警窗实现的。

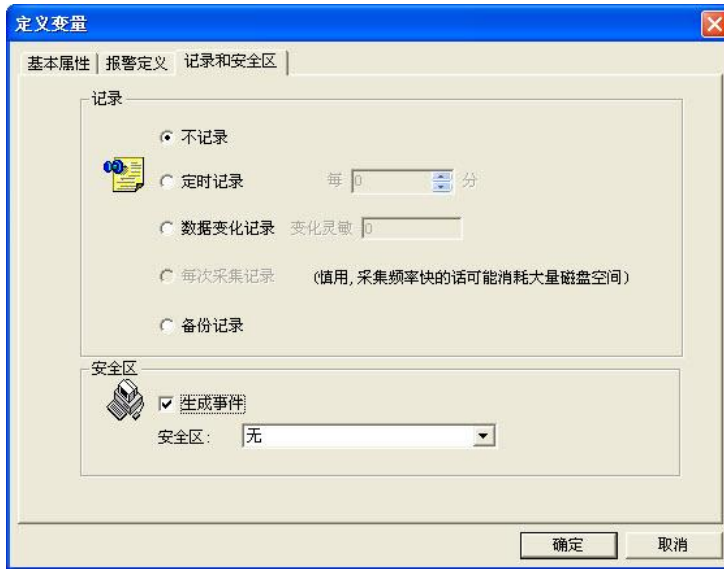
9.4.1 操作事件

操作事件是指用户修改有“生成事件”定义的变量的值或其域的值进行修改时，系统

产生的事件。如修改重要参数的值，或报警限值、变量的优先级等。这里需要注意的时，同报警一样，字符串型变量和字符串型的域的值值的修改不能生成事件。操作事件可以进行记录，使用户了解当时的值是多少，修改后的值是多少。

变量要生成操作事件，必须先要定义变量的“生成事件”属性。

- 1、在组态王数据词典中新建内存整型变量“操作事件”，选择“定义变量”的“记录和安全区”属性页，如图 9.24 所示，在“安全区”栏中选择“生成事件”选项。单击“确定”，关闭对话框。
- 2、新建画面，在画面上创建一个文本，定义文本的动画连接——模拟值输入和模拟值输出连接，选择连接变量为“操作事件”。再创建一个文本，定义文本的动画连接——模拟值输入和模拟值输出连接，选择连接变量为“操作事件”的优先级域“Priority”。
- 3、在画面上创建一个报警窗，定义报警窗的名称为“事件”，类型为“历史报警窗”。保存画面，切换到组态王运行系统。
- 4、打开该画面，分别修改变量的值和变量优先级的值，系统产生操作事件，在报警窗中显示，如图 9.25 所示。报警窗中第二行为修改变量的值的操作事件，其中事件类型为“操作”，域名为“值”；第三行为修改变量优先级的值，域名为“优先级”。另外，还可以看到旧值和新值。



9.24 变量定义“生成事件”

事件日期	事件时间	事件类型	变量名	报警值/旧值	恢复值/新值	界限值	优先级	报警组名	域名	操作员
02/11/25	16:10:31.620	启动	—	—	—	—	—	—	—	—
02/11/25	16:10:43.920	操作	操作事件	2.0	7.0	—	88	—	值	无
02/11/25	16:10:56.880	操作	操作事件	200.0	88.0	—	88	—	优先级	无

9.25 生成的操作事件

9.4.2 用户登录事件

用户登录事件是指用户向系统登录时产生的事件。系统中的用户，可以在工程浏览器——用户配置中进行配置，如用户名、密码、权限等。

用户登录时，如果登录成功，则产生“登录成功”事件；如果登录失败或取消登录过

程，则产生“登录失败”事件；如果用户退出登录状态，则产生“注销”事件。

💡 例如：

使用 9.4.1 节中的工程例子，切换到组态王运行系统，打开画面。选择菜单“特殊/登录开”，在弹出的用户登录对话框中选择“系统管理员”，输入密码，单击确定，产生登录成功事件；如果同样选择该用户，在登录对话框上选择取消，产生登录失败事件；选择菜单“特殊/登录关”，产生注销事件。如图 9.26 所示。

事件日期	事件时间	事件类型	变量名	报警值/旧值	恢复值/新值	界限值	优先级	报警组名	域名	操作员
02/11/25	16:55:10.940	启动	—	—	—	—	—	—	—	—
02/11/25	16:55:23.570	注销	—	—	—	—	—	—	—	无
02/11/25	16:55:23.630	登录成功	—	—	—	—	—	—	—	系统管理员
02/11/25	16:55:38.180	登录失败	—	—	—	—	—	—	—	系统管理员
02/11/25	16:55:44.440	注销	—	—	—	—	—	—	—	系统管理员

9.26 登录事件

9.4.3 应用程序事件

如果变量是 I/O 变量，变量的数据源为 DDE 或 OPC 服务器等应用程序，对变量定义“生成事件”属性（如图 9.24）后，当采集到的数据发生变化时，产生该变量的应用程序事件。

💡 例如：

建立一个 EXCEL 的 DDE 设备的变量，产生该变量的应用程序事件。

- 1、在组态王中新建“DDE”设备，设备的逻辑名称为“Excel 设备”，服务程序名称为“Excel”，话题名为“Sheet1”。
- 2、在数据词典中新建变量，变量名称为“DDE 事件”，变量类型为 I/O 实型，变量连接的

设备为“Excel 设备”，项目名称为“rlc1”。

- 3、在变量的“记录和安全区”属性页中选择“生成事件”选项，单击确定，关闭对话框。
- 4、在 9.4.1 节中建立的画面中创建一个文本，并建立动画连接——模拟值输出，关联的变量为“DDE 事件”。保存画面，启动 Excel，切换到组态王运行系统，打开该画面。
- 5、修改 Excel 的 Sheet1 工作表的 R1C1 单元格中的数据，每当组态王检测到数据变化时，产生应用程序事件，如图 9.27 所示。

事件日期	事件时间	事件类型	变量名	报警值/旧值	恢复值/新值	界限值	优先级	报警组名	域名	操作员
02/11/26	09:31:14.050	启动	—	—	—	—	—	—	—	—
02/11/26	09:31:33.630	应用程序	DDE事件	0.0	1.0	—	1	—	值	—
02/11/26	09:31:42.230	应用程序	DDE事件	1.0	8.8	—	1	—	值	—
02/11/26	09:32:12.330	应用程序	DDE事件	8.8	22.0	—	1	—	值	—

9.27 应用程序事件

9.4.4 工作站事件

所谓工作站事件就是指某个工作站站点上的组态王运行系统的启动和退出事件，包括单机和网络。组态王运行系统启动，产生工作站启动事件；运行系统退出，产生退出事件。如图 9.27 报警窗中第一条信息为工作站启动事件。

9.5 如何记录、显示报警

组态王中提供了多种报警记录和显示的方式，如报警窗、数据库、打印机等。系统提供一个预定的缓冲区，对产生的报警信息首先保存在缓冲区中，报警窗根据定义的条件，从缓冲区中获取符合条件的信息显示。当报警缓冲区满或组态王内部定时时间到时，将信

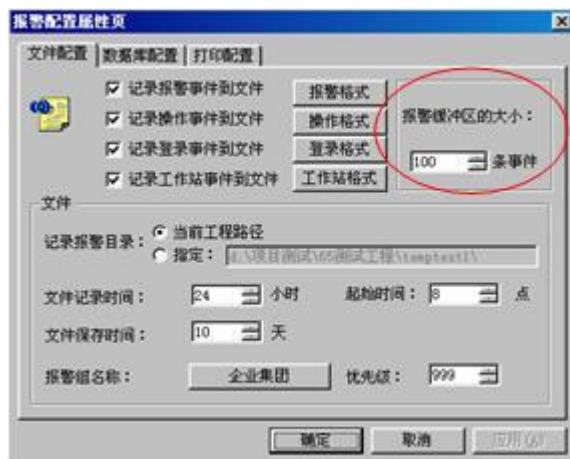
息按照配置的条件进行记录。

9.5.1 报警输出显示：报警窗口

组态王运行系统中报警的实时显示是通过报警窗口实现的。报警窗口分为两类：实时报警窗和历史报警窗。实时报警窗主要显示当前系统中存在的符合报警窗显示配置条件的实时报警信息和报警确认信息，当某一报警恢复后，不再在实时报警窗中显示。实时报警窗不显示系统中的事件。历史报警窗显示当前系统中符合报警窗显示配置条件的所有报警和事件信息。报警窗口中最大显示的报警条数取决于报警缓冲区大小的设置。

9.5.1.1 报警缓冲区大小的定义

报警缓冲区是系统在内存中开辟的用户暂时存放系统产生的报警信息的空间，其大小是可以设置的。在组态王工程浏览器中选择“系统配置/报警配置”，双击后弹出“报警配置属性页”，如图 9.28 所示，在对话框的右上角为“报警缓冲区的大小”设置项，报警缓冲区大小设置值按存储的信息条数计算，值的范围为 1~10,000。报警缓冲区大小的设置直接影响着报警窗显示的信息条数。



9.28 报警缓冲区大小设置

9.5.1.2 创建报警窗口

在组态王中新建画面，在工具箱中单击报警窗口按钮，如图 9.29 所示，或选择菜单“工具\报警窗口”，鼠标箭头变为单线“十”字型，在画面上适当位置按下鼠标左键并拖动，绘出一个矩形框，当矩形框大小符合报警窗口大小要求时，松开鼠标左键，报警窗口创建成功，如图 9.30 所示。

改变报警窗在画面上的位置时，将鼠标移动到选中的报警窗的边缘，当鼠标箭头变为双“十”字型时，按下鼠标左键，拖动报警窗口，到合适的位置，松开鼠标左键即可。

选中的报警窗口周围有 8 个带箭头的小矩形，将鼠标移动到小矩形的上方，鼠标箭头变为双向箭头时，按下鼠标左键，并拖动，可以修改报警窗的大小。



9.29 报警窗口按钮

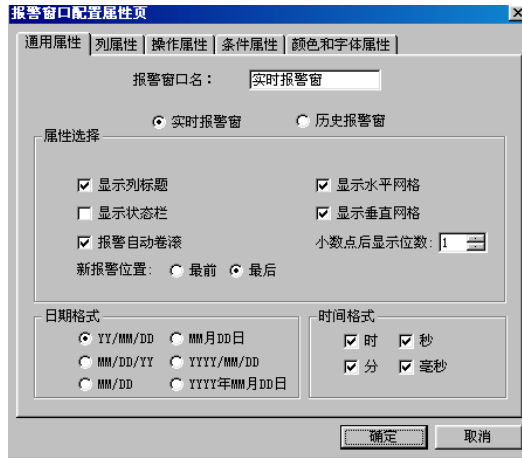
事件日期	事件时间	报警日期	报警时间	变量名	报警类型	报警

9.30 报警窗口

9.5.1.3 如何配置实时和历史报警窗

报警窗口创建完成后，要对其进行配置。双击报警窗口，弹出报警窗口配置属性页，如图 9.31 所示，首先显示的是通用属性页。在该页中有一个实时报警窗和历史报警窗的选项，选择当前报警窗是哪一个类型：如果选择“实时报警窗”，则当前窗口将成为实时

报警窗；否则，如果选择“历史报警窗”，则当前窗口将成为历史报警窗。实时和历史报警窗的配置选项大多数相同。在本节的说明中，如果没有特殊说明，则配置选项为公用选项。



9.31 报警窗口配置属性页——通用属性

1、通用属性页

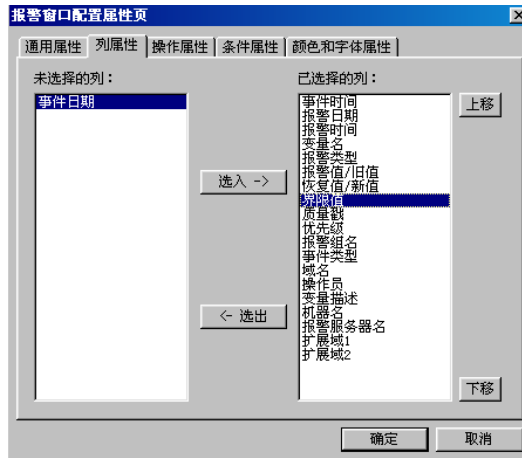
- ◇ **报警窗口名：**定义报警窗口在数据库中的变量登记名。此报警窗口变量名可在为操作报警窗口建立的命令语言连接程序中使用。报警窗口名的定义应该符合组态王变量的命名规则。
- ◇ **属性选择：**属性选择有七项选项：
 - 是否显示列标题：选中后，开发和运行中在窗口的上部均出现每一列的列标题。如显示报警时间的列的上部，会有标题显示“报警时间”。
 - 是否显示状态栏：选中后，开发和运行中在窗口的下部均出现报警窗的状态信

息栏。状态栏中显示当前报警窗中报警条数等

- **报警自动卷滚：**选中后，系统运行时，如果报警窗中的信息显示超过当前窗口一页显示，当出现新的报警时，报警窗会自动滚动，显示新报警。
 - **是否显示水平网格：**选中后，开发和运行中在窗口的信息显示部位均出现水平网格线。
 - **是否显示垂直网格：**选中后，开发和运行中在窗口的信息显示部位均出现垂直网格线。
 - **小数点后显示位数：**定义报警窗中数据显示部分各种数据显示时的小数位数。
 - **新报警出现位置：**产生一条报警或事件后，显示到报警窗口的位置。“最前”为新报警出现在报警窗口的最上方，先前显示的报警在窗口中依次向下移动一行；“最后”为新报警出现在报警窗的最后一行。
- ◇ **日期格式：**选择报警窗中日期的显示格式，只能选择一项。
- ◇ **时间格式：**选择报警窗中时间的显示格式，即显示时间的哪几个部分。如“xx 分 xx 秒”、或“xx 时 xx 分 xx 秒”。该选择应该符合逻辑，例如只选择时和秒是错误的，时间格式选择错误时，系统会提示“时间格式不对”。

2、列属性页

单击报警窗口配置属性页中的“列属性”标签，设置报警窗口的列属性，如图 9.32 所示。



9.32 报警窗口配置属性页——列属性

列属性主要配置报警窗口究竟显示哪些列,以及这些列的顺序,这就是所谓的列属性。

各项意义如下:

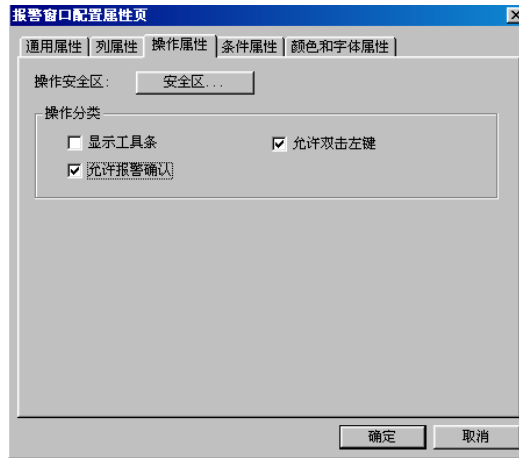
- ◇ **未选择的列:** 在此框中出现的列将不在报警窗口中显示。
- ◇ **已选择的列:** 在此框中出现的列将在报警窗口中显示。
- ◇ **选入:** 选入某一列项,在报警窗中显示,在“未选择的列”中选一列名称后,单击“选入”按钮就可将列选入到“已选择的列”中,即可在报警窗中显示。注意,必须在左边**未选择的列**框中选择某一列后,才可以执行此功能,
- ◇ **选出:** 将“已选择的列”选出,不在报警窗中显示,在“未选择的列”中选一列名称后,单击“选出”按钮就可将列选出到“未选择的列”中,即不在报警窗中显示。注意,必须在右边**已选择的列**框选择某一个列名后,才可以执行此功能。
- ◇ **上移:** 选入的列的排序,上移某一列名,必须在右边框选择某一个列名后并且选择

的列名不在右边框的最前面时，才能执行此功能。

- ◇ **下移：**选入的列的排序，下移某一列名，必须在右边框选择某一个列名后并且选择的列名不在右边框的最后面时，才能执行此功能。

3、操作属性页

单击报警窗口配置属性页中的“操作属性”标签，设置报警窗口的操作属性，如图 9.33 所示。



9.33 报警窗口配置属性页——操作属性

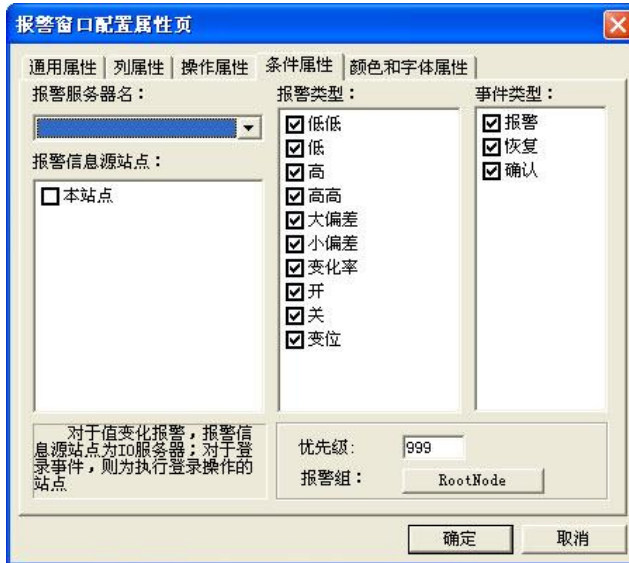
- ◇ **操作安全区：**配置报警窗口在运行系统中的操作权限——允许操作该报警窗的安全区。安全区可以选择多个。
- ◇ **操作分类：**配置报警窗口在运行时支持的操作内容和方式：
- ◇ **允许报警确认：**系统运行时，允许通过图标等操作方式对报警进行确认
- ◇ **显示工具条：**选中时，开发和运行中在报警窗顶部显示快捷按钮，并允许用户在系

统运行时通过图标操作报警窗

- ◇ **允许双击左键：**系统运行时，允许在某一报警条上双击左键执行预置自定义函数功能。

4、条件属性页

单击报警窗口配置属性页中的“条件属性”标签，设置报警窗口的报警信息显示的过滤条件，如图 9.34 所示。条件属性在运行期间可以在线修改。



9.34 报警窗口配置属性页——条件属性

- ◇ **报警服务器名：**如果系统为单机模式，则该选项不用选择。如果为网络模式，网络中各站点的报警信息存在于报警服务器上，配置网络后，指定本机是哪些报警服务器的客户（网络配置请参见第二十一章 网络功能）。该列表框中将列出所有本机的

报警服务器，可指定将哪个报警服务器上的报警信息显示在该报警窗中。

- ◇ **报警信息源站点：**如果系统为单机模式，默认为本地，该选项不用选择。如果为网络模式，因为数据的报警信息最初是来自于 I/O 服务器，网络配置后，指定本机是哪些 I/O 服务器的客户（网络配置请参见第二十一章 网络功能），该列表框中将列出所有本机当前选择的报警服务器下的 I/O 服务器名称，可选择将当前报警服务器下的哪些 I/O 服务器上的报警信息显示在该报警窗中。本项可以多选。
- ◇ **优先级：**选择报警窗中要显示的报警和事件的最高优先级条件，高于设定优先级的报警和事件将显示在报警窗中。如规定优先级条件为 500，则 1~500 间的报警和事件信息将显示在报警窗中，而优先级在 501~999 间的报警和事件将不显示。优先级选择的范围为 1 至 999 之间的整数。
- ◇ **报警组名：**选择报警窗中要显示的报警和事件的报警组条件，选择的报警组及其下子报警组的报警和事件允许显示在报警窗中，只能选择一个报警组。
- ◇ **报警类型：**选择报警窗口中允许显示何种类型的报警。
- ◇ **事件类型：**选择报警窗口中允许显示何种类型的事件。**注意：对于实时报警窗，事件类型选项不可用。只能在运行系统中进行修改，实时报警窗的事件类型只有三种：报警、确认和恢复。**

5、颜色和字体属性页


单击报警窗口配置属性页中的“颜色和字体属性”标签，设置报警窗口的报警和事件信息显示的字体颜色和字体型号、字体大小等，如图 9.35 所示。



9.35 报警窗口配置属性页——颜色和字体属性

单击标签后面的颜色按钮，弹出颜色选择对话框，选择该标签信息要显示的颜色；单击标签后面的“字体”按钮，弹出字体选择对话框，选择标签对应信息显示时的显示字体等。

例如：

定义工作站启动事件显示时信息的颜色和字体。单击“工作站启动”后的按钮  弹出调色板，如图 9.36 所示：

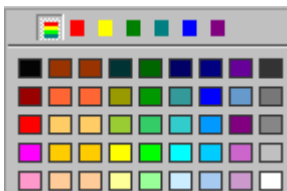



图9.36 调色板

选中一种颜色后，按钮上出现选中的颜色。调色板自动关闭。单击按钮  弹出字体选择对话框，如图 9.37 所示选择完字体后，单击确定，关闭对话框。

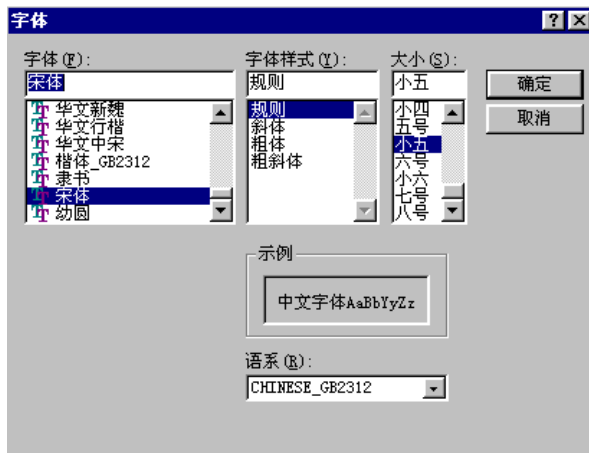


图 9.37 字体选择对话框

9.5.1.4 运行系统中报警窗的操作


如果报警窗配置中选择了“显示工具条”和“显示状态栏”，则运行时的标准报警窗显示如图 9.38 所示。







图 9.38 运行系统标准报警窗

标准报警窗共分为三个部分：工具条、报警和事件信息显示部分、状态栏。

1、工具条中按钮的作用为：

- ◇  确认报警：在报警窗中选择未确认过的报警信息条，该按钮变为有效，单击该按钮，确认当前选择的报警。
- ◇  报警窗暂停/恢复滚动：每单击一次该按钮，暂停/恢复滚动状态发生一次变化。假如在报警窗中不断滚动显示报警时，可以单击该按钮暂停滚动，仔细查看某条报警，然后在单击该按钮，继续滚动，报警窗的暂停滚动并不影响报警的产生等，恢复滚动后，在暂停期间没有显示出来的报警会全部显示出来。暂停和恢复滚动在状态栏第三栏有相应显示。
- ◇  更改报警类型：更改当前报警窗显示的报警类型的过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“报警类型”选项相同。单击该按钮时，弹出一个报警类型对话框，对话框中的列表框中列出了所有报警类型供选择，选择完成后，单击对话框上的确定按钮关闭对话框。选择完后，只显示符合当前选择的报警类型的报警信息。
- ◇  更改事件类型：更改当前报警窗显示的事件类型的过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“事件类型”选项相同。单击该按钮时，弹出一个事件类型对话框，对话框中的列表框中列出了所有事件类型供选择，选择完成后，单击对话框上的确定按钮关闭对话框。选择完后，只显示符合当前选择的事件类型的事件

信息。

- ◇  更改优先级：更改当前报警窗显示的优先级过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“优先级”选项相同。单击该按钮时，弹出一个优先级编辑对话框，编辑优先级后，单击对话框上的确定按钮关闭对话框。选择完后，只显示符合当前选择的优先级的报警和事件信息。
- ◇  更改报警组：更改当前报警窗显示的报警组过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“报警组”选项相同。单击该按钮时，弹出一个报警组选择对话框，选择完报警组后，单击对话框上的确定按钮关闭对话框。选择完后，只显示符合当前选择的报警组及其子报警组的报警和事件信息。
- ◇  更改报警信息源：更改当前报警窗显示的报警信息源过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“报警信息源”选项相同。单击该按钮时，弹出一个报警信息源选择对话框，对话框中的列表框中列出了可供选择的报警信息源，选择完后，单击对话框上的确定按钮关闭对话框。则报警窗只显示符合当前选择的报警信息源的报警和事件信息。
- ◇  更改当前报警窗显示的报警服务器过滤条件。同 9.5.1.3 节报警窗配置——条件属性中“报警服务器”选项相同。单击列表框右侧的下拉箭头，从中选择报警服务器，选择完后。则报警窗只显示符合当前选择的报警服务器的报警和事件信息。

2、状态栏

共分为三栏：第一栏显示当前报警窗中显示的报警条数；第二栏显示新报警出现的位置；第三栏显示报警窗的滚动状态。

运行系统中的报警窗可以按需要不配置工具条和状态栏。

9.5.1.5 报警窗单击事件转发控件使用说明

◇ 报警窗单击事件转发控件

当用户在报警窗口使用鼠标单击某条报警（报警窗单击事件发生）时，可以通过“报警窗单击事件转发控件”KvAlmWinEv Control 来获得报警窗内某条报警的报警时间，报警类型，报警值等信息。

◇ 创建报警窗单击事件转发控件

在组态王开发系统中新建画面，在工具箱中单击“插入通用控件”或选择菜单“编辑”下的“插入通用控件”命令，弹出“插入控件”对话框，在列表中选择“KvAlmWinEv Control”，放置该控件。该控件在画面上显示为灰色方块。

◇ 报警窗单击事件转发控件的使用

KvAlmWinEv Control 控件具有如下的控件属性：

strVarName：报警变量名

strType：报警类型

strAlmDate：报警日期

strAlmTime：报警时间

strComment：报警变量描述

strOperator: 操作员

strGroup : 报警组

strMachine : 机器名

strEventType: 事件类型

strEveDate : 事件日期

strEveTime : 事件时间

strExtField1 : 扩展域 1

strExtField2 : 扩展域 2

strIOSerName : IOServer 名

以上属性均为字符串类型。

fAlmValue : 报警值

该属性为 FLOAT 型。

KvAlmWinEv Control 控件提供 AlmSelChg 事件，用户需要该事件的关联。函数命令语言中调用控件属性来获得报警窗中的报警信息。鼠标双击画面上的 KvAlmWinEv Control 控件，在事件属性页中关联函数，如图 9.39 所示。



图9.39 控件事件属性页

在控件事件函数命令语言中调用控件属性。

例如：组态王数据词典中定义 IO 整型变量 alarmR1，变量和变量的报警属性的定义如图 9.40，9.41 所示：



图9.40 报警变量定义

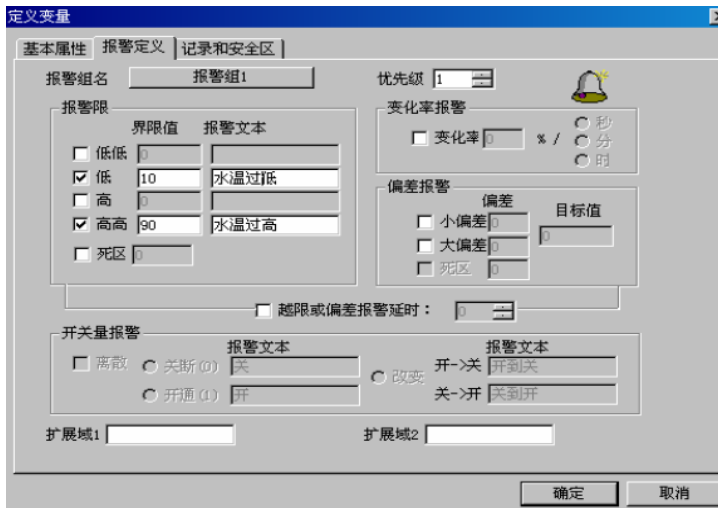


图9.41 变量的报警属性页

组态王数据词典中定义内存字符串变量：VarName, AlmDate, AlmTime, Comment, Group, Machine, Operator, Type, 分别表示变量名, 报警日期, 报警时间, 变量描述, 变量组, 机器名, 操作员, 报警类型。在画面上放置文本, 关联这些变量。画面上放置历史报警窗。

在画面上放置名为AlmWinEv的控件, 事件AlmSelChg关联的函数名为EvAlmSelchg。在控件事件命令语言中输入以下内容:

```
VarName=AlmWinEv.strVarName;  
AlmDate=AlmWinEv.strAlmDate;  
AlmTime=AlmWinEv.strAlmTime;  
Comment=AlmWinEv.strComment;  
Group=AlmWinEv.strGroup;  
Machine=AlmWinEv.strMachine;  
Operator=AlmWinEv.strOperator;  
Type=AlmWinEv.strType;
```

运行组态王, 鼠标单击报警窗中的某条报警信息, 画面显示该条报警的报警信息。如图9.42单击报警窗中的“水温过低”报警信息。

报警日期	报警时间	变量名	报警类型	事件类型	报警值/旧值	报警组名	操作员	变量描述	机器名
06/09/19	08:56:04...	alarmE1	水温过低	报警	8.0	报警组1	---	水温监测	本站点
06/09/19	08:56:04...	alarmE1	水温过低	恢复	8.0	报警组1	---	水温监测	本站点
06/09/19	08:57:42...	alarmE1	水温过高	报警	95.0	报警组1	---	水温监测	本站点
06/09/19	08:57:42...	alarmE1	水温过高	确认	95.0	报警组1	系统管理员	水温监测	本站点

报警的数目：4 新报警出现的位置：后面 滚动

报警窗单击事件转发控件


变量名	alarmE1	报警类型	水温过低
报警日期	2006-09-19	操作员	---
报警时间	08:56:04	报警组	报警组1
变量描述	水温监测	机器名	本站点

图9.42 单击报警窗中的报警信息

如图9.43，单击报警窗中的“确认”报警信息。



图9.43 单击报警窗中的报警确认信息

 注意:

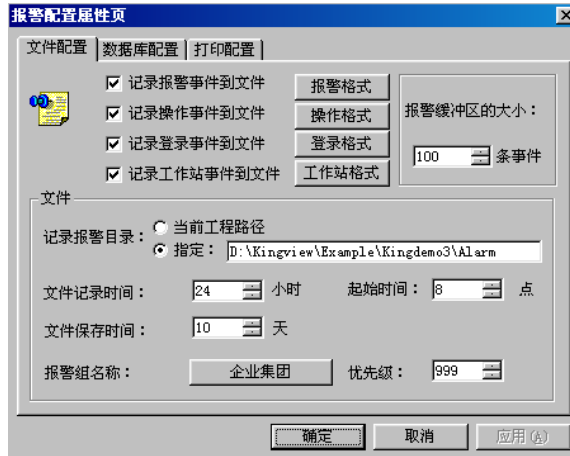
- 1、只能在控件的事件命令语言中调用控件属性，才能得到报警窗中的报警信息。
- 2、目前组态王只支持报警窗中的报警事件的单击转发，不支持操作事件，用户登录事件，应用程序事件，工作站事件的单击转发。

9.5.2 报警记录输出一：文件输出

系统的报警信息可以记录到文本文件中，用户可以通过这些文本文件来查看报警记录。记录的文本文件的记录时间段、记录内容、保存期限等都可定义。文件的后缀名称为“.a12”。文件的命名方法请参见 9.5.2.1 节中的内容。

9.5.2.1 报警配置——文件输出配置

打开组态王工程管理器，在工具条中选择“报警配置”，或双击列表项“系统配置/报警配置”，弹出报警配置属性页对话框，如图 9.44 所示。



9.44 报警配置属性——文件配置

文件配置对话框中各部分的含义是：

◇ 记录内容选择：

其中包括：“记录报警事件到文件”选项、“记录操作事件到文件”选项、“记录登录事件到文件”选项、“记录工作站事件到文件”选项。只有当选择某一项时，该项才有可能被记录到文件，否则不记录与该项有关的信息。如不选择“记录工作站事件到文件”选项，则系统运行时，就不会记录任何工作站事件的信息。记录内容中除了这些的选项外，在各个选项中，还可以定义具体记录内容、格式等，即各选项后的格式按钮，如“报警格式”，关于各选项格式的定义在 9.5.2.2 节中详细

讲述。

◇ 记录报警目录：

定义报警文件记录的路径。有两个选项：

- **当前工程路径：**记录到当前组态王工程所在的目录下
- **指定：**当选择该项时，其后面的编辑框变为有效，在编辑框中直接输入报警文件将要存储的路径。

◇ 文件记录时间：

报警记录的文件一般有很多个，该项指定没有记录文件的记录时间长度，单位为小时，指定数值范围为 1~24。如果超过指定的记录时间，系统将生成新的记录文件。

如定义文件记录时间为 8 小时，则系统按照定义的起始时间，每 8 小时生成一个新的报警记录文件。

◇ 起始时间：

指报警记录文件命名时的时间（小时数），表明某个报警记录文件开始记录的时间。

其值为 0~23 之间的一个整数。组态王根据“起始时间”和“记录时数”来生成一系列的报警记录文件，文件命名规则为：YYMMDDHH.AL2，其中 YY、MM、DD、HH 代表年、

月、日、时。如定义“文件记录时间”为 8 小时，“起始时间”为 8 点，当前日期

为 2002 年 11 月 18 日，则当当前时间在 8~16 点间时，系统运行生成的文件名为：

02111808.al2，而这段时间内的报警和事件信息记录到该文件中。同样，当当前时

间在 16 点到 19 日 0 点间时，系统运行生成的文件名为：02111816.al2；当当前时

间在 19 日 0 点到 8 点间时，系统运行生成的文件名为：02111900.al2。

◇ 文件保存时间:

规定记录文件在硬盘上的保存天数（当日之前）。超过天数的记录文件将被自动删除。保存天数为 1~999 之间。

◇ 报警组名称:

选择要记录的报警和事件的报警组名称条件，只有符合定义的报警组及其子报警组的报警和事件才会被记录到文件。

◇ 优先级:

规定要记录的报警和事件的优先级条件。只有高于规定的优先级的报警和事件才会被记录到文件中。

文件配置完成后，单击确定关闭对话框。文件记录中每条报警和事件记录占用一行，每条记录中每项记录都用 “[]” 隔离开来。如本章前几节中产生的记录，如果按照默认格式配置，则记录的部分内容为：

◇ 工作站启动事件:

[机器名:本站点] [事件类型:工作站启] [工作站时间:09 时 31 分 14 秒]

◇ 应用程序事件:

[操作时间:09 时 31 分 33 秒] [事件类型:应用程序] [成功标志:成功] [变量名:DDE 事件] [变量注释:] [操作员:] [新值:1.000000] [域名:值] [旧值:0.000000] [机器名:本站点] [I/O 服务器名:本站点]

◇ 报警:

[报警时间:15 时 38 分 25 秒 380 毫秒] [确认或恢复时间:] [变量名:

压力测量] [报警值:5.5508] [限值:3.0000] [报警类型:大偏差] [事件类型:报警] [恢复值:] [操作员名:] [质量戳:192] [优先级:200] [机器名:本站点] [IO 服务器名:本站点]

◇ 报警恢复:

[报警时间:15 时 38 分 25 秒 660 毫秒] [确认或恢复时间:15 时 38 分 25 秒 770 毫秒] [变量名:压力测量] [报警值:-0.238] [限值:2.0000] [报警类型:小偏差] [事件类型:报警恢复] [恢复值:0.3252] [操作员名:] [质量戳:192] [优先级:200] [机器名:本站点] [IO 服务器名:本站点]

◇ 登录事件:

[登录时间:16 时 55 分 23 秒] [事件类型:登录退出] [成功标志:成功] [操作员:无] [机器名:本站点]

[登录时间:16 时 55 分 23 秒] [事件类型:登录进入] [成功标志:成功] [操作员:系统管理] [机器名:本站点]

◇ 工作站退出:

[机器名:本站点] [事件类型:工作站退] [工作站时间:15 时 47 分 59 秒]

9.5.2.2 通用报警和事件记录格式配置

在规定报警和事件信息输出时,同时可以规定输入的内容和每项内容的长度。这就是格式配置,格式配置在文件输出、数据库输入和打印输出中都相同。这里统一描述。

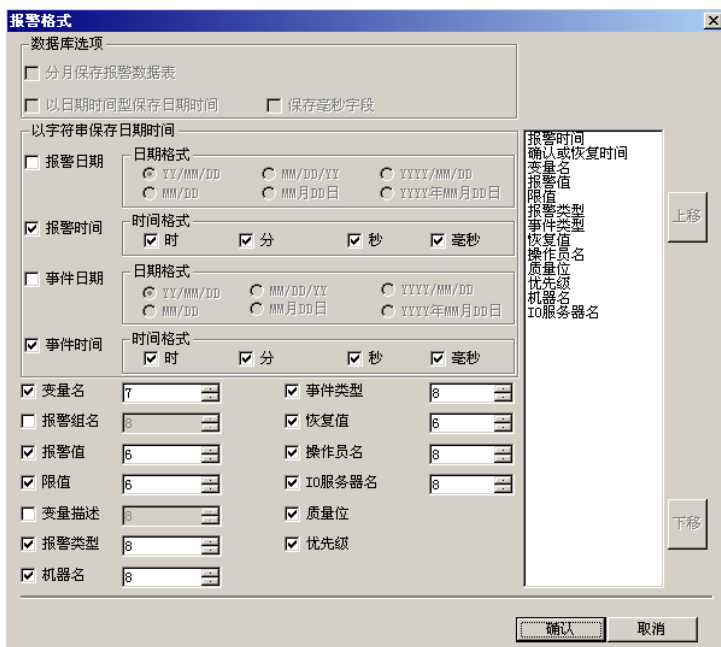
1、报警格式:

如图 9.45 所示,每个选项都有格式或字符长度设置,当选中某一项时,在对话框右

侧的列表框中会显示该项的名称，在进行文件记录和实时打印时，将按照列表框中的顺序和列表项；在数据库记录时，只记录列表框中有的项，没有的项将不被记录。选中列表框中的某一项，单击对话框右侧的“上移”或“下移”按钮，可以移动列表项的位置。

注意：

报警格式中的事件是指报警确认、报警恢复事件。记录时，只按照规定的字符长度记录，如果实际字符长度小于规定长度，则记录时用空格补齐；如果实际字符长度大于规定字符长度，记录时将自动截取规定长度的字符。本项适合于所有记录格式。



The dialog box titled "报警格式" (Alarm Format) is used for configuring database options and alarm/event recording formats. It is divided into several sections:

- 数据库选项 (Database Options):**
 - 分月保存报警数据表 (Save alarm data table by month)
 - 以日期时间型保存日期时间 (Save date and time as date-time type)
 - 保存毫秒字段 (Save millisecond fields)
- 以字符串保存日期时间 (Save date and time as string):**
 - 报警日期 (Alarm date): Date format options: YY/MM/DD, MM/DD/YY, YYYY/MM/DD, MM/DD, MM月DD日, YYYY年MM月DD日.
 - 报警时间 (Alarm time): Time format options: 时, 分, 秒, 毫秒.
 - 事件日期 (Event date): Date format options: YY/MM/DD, MM/DD/YY, YYYY/MM/DD, MM/DD, MM月DD日, YYYY年MM月DD日.
 - 事件时间 (Event time): Time format options: 时, 分, 秒, 毫秒.
- 报警和事件记录格式 (Alarm and Event Recording Format):**
 - 变量名 (Variable name): 7 characters
 - 报警组名 (Alarm group name): 3 characters
 - 报警值 (Alarm value): 6 characters
 - 限值 (Limit value): 6 characters
 - 变量描述 (Variable description): 3 characters
 - 报警类型 (Alarm type): 8 characters
 - 机器名 (Machine name): 8 characters
 - 事件类型 (Event type): 8 characters
 - 恢复值 (Recovery value): 6 characters
 - 操作员名 (Operator name): 8 characters
 - IO服务器名 (IO server name): 8 characters
 - 质量位 (Quality bit)
 - 优先级 (Priority)
- 列表框 (List Box):** A list of items to be recorded, including: 报警时间, 确认或恢复时间, 变量名, 报警值, 限值, 报警类型, 事件类型, 恢复值, 操作员名, 质量位, 优先级, 机器名, IO服务器名.
- 操作按钮 (Buttons):** "上移" (Up), "下移" (Down), "确认" (OK), and "取消" (Cancel).

图9.45 报警格式配置对话框

- ◇ 数据库选项：将在章节 9.5.3.3 报警输出数据库配置中详细介绍。
- ◇ 报警日期：选中该项时，后面的格式选项有效，任选一种记录的报警日期的格式。
- ◇ 报警时间：选中该项时，后面的格式选项有效，选择记录的报警时间格式。
- ◇ 事件日期：选中该项时，后面的格式选项有效，任选一种记录事件日期的格式。
- ◇ 事件时间：选中该项时，后面的格式选项有效，选择记录的事件时间显示格式。
- ◇ 变量名：选中该项时，表示输出中记录变量名称，编辑框中定义记录该字符的长度，为 1~127 之间的整数。
- ◇ 报警组名：选中该项时，表示输出中记录报警组名称，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 报警值：选中该项时，表示输出中记录报警值，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 限值：选中该项时，表示输出中记录报警的限值，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 变量描述：选中该项时，表示输出中记录变量描述内容，编辑框中定义记录该字符的长度，为 1~39 之间的整数。
- ◇ 报警类型：选中该项时，表示输出中记录报警类型，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 机器名：选中该项时，表示输出中记录机器名信息，编辑框中定义记录该字符的长度，为 1~31 之间的整数。

- ◇ 事件类型：选中该项时，表示输出中记录事件类型，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 恢复值：选中该项时，表示输出中记录报警的恢复值，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 操作员名：选中该项时，表示输出中记录操作员名称信息，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ I/O 服务器名：选中该项时，表示输出中记录 IO 服务器名称，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 质量位(质量戳的整型值)：选中时表示输出中记录 IO 数据的质量戳数值。
- ◇ 优先级：选中时表示输出中记录报警或事件的优先级。

2、操作格式：

如图 9.46 所示，每个选项都有格式或字符长度设置，当选中某一项时，在对话框右侧的列表框中会显示该项的名称，在进行文件记录和实时打印时，将按照列表框中的顺序和列表项；在数据库记录时，只记录列表框中有的项，没有的项将不被记录。选中列表框中的某一项，单击对话框右侧的“上移”或“下移”按钮，可以移动列表项的位置。



9.46 操作格式配置对话框

- ◇ 事件日期：选中该项时，后面的格式选项有效，任选一种记录操作事件日期的格式。
- ◇ 事件时间：选中该项时，后面的格式选项有效，选择记录的操作事件时间格式。
- ◇ 事件类型：选中该项时，表示输出中记录事件的类型，编辑框选择记录该字符的长度，为 1~31 之间的整数。
- ◇ 成功标志：选中该项时，表示输出中记录操作成功标志，编辑框选择记录该字符的长度，为 1~31 之间的整数。
- ◇ 变量名：选中该项时，表示输出中记录操作的变量的名称，编辑框选择记录该字符的长度，为 1~127 之间的整数。
- ◇ 变量描述：选中该项时，表示输出中记录操作的变量的变量描述，编辑框选择记录该字符的长度，为 1~39 之间的整数。
- ◇ 机器名：选中该项时，表示输出中记录机器名信息，编辑框中定义记录该字符的长度，为 1~31 之间的整数。

- ◇ 操作员名：选中该项时，表示输出中记录操作员名称信息，编辑框中定义记录该字符的长度，为 1~31 之间的整数。
- ◇ 新值：选中该项时，表示输出中记录操作变量前的变量的值，编辑框选择记录该字符的长度，为 1~31 之间的整数。
- ◇ 域：选中该项时，表示输出中记录操作的变量的域，编辑框选择记录该字符的长度，为 1~31 之间的整数。
- ◇ 旧值：选中该项时，表示输出中记录操作变量后的变量的值，编辑框选择记录该字符的长度，为 1~31 之间的整数。
- ◇ I/O 服务器名：选中该项时，表示输出中记录 IO 服务器名称，编辑框中定义记录该字符的长度，为 1~31 之间的整数。

3、登录格式：

如图 9.47 所示，每个选项都有格式或字符长度设置，当选中某一项时，在对话框右侧的列表框中显示该项的名称，在进行文件记录和实时打印时，将按照列表框中的顺序和列表项；在数据库记录时，只记录列表框中有的项，没有的项将不被记录。选中列表框中的某一项，单击对话框右侧的“上移”或“下移”按钮，可以移动列表项的位置。



9.47 登录格式配置对话框

- ◇ 事件日期：选中该项时，后面的格式选项有效，任选一种记录登录事件的日期格式。
- ◇ 事件时间：选中该项时，后面的格式选项有效，选择记录登录事件的时间格式。
- ◇ 事件类型：选中该项时，表示输出中记录事件的类型，编辑框选择记录该字符的长度，为1~31之间的整数。
- ◇ 成功标志：选中该项时，表示输出中记录登录的成功标志，编辑框选择记录该字符的长度，为1~31之间的整数。
- ◇ 操作员名：选中该项时，表示输出中记录操作员名称信息，编辑框中定义记录该字符的长度，为1~31之间的整数。
- ◇ 机器名：选中该项时，表示输出中记录机器名信息，编辑框中定义记录该字符的长度，为1~31之间的整数。

4、工作站格式：

如图 9.48 所示，每个选项都有格式或字符长度设置，当选中某一项时，在对话框右侧的列表框中显示该项的名称，在进行文件记录和实时打印时，将按照列表框中的顺

序和列表项；在数据库记录时，只记录列表框中有的项，没有的项将不被记录。选中列表框中的某一项，单击对话框右侧的“上移”或“下移”按钮，可以移动列表项的位置。



9.48 工作站格式配置对话框

- ◇ 事件日期：选中该项时，后面的格式选项有效，任选一种记录工作站事件的日期格式。
- ◇ 事件时间：选中该项时，后面的格式选项有效，选择记录工作站事件的时间格式。
- ◇ 事件类型：选中该项时，表示输出中记录事件的类型，编辑框选择记录该字符的长度，为1~31之间的整数。
- ◇ 机器名：选中该项时，表示输出中记录机器名信息，编辑框中定义记录该字符的长度，为1~31之间的整数。

9.5.3 报警记录输出二：数据库

组态王产生的报警和事件信息可以通过 ODBC 记录到开放式数据库中，如 Access、SQL Server 等。在使用该功能之前，应该做些准备工作：首先在数据库中建立相关的数据表

和数据字段（9.5.3.1节），然后在系统控制面板的 ODBC 数据源中配置一个数据源（用户 DSN 或系统 DSN），该数据源可以定义用户名和密码等权限。

9.5.3.1 定义报警记录数据库

报警输出数据库中的数据表与配置中选项相对应，有四种类型的数据表格，这四种表格的名称为：**Alarm（报警事件）、Operate（操作事件）、Enter（登录事件）、Station（工作站事件）**。可以按照需要建立相关的表格。各个表中的字段对应记录格式中的选项，如 Alarm（报警事件）表中的 AlarmDate 字段对报警记录格式中的“报警日期”选项。具体参见以下表。

- **Alarm（报警）**

当记录报警事件到数据库的报警格式（参见 9.5.3.3 节）选择“以字符串保存日期时间”时，Alarm 表的字段包括

字段名称	说明
AlarmDate	报警日期
AlarmTime	报警时间
VarName	变量名
GroupName	报警组名
AlarmValue	报警值
LimitValue	限值
AlarmType	报警类型
Pri	优先级
Quality	质量位

AcrDate	事件日期
AcrTime	事件时间
OperatorName	操作员名
VarComment	变量描述
ResumeValue	恢复值
EventType	事件类型
MachineName	工作站名称
IOServerName	报警服务器名称

当报警格式选择“以日期时间型保存日期时间”时，Alarm表的字段包括

字段名称	说明
AlarmDateTime	报警日期时间
AlarmMilliSecond	报警时间的毫秒
AlarmType	报警类型
AcrDateTime	确认或恢复时期时间
AcrMilliSecond	确认或恢复时期时间的毫秒
EventType	事件类型
VarName	变量名
AlarmValue	报警值
LimitValue	限值
ResumeValue	恢复值
OperatorName	操作员名
Quality	质量位
Pri	优先级

GroupName	报警组名
VarComment	变量描述
MachineName	工作站名称
IOServerName	报警服务器名称

其中 AlarmDateTime 和 AcrDateTime 字段类型必须为日期时间型。

- **Operate (操作)** 表有如下字段:

字段名称	说明
VarName	变量名
OldValue	旧值
NewValue	新值
OperatorName	操作员名
AlarmDate	事件日期
AlarmTime	事件时间
SuccessOa	成功标志
EventType	事件类型
FieldName	域名
VarComment	变量描述
MachineName	工作站名称
IOServerName	报警服务器名称


- **Enter (登录)** 表有如下字段:

字段名称	说明
------	----

OperatorName	操作员名
SuccessEnter	成功标志
EventType	事件类型
AlarmDate	事件日期
AlarmTime	事件时间
MachineName	工作站名称

- **Station**（工作站）表有如下字段：

字段名称	说明
EventType	事件类型
AlarmDate	事件日期
AlarmTime	事件时间
MachineName	工作站名称

 注意：

- 1、 定义数据表时字段名称应该正确，否则会造成记录不了的情况发生。
- 2、 对于和日期时间相关的字段，如果使用组态王的 SQL 功能进行数据库查询，这些字段必须定义成文本类型。

9.5.3.2 报警输出数据库配置

按 9.5.3.1 节中定义好报警记录数据库和定义完 ODBC 数据源后，就可以在组态王中定义数据库输出配置了。如图 9.49 所示为报警配置——数据库配置对话框。




9.49 数据库配置

对话框中各项含义为:

- **记录报警事件到数据库:** 记录报警数据库时, 是否包括报警事件。关于报警格式的定义, 请参见 9.5.3.3 节。
- **记录操作事件到数据库:** 记录报警数据库时, 是否包括操作事件。关于报警格式的定义, 请参见 9.5.2.2 节。
- **记录登录事件到数据库:** 记录报警数据库时, 是否包括登录事件。关于报警格式的定义, 请参见 9.5.2.2 节。
- **记录工作站事件到数据库:** 记录报警数据库时, 是否包括工作站事件。关于报警格

式的定义，请参见 9.5.2.2 节。

- **使用默认数据源：**在“报警配置属性页”，使“文件配置”中默认事件保存不选中，图 9.49 中“使用默认数据源”是否有效，是由上面的四个记录事件确定的。
- **用户名、密码：**输入在定义 ODBC 数据源时定义的用户名和密码，如果没有，置为空即可。
- **数据源：**输入定义的与报警数据库连接的 ODBC 数据源名称。也可通过单击“数据源”标签后的  按钮，在弹出的“ODBC”数据源对话框中选择，该对话框包含“系统 DSN”和“用户 DSN”两项，分别列出当前系统中已有的数据源名称。
- **组名：**选择记录到数据库中的报警和事件的报警组条件，只有符合当前选中的报警组及其子报警组的报警和事件信息才会被记录到数据库中，报警组组名只能选择一个。
- **优先级：**选择记录到数据库中的报警和事件的优先级条件，只有比当前优先级高的报警和事件信息才会被记录到数据库中。优先级范围为 1~999 之间的整数。

当 4 个记录事件之一有效时，“使用默认数据源”选项有效，即可以对“使用默认数据源”进行操作。如果不选“使用默认数据源”，则用“用户名”、“密码”和“数据源”三个选项有效，可以进行相应的操作。

9.5.3.3 记录报警事件到数据库的格式配置

报警事件可以记录到数据库中的一个表中，也可以分月保存。报警的日期和时间可以记录成日期时间型，也可以记录成字符串型。

记录报警事件到数据库的报警格式配置界面如图 9.50 所示：



图 9.50 数据库报警格式配置对话框

- **分月保存报警数据表：**把报警数据按照月份的不同存储在数据库的不同表中。无论是以日期时间类型保存日期时间，还是以字符串保存时期时间，对报警数据都可以进行分月保存。如图 9.50，选中“分月保存报警数据表”，组态王系统运行之后，将以数据库中“Alarm”表为标准，在警数据库中每月自动创建一张新表来保存这个月的报警数据。以 Access 数据库为例，创建的新表如下图所示，新表名称以“Alarm”+“年月”表示。

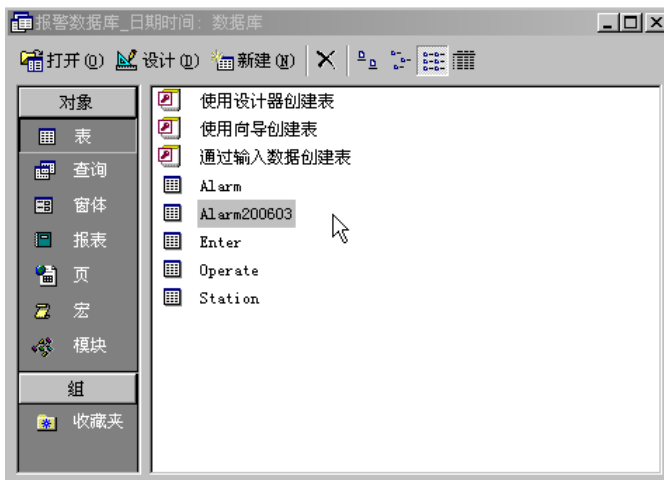


图 9.51 创建了新表的 Access 数据库

报警数据保存在不同月份的报警表中，“Alarm”表只是作为创建分月表的模板，它本身并不保存报警数据。如果不选择“分月保存报警数据表”，所有的报警数据都将保存在同一个表“Alarm”中。

- **以日期时间型保存日期时间：**选择该项，如图 9.52 所示，组态王报警日期时间和报警确认、恢复的日期时间在数据库中以日期时间型保存。

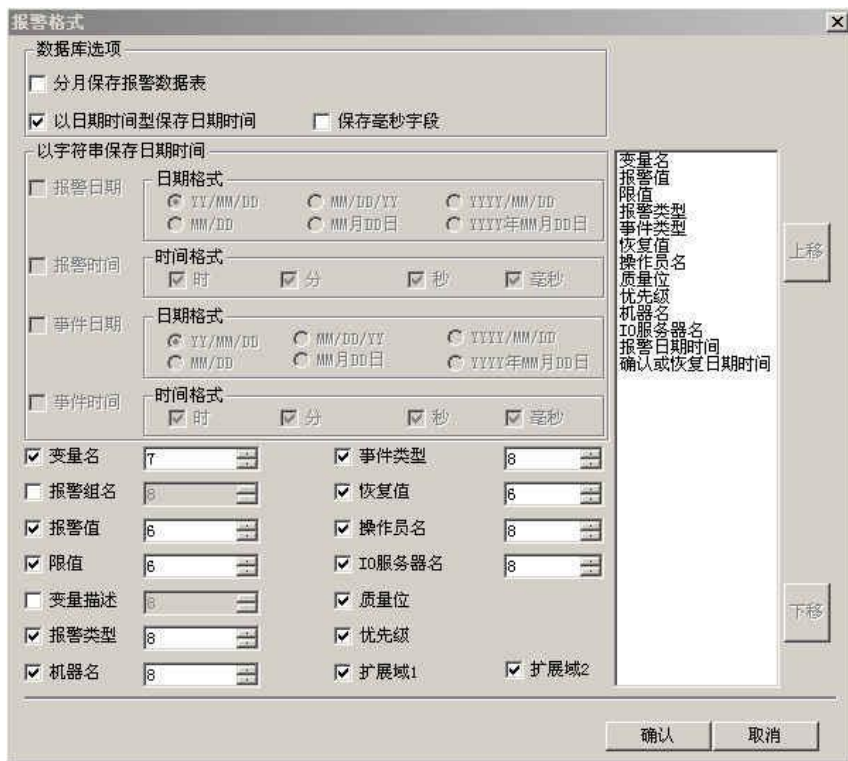


图 9.52 报警配置以日期时间型保存日期时间对话框

选择该选项时，在数据库中建立的“Alarm”表包括的字段内容与以字符串保存日期时间所用的“Alarm”表的字段内容略有不同，参见 9.5.3.1 节。

注意，选择该选项时，组态王运行系统会自动按系统时间格式向数据库插入报警记录，如果报警数据库与组态王报警服务器不位于同一台计算机上，要保证两台计算机系统日期时间格式的一致。对于 Windows2000 的设置，在开始>设置>控制面板>区域选项>时间中设置时间格式。建议用户使用 HH:mm:ss（时：分：秒）的时间格

式。如果您使用的是 SQLserver 数据库，请不要使用 tt h:mm:ss 或 tt hh:mm:ss 类型的时间格式（tt 指上午或下午）。

- **保存毫秒字段：**选择该项，则在数据库中记录组态王报警日期时间的毫秒数和报警确认、恢复日期时间的毫秒数。只有在选择了“以日期时间型保存日期时间”后，“保存毫秒字段”选择才有效。选择后，在报警格式的选择列表中出现“报警毫秒”和“确认或恢复毫秒”字段，如下图所示。这两个字段用来记录对应报警时间的毫秒部分。



- **以字符串保存日期时间：**不选择“以日期时间型保存日期时间”，即为以字符串保存日期时间。
- **扩展域 1、扩展域 2：**选择该项，组态王运行系统则在数据表“Alarm”中保存相应报警变量的扩展域信息。

以字符串保存日期时间时，报警日期、报警时间、事件日期、事件时间的格式设置和字符长度设置请参见 9.5.2.2 节。

9.5.4 报警记录输出三：实时打印输出

组态王产生的报警和事件信息可以通过计算机并口实时打印出来。首先应该对实时打印进行配置。如图 9.53 所示，为报警打印配置对话框。



9.53 报警打印配置对话框

- **实时打印报警事件：**打印报警时，是否包括报警事件。关于报警格式的定义，请参见 9.5.2.2 节。

- **实时打印操作事件：**打印报警时，是否包括操作事件。关于报警格式的定义，请参见 9.5.2.2 节。
- **实时打印登录事件：**打印报警时，是否包括登录事件。关于报警格式的定义，请参见 9.5.2.2 节。
- **实时打印工作站事件：**打印报警时，是否包括工作站事件。关于报警格式的定义，请参见 9.5.2.2 节。
- **打印机名：**计算机并口连接的打印机，打印机名的设定只能选择 LPT1、LPT2、LPT3 中的一个。
- **组名：**选择打印报警时的报警组条件，只有当前选中的报警组及其子报警组的报警和事件信息才能被输出打印，报警组组名只能选择一个。
- **优先级：**选择打印报警时的优先级条件，只有高于当前优先级的报警和事件信息才能被输出打印。

按照用户在“报警配置”中定义的报警事件的打印格式及内容，系统将报警信息送到指定的打印端口缓冲区，将其实时打印出来。在打印时，某一条记录中间的各个字段以“/”分开，每个字段包含在“<>”内，并且字段标题与字段内容之间用冒号分割。打印时，两条报警信息之间以-----分隔。

 例如：工作站事件打印。

<工作站日期:2002 年 11 月 18 日>/<工作站时间:14 时 24 分 7 秒>/<事件类型:工作站启动 >/<机器名:本站点 >

<工作站日期:2002 年 11 月 18 日>/<工作站时间:14 时 24 分 14 秒>/<事件类型:工作站退出 >/<机器名:本站点 >



注意:

因为组态王的实时报警信息是直接输出到打印端口的（如 LPT1），建议用户在使用实时报警打印时，最好使用带硬字库的针式打印机（即打印机本身带字库，市场上其它类型的打印机，如激光式、喷墨式、部分针式打印机等，其本身不带字库，均使用系统的字库），如 EPSON 的 1600II、1600KIII 等，否则会出现打印出来的报警信息中的汉字为乱码的情况。打开工程浏览器，在其目录显示区中单击系统配置 → 双击打印配置，在弹出的打印配置窗口中进行打印机的设置。这种设置方法能支持报警网络打印。

9.6 报警相关的函数和变量的报警域

组态王中提供了函数、变量和变量的报警域。

9.6.1 \$新报警变量的使用

\$新报警变量是组态王的一个系统变量，主要表示当前系统中是否有新的报警产生。当系统中有无任何类型的新报警产生时，该变量被自动置为 1。但需要注意的是，该变量不能被自动清 0，需要用户人为的将其清 0。

9.6.2 报警相关的函数

组态王提供了一些报警操作函数：

- **报警确认函数 Ack(Tagname or GroupName)：**

对变量进行报警确认，或对报警组进行报警确认。如果函数参数为变量名称，则只对该变量进行报警确认；如果函数参数为报警组名称，则确认所有属于该报警组及其子报警组的变量。该函数的参数只能是变量名或报警组名，不可以为字符串变量。



例如：

```
Ack(\\本站点\液位)
```

```
Ack(报警组 1)
```

- **报警确认函数 AckByTagName("tag_name")：**

对变量进行报警确认。函数参数可以是一个字符串变量，也可以是一个表示变量名的字符串。



例如：

```
AckByTagName ("\\本站点\液位")
```

```
AckByTagName (Varname)，其中 Varname 为字符串变量。
```

- **报警确认函数 AckByGroupName("station_name", "group_name")：**

对报警组进行确认。station_name 为产生报警的 I0 服务器名，group_name 为报警组名。函数参数可以是字符串变量，也可以是表示 I0 服务器名或报警组名的字符串。



例如：

```
AckByGroupName ("PC1", "报警组 1")
```


AckByGroupName (PCName, GroupName), PCName, GroupName 为字符串变量。

除了通过报警窗上的按钮对报警进行确认之外，也可以在组态王的命令行言中使用上面的函数，对报警或报警组进行确认。

- **设定报警消失条件的函数(默认是恢复即消失) void SetAlarmWinDis(string name, long nSet):**

对已经发生报警还未恢复的变量，分别显示确认和未确认的变量。string name: 报警窗的名称。nSet=1: 恢复+确认，nSet=2: 确认，其它为恢复。此函数仅对于实时报警窗有效。

当设置为确认报警消失时，即 nSet=2 时，可以实现显示未确认的报警；若想显示已确认的报警，则可以在“报警窗口属性配置页”中“事件类型”只勾选“确认”，如下图所示。

 **注意:**

实时报警窗口的事件类型选项只在运行是可用，且该函数仅在第二十四章全新的 Web 发布中有效。

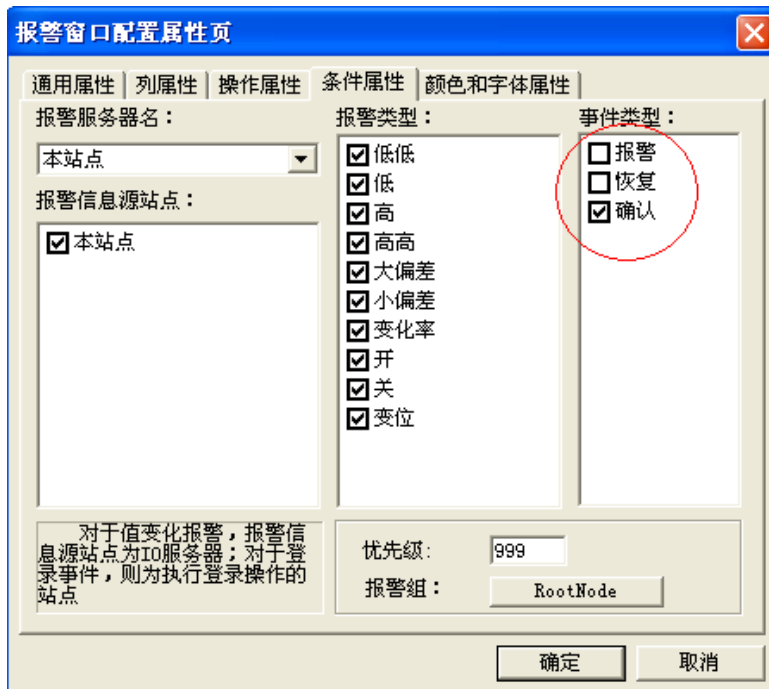


图 9.54 运行时的报警窗口配置属性页

- 获取报警组名称 `GetGroupName(StationName, GroupID)`

在组态王中，变量的报警组域返回的是报警组的 ID 号（数字），通过该函数可以获得报警组的名称。除按钮命令语言外，在组态王命令语言中修改变量的值是不会生成操作事件的。使用下面两个函数可以解决该问题。

- 生成实型变量的操作事件函数 `SetRealDBForFloat("VarName", Value)`

VarName 为变量名，Value 为变量值。



例如：

在画面 “显示时” 命令语言中输入：

```
SetRealDBForFloat (“\\本站点\液位”, 1.5)
```

设置实型变量 “\\本站点\液位” 的值为 1.5。

画面显示时，生成实型变量 “\\本站点\液位” 的操作事件。

- **生成整型变量的操作事件函数 SetRealDBForInt (“VarName”, Value)**

VarName 为变量名，Value 为变量值。



例如：

在画面 “显示时” 命令语言中输入：

```
SetRealDBForInt (“\\本站点\行”, 10)
```

设置整型变量 “\\本站点\行” 的值为 10。

画面显示时，生成整型变量 “\\本站点\行” 的操作事件。

- **报警窗口信息显示条件函数 SetAlarmWinDis(long nSet) ;**

参数说明：nSet=1：恢复并且确认后，报警信息才消失。

非 1：恢复

默认时为报警恢复即报警消失；

报警恢复+报警确认：既显示报警事件又显示报警恢复事件，当报警被确认以后报警信息消失。

- **报警组当前报警变量数量函数** `long GetAlarmNumInGroup("MachineName", "GroupName");`

功能：对某一个报警组当前的报警变量数量进行统计。

参数说明：

MachineName：报警组所在的站点名称(对于单机网络，使用时用空字符串代替)

GroupName：要获取报警变量个数的报警组名

返回值：整型，返回值表示报警变量数量



注意：

对于网络节点，需要把“本机为报警服务器”钩上，才能正确获取到远程节点的报警变量个数。

- **全局报警使能函数** `long EnableAllAlarm(is_enable);`

参数说明：

is_enable 整数变量或数值

0—禁止所有报警

非 0—使能所有报警

返回值：整数型，反映当前全局报警使能的状态。

若为非 0：使能；


为 0：禁止

- **报警实时打印函数** `BOOL SetPrintAlarm(BOOL bitset)`

参数说明:

bitset: bool 类型, 为 1 时则停止报警打印, 为 0 开始报警打印

返回值: bool 类型, 能反映设置后的打印状态, 为 1 时则停止报警打印, 为 0 开始报警打印

 注意:

此函数只有当配置了报警配置中的报警打印后才能起作用。

函数的具体使用方法请参见《组态王函数使用手册》。

9.6.3 变量的报警域

变量的报警域类型有很多种, 离散变量有以下几种报警域:

- ◇ **.Ack:** 表示变量是否被确认。初始或变量产生新报警而未被确认时, 该域的值为 0, 当变量的报警被确认后, 域的值为 1。
- ◇ **.Alarm:** 表示变量当前是否处于报警状态。变量处于正常状态时, 域的值为 0, 变量处于报警状态时, 无论是否被确认, 域的值为 1。
- ◇ **.AlarmEnable:** 表示变量的报警使能状态, 可读写。当 AlarmEnable 置 0 时, 变量即使满足报警条件也不会产生报警, 只有将 AlarmEnable 置 1, 变量才会产生报警。AlarmEnable 默认值为 1。
- ◇ **.Group:** 变量的报警组 ID 号。
- ◇ **.Priority:** 变量的优先级。

模拟变量除了具有以上四种报警域外,按照变量的报警类型的不同,有不同的报警域。

1、越限报警:

- ◇ 报警界限值域: 高高限值 (.HiHiLimit)、高限值 (.HiLimit)、低限值 (.LoLimit)、低低限值 (.LoLoLimit)。这些界限值域是可读写的,可以在线修改。
- ◇ 报警状态值: 对应越限报警类型,都有相应的报警状态域,这些域的值类型是离散型的。高高限状态 (.HiHiStatus)、高限状态 (.HiStatus)、低限状态 (.LoStatus)、低低限状态 (.LoLoStatus)。如变量处于高限报警时,变量的.HiStatus 域的值被自动置为 1,当高限报警恢复后,该域的值被自动置为 0。

2、偏差报警:

- ◇ 报警界限值域: 大偏差限值 (.MajorDevPct)、小偏差限值 (.MinorDevPct)、偏差目标值 (.DevTarget)。这些界限值域是可读写的,可以在线修改。
- ◇ 报警状态值: 对应偏差报警类型,都有相应的报警状态域,这些域的值类型是离散型的。大偏差状态 (.MajorDevStatus)、小偏差状态 (.MinorDevStatus)。如变量处于大偏差报警时,变量的.MajorDevStatus 域的值被自动置为 1,当大偏差报警恢复后,该域的值被自动置为 0。

3、越限或偏差报警延时:

- ◇ 报警延时值域: 越限和偏差报警信息的延时时间,该值域是可读写的,可以在线修改。该域的值类型是模拟型的。

4、变化率报警：

- ◇ 报警界限值域：变化率限值（. ROCpct）。该域是可读写的，可以在线修改。
但变化率报警的时间单位（秒、分、时）不能在线修改。
- ◇ 报警状态值：变化率报警的报警状态域为（. ROCStatus），该域的值的类型是离散型的。

9.7 报警预置自定义函数

在组态王自定义函数中，有三个系统预置的报警自定义函数。分别为实型变量报警事件（\$System_RealAlarm）、整型变量报警事件（\$System_LongAlarm）和离散型变量报警事件（\$System_DiscAlarm）预置自定义函数。


注意：

- 1、这三个预置自定义函数名区分大小写。在函数中使用函数形式参数时，区分大小写。
 - 2、这里的报警事件是指报警的产生事件、报警的确认事件和报警的恢复事件。其它如登录事件、操作事件等的产生都不会调用这些函数。
-

自定义函数的调用执行有两种方式：

- 系统产生报警事件后会自动调用相应数据类型的函数,如整型变量产生报警时,系统自动调用整型预置自定义函数；
- 如果在配置报警窗的“操作属性”时，选择了“允许双击报警条”，则系统运行时

双击报警事件报警条，也会自动调用相应数据类型的函数。

 注意：

预置自定义函数不可以在其它命令语言中调用。函数名称及参数是不能修改的。
在每个函数中不能对产生报警事件的变量或变量的域赋值。

三种预置自定义函数如下：


1. 实型：void \$System_RealAlarm(RealTag rTag, long time, long nEvent, long nAction)
{
}
2. 整型：void \$System_LongAlarm(LongTag lTag, long time, long nEvent, long nAction)
{
}
3. 离散型：void \$System_DiscAlarm(DiscTag dTag, long time, long nEvent, long nAction)
{
}

各个预置自定义函数中的参数的含义为（以实型为例，其它都相同）：

```
Void $System_RealAlarm(RealTag rTag, long nTime, long nEvent, long nAction)
```

- rTag: 表示变量, 即普通变量, 和组态王系统变量一样具有值和变量所有的域, 这些值都是只读的。
- nTime: 表示自从格林威治时间 1970 年 1 月 1 日 0 起到报警事件产生时的秒数, 表征报警事件产生的时间;
- nEvent: 表示当前产生或双击报警窗时的报警类型。报警类型的返回值如下:
 - 0—— 报警
 - 1—— 恢复
 - 2—— 确认
- nAction: 表示当 nAction==1 时表示是双击报警条, 当 nAction==0 时表示产生报警事件。

预置自定义函数函数体初始内容为空, 需要用户在里面添加命令语言。利用报警预置自定义函数, 可以实现用户自己想在报警产生后做的一些处理。

 例如:

某厂的监控系统共定义了“炼钢车间”和“反应车间”两个报警组, 反应车间下有“液位”报警组, 每个报警组下都有重要的报警信息。现在要求将“反应车间”报警组报警产生时的变量名、变量的当前值、报警组名、报警时间、变量的描述等信息记录到一个组态王的报表中, 并实时打印出来, 上报到相关人员。变量类型为整型。

在这个例子中, 可以通过报警的预置自定义函数完全轻松实现。另外需要制作一个与之相配套的组态王报表。

1. 在组态王中新建一个画面, 在画面中创建一个 10 行 2 列的报表窗口, 名称为“报警

报表”。定义报表单元格的属性，并在报表单元格中输入一些项目名。如图 9.55 所示。

	A	B
1	报警报表	
2	变量名称	
3	变量描述	
4	报警组	
5	优先级	
6	报警值	
7	报警日期	
8	报警时间	
9	处理建议	
10		

9.55 创建后的报警报表窗口

2. 在工程浏览器中选择“命令语言”下的“自定义函数命令语言”，选择整型自定义函数——void \$System_LongAlarm(LongTag lTag, longtime, long nEvent, long nAction)，双击它。在弹出的命令语言编辑器中输入以下程序：

```
string sGroupName; //报警组名称

string tempstr;    //临时变量

//获取报警组名称

sGroupName=GetGroupName(" ", lTag.Group);

//如果是产生报警，而不是报警确认或报警恢复

if (nEvent==0)

{ //如果变量报警组属于反应车间，则添加报表

if (sGroupName=="反应车间" || sGroupName=="液位")

{

ShowPicture("报警报表画面");
```

```
//向报表中添加变量名称
ReportSetCellString("报警报表", 2, 2, lTag.Name);

//向报表中添加变量描述
ReportSetCellString("报警报表", 3, 2, lTag.Comment);

//向报表中添加变量报警组
ReportSetCellString("报警报表", 4, 2, sGroupName);

//向报表中添加变量优先级
ReportSetCellValue("报警报表", 2, 2, lTag.Priority);

//向报表中添加变量值
ReportSetCellValue("报警报表", 6, 2, lTag);

//向报表中添加变量报警日期
tempstr=StrFromTime( nTime, 1 );
ReportSetCellString("报警报表", 7, 2, tempstr);

//向报表中添加变量报警时间
tempstr=StrFromTime( nTime, 2 );
ReportSetCellString("报警报表", 8, 2, tempstr);

//向报表中添加报警处理建议
ReportSetCellString("报警报表", 9, 2, 处理建议);

//打印报表
ReportPrint2("报警报表");
```

```
}  
  
}
```

系统运行后，当报警产生时，系统自动调用预置自定义函数，将报警信息自动填充报表，结果如图 9.56 所示。

报警报表	
变量名称	液位测量
变量描述	反应车间液位测量
报警组	液位
优先级	200.00
报警值	820.00
报警日期	02-11-18
报警时间	14:00:02
处理建议	

9.56 运行后的报警报表

9.8 利用短信和 E-Mail 发送您的报警信息

随着互连网技术和通讯技术的飞速发展，网络 and 手机已经是现代人生活中离不开的了。如果您能在任何地方都能清楚的了解到现场的生产情况，现在已经不是一种梦想了。

组态王除了可以将画面发布为 WEB 页面，使用户通过 IE 来访问现场的动态画面、实时数据信息、查询历史数据外，还提供了短信和 E-Mail 功能。这两个功能已通用驱动程序的方式提供，使用方便。使用任意一个或两个功能，组态王就可以自动将需要的特定数据或报警信息，通过手机短信的方式或电子邮件的方式发送给指定的用户。

9.8.1 短信功能

使用手机短信功能时,用户需要向硬件厂商购买一个手机模块,另外需要一个手机卡,这样在基础上构成了一个手机。目前组态王支持与西门子 Siemens T35 Terminal 设备之间短信息的发送、接收,设备中数据以短消息的形式传输。短信设备的寄存器表如下所示。

寄存器名称	读写属性	变量类型	寄存器说明
AT	只读	I/O 实数, I/O 整数	联机状态: 当该寄存器的值为 1 时, 表示设备联机通讯正常
CMGF	读/写	I/O 实数	设置短消息格式: 1: 文本格式 0: PDU 编码, 本驱动只支持 PDU 方式
CSCA	读/写	I/O 字符串	设置短消息中心号码
CMGD	只写	I/O 实数, I/O 整数	删除指定序号的短消息: 序号是指短消息在 SIM 卡中储存的位置 CMGD=1, 表示删除 SIM 卡中序号为 1 的短消息。它的值应该从 1 开始。
NEW	只写	I/O 离散	读新短消息命令: 读 SIM 卡中新收到的短消息, 并将其内容写到 MsgNew 和 MsgNec 从寄存器中

MsgNew <i>d</i>	只读	I/O 字符串	读新短消息：返回上次从 SIM 卡中读到的新短消息，用 MsgNew, MsgNec, MsgNed 显示
MsgNec	只读	I/O 字符串	
MsgNed <i>dd</i>	只读	I/O 字符串	
OLD	只写	I/O 离散	读旧短消息命令：读 SIM 卡中已读的短消息，并将其内容写到 MsgOld 和 MsgInf 寄存器中
MsgOld <i>d</i>	只读	I/O 字符串	读已读短消息：返回上次从 SIM 卡中读到的已读短消息，分两部分 MsgOld, MsgInf 显示
MsgInf <i>d</i>	只读	I/O 字符串	
SEND	只写	I/O 离散	发送短消息命令：将 MsgSend 寄存器的内容发送到 Tele 寄存器记录的号码中
Tele	读/写	I/O 字符串	设置接收方电话号码：要发送短消息前，先写该寄存器，接收方电话号码
MsgSend	读/写	I/O 字符串	设置要发送的短消息内容：要发送短消

<i>dd</i>		串	息前, 先将要发送的内容写到该寄存器, 当发送信息超过 63 汉字时, 将发送消息放在 msgsend1 中, 否则 msgsend1 设为空
EDelay	读/写	I/O 整数	设置当通讯失败时, 重启设备的延长时间
SReturn	读/写	I/O 整数	返回信息是否发送成功。1: 成功; 2: 失败。在发送信息之前先将 SReturn 写为 0, 再发送信息, 发送后可以根据该寄存器值判断发送是否成功。
CodeMod	读/写	I/O 整数	(当 CMGF=0 时使用)。编码模式。0: 纯英文编码, 7 位编码。1: 中英文混合文本编码, 8 位编码。说明: 缺省为中英文混合编码。0 编码方式可以发送 160 个英文字母。1 编码方式只能发送 70 个字符。
ReSTime	读/写	I/O 整数	设置短信发送失败时的重发次数, ReSTime=0,1 表示不重发。ReSTime=3 表示发送 3 次。不要太大, 建议不要大于 3

上面只是短信功能的简单使用说明,具体使用方法请参见组态王设备帮助——智能仪表/组态王 SIM 卡短信,或与亚控公司技术支持联系。

9.8.2 E-Mail 功能

组态王可以通过 E-Mail 的方式向外发送较简单的邮件(不超过 256 个字符)。使用本功能之前,应该保证要发送邮件的主机已经连接到了局域网或广域网上。本地不需要特别的设置。

在工程浏览器中选择“设备”栏,双击“新建…”,在弹出的设备创建向导中选择——智能模块>研华>EMail>EMail。建立完设备后,建立相应变量。设备的寄存器及含义如下表所示。如用户的邮箱地址为 testemail01@163.com,用户名称为 username,密码为 password,收件人的地址为 testemail02@sohu.com,则各个寄存器的值设置如表所示:

寄存器名称	含 义	数据类型	变量名称 举例	寄存器值举例
Str0	服务器地址	IO 字符串	服务器	Smt.163.com
Str1	收件人地址	IO 字符串	收件人	testemail02@sohu.com
Str2	发件人地址	IO 字符串	发件人	testemail01@163.com
Str3	用户名称	IO 字符串	用户名	username
Str4	密码	IO 字符串	密码	password
Str5	信件标题	IO 字符串	标题	(128 字节内的字符串)

Str6	信件正文	IO 字符串	正文	(128 字节内的字符串)
Control	发送控制和标识： 0: 停止发送或发送完毕 1: 发送邮件或正在发送	IO 整型	控制	置 1 发送 为 0 时发送完毕 发送过程中设置为 0 停止发送
Status	发送状态： 0: 初始化状态 1: 发送邮件成功 255: 发送邮件失败	IO 整型	状态	1: 发送邮件成功 255: 发送邮件失败



注意：

本功能用于发送较简单的邮件（如发送报警信息等），不能超过 256 个字符！

E-Mail 功能的具体使用方法请参见组态王设备帮助——智能模块/组态王 E-Mail，或咨询亚控技术支持工程师。

第十章 命令语言程序

- ☞ 介绍 KingView 命令语言的语法结构
- ☞ 教您如何编写命令语言程序

10.1 命令语言类型

组态王7.0中命令语言是一种在语法上类似C语言的程序，工程人员可以利用这些程序来增强应用程序的灵活性、处理一些算法和操作等。它具有完备的词法语法查错功能和丰富的运算符、数学函数、字符串函数、控件函数、SQL函数和系统函数。各种命令语言通过“命令语言编辑器”编辑输入，在“组态王7.0”运行系统中被编译执行。

组态王7.0命令语言都是靠事件触发执行的，如定时、数据的变化、键盘键的按下、鼠标的点击等。根据事件和功能的不同，包括应用程序命令语言、热键命令语言、事件命令语言、数据改变命令语言、自定义函数命令语言、动画连接命令语言和画面命令语言等。其中应用程序命令语言、热键命令语言、事件命令语言、数据改变命令语言可以称为“后台命令语言”，它们的执行不受画面打开与否的限制，只要符合条件就可以执行，而画面和动画连接命令语言的执行不受影响。另外可以使用运行系统中的菜单“特殊/开始执行后台任务”和“特殊/停止执行后台任务”来控制应用程序命令语言的“运行时”是否执行，也可以通过修改系统变量“\$启动后台命令语言”的值来实现上述控制，该值置0时

停止执行，置 1 时开始执行。

10.1.1 应用程序命令语言

应用程序命令语言是指在组态王7.0运行系统应用程序启动时、运行期间和程序退出时执行的命令语言程序。如果是在运行系统运行期间,该程序按照指定时间间隔定时执行。

10.1.1.1 认识命令语言编辑器

在工程浏览器的“目录显示区”，选择“文件\命令语言\应用程序命令语言”，则在右边的“内容显示区”出现“请双击这儿进入<应用程序命令语言>对话框...”图标，如图 10.1 所示。

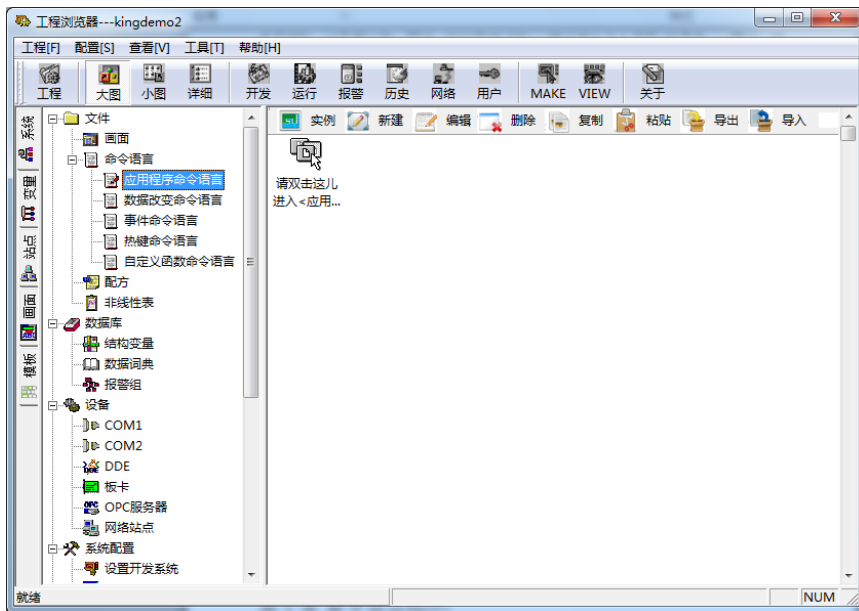


图 10.1 选择应用程序命令语言

双击图标，则弹出“应用程序命令语言”对话框，如图 10.2 所示。

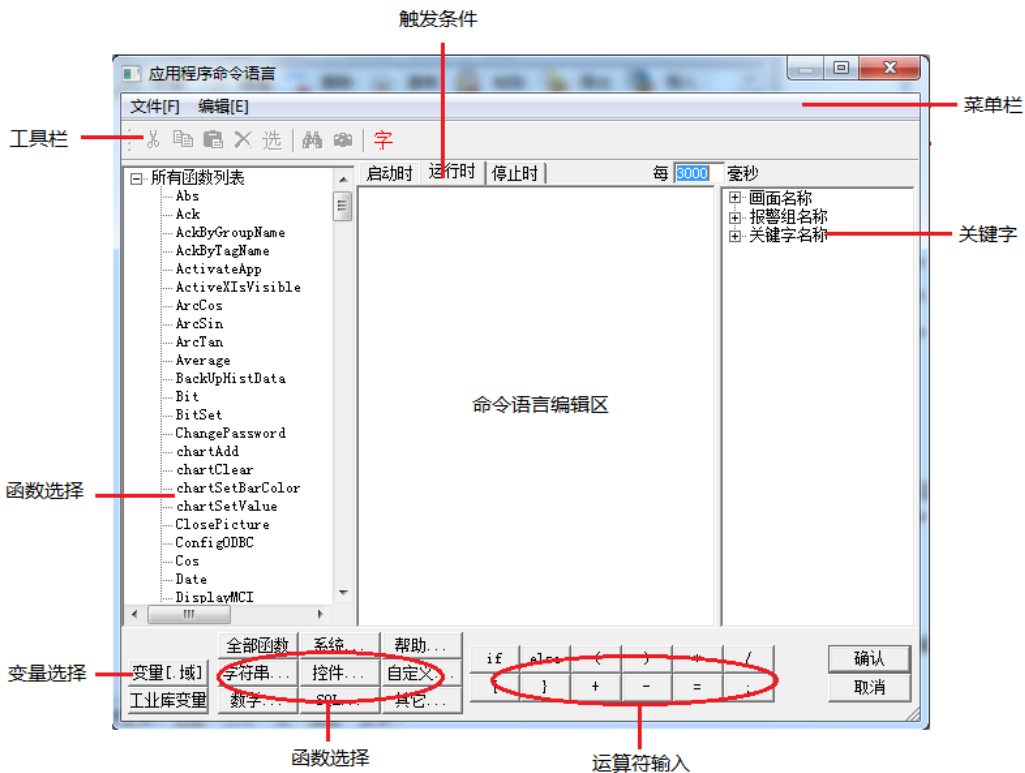


图 10.2 应用程序命令语言

命令语言编辑器是组态王 7.0 提供的用于输入、编辑命令语言程序的地方。编辑器的组成部分如图 10.2 所示。所有命令语言编辑器的大致界面和主要部分及功能都相同，唯一不同的是，按照触发条件的不同，在界面上“触发条件”部分会有所不同。编辑器各部分的大致功能为：

1. 菜单条

提供给编辑器的操作菜单，包括“文件”和“编辑”菜单。

◇ “文件”菜单：

“文件”菜单下有个“确认”和“取消”菜单项，这两个菜单项与编辑器右下角的确认和取消按钮作用相同。

- “确认”表示保存当前在编辑器中内容的修改，然后关闭编辑器；
- “取消”表示直接关闭编辑器，不保存当前在编辑器中内容的修改。

◇ “编辑”菜单：

提供使用编辑器编辑命令语言时提供的一些操作工具，其作用同工具条。具体使用方法请参见本节介绍。

2. 工具栏

提供命令语言编辑时的工具。包括剪切、复制、粘贴、删除、全选、查找、替换、更改命令语言编辑器中的内容的显示字体、字号等。

3. 关键字选择列表

可以在这里直接选择现有的画面名称、报警组名称、其它关键字（如运算连接符等）到命令语言编辑器里。如选中一个画面名称，然后双击它，则该画面名称就被自动添加到了编辑器中。

4. 函数选择

单击某一按钮，弹出相关的函数选择列表，直接选择某一函数到命令语言编辑器中。

- ◇ “全部函数” — 显示组态王 7.0 提供的所有函数列表；
- ◇ “系统” — 只显示系统函数列表；
- ◇ “字符串” — 只显示与字符串操作相关的函数列表；
- ◇ “数学” — 只显示数学函数列表；

- ◇ “SQL” — 只显示 SQL 函数列表；
- ◇ “控件” — 选择 Active X 控件的属性和方法，请参见“第十四章 控件”；
- ◇ “自定义” — 显示自定义函数列表。

点击函数列表树上的某个函数时，自动将该函数添加到命令语言编辑区中。

5. 运算符输入

单击某一个按钮，按钮上标签表示的运算符或语句自动被输入到编辑器中。

6. 变量选择

选择变量或变量的域到编辑器中。单击该按钮时，弹出变量浏览器—“选择变量名”对话框，该对话框的使用请参见第五章 变量定义和管理中的变量浏览器的使用。

7. 命令语言编辑区

输入命令语言程序的区域。

8. 函数在线帮助

当用户不知道函数的用法时，可以单击“帮助”进入在线帮助，查看使用方法或点击函数列表数上的某个函数时，在线帮助区自动显示该函数的说明和实例。


9. 触发条件

触发命令语言执行的条件，不同的命令语言类型有不同的触发条件，下面各节将详细介绍。

10. 命令语言编辑器中的操作

剪切

首先在编辑器中选定要剪切的内容，然后进行剪切，剪切操作后，编辑器中原位置内容消失。剪切的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/剪切”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“剪

切”，或在弹出快捷菜单时按下<Ctrl>+<T>键


◇ 使用 Windows 的快捷键方式：<Ctrl>+<X>。



10.3 命令语言编辑器快捷菜单

复制


首先在编辑器中选中要复制的内容，然后进行复制，复制操作后，编辑器中原位置内容不变。复制的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/复制”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“复制”，或在弹出快捷菜单时按下<Ctrl>+<C>键
- ◇ 使用 Windows 的快捷键方式：<Ctrl>+<C>。

粘贴


首先将鼠标定位到编辑器要粘贴内容的地方，然后进行粘贴，粘贴的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/粘贴”

- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“粘贴”，或在弹出快捷菜单时按下<Ctrl>+<P>键
- ◇ 使用 Windows 的快捷键方式：<Ctrl>+<V>。


删除

首先在编辑器中选择要删除的内容，然后进行删除，删除的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/删除”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“删除”，或在弹出快捷菜单时按下<Ctrl>+<D>键
- ◇ 使用键盘上的键。

全选


当需要将命令语言编辑器中的内容全部选定时，可以使用“全选”功能。全选的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/全选”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“全选”，或在弹出快捷菜单时按下<Ctrl>+<L>键
- ◇ 使用键盘上的<Ctrl>+<A>键。

查找字符

当需要在命令语言编辑器中查找某个字符时，可以使用“查找”功能。启动查找功

能的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/查找”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“查找”，或在弹出快捷菜单时按下<Ctrl>+<F>键。

选择查找后，系统弹出“查找”对话框，如图10.4所示。



10.4 查找对话框

首先将鼠标光标定位到编辑器中要开始查找的地方。

- 打开查找对话框，在“查找内容”对话框中输入要查找的字符串。
- 如果查找的内容是一个单字、词，则选择“全字匹配”选项，否则不选。不选时包含在其它字符串中的相同字符也会被查找出来。
- 如果希望在查找的过程中区分字符的大小写，则选择“区分大小写”选项，否则不选。不选时，与查找内容相同的无论大小写的字符都会被查找出来。
- “方向”表示查找时从当前鼠标光标位置向程序头还是程序尾方向


查找。如果选择“向上”，则从光标当前位置开始，向程序的开始方向查找；如果选择“向下”，则从当前光标位置开始，向程序的末尾方向查找。

- 当每查找到一个相匹配的字符时，编辑器中会将该位置的字符选中，以高亮度显示。如果希望继续查找，则单击“查找下一个”按钮。如果要停止查找，单击“取消”按钮，关闭对话框。

替换字符

当需要将命令语言编辑器中的某个字符替换为另外的字符时，可以使用“替换”功能。

启动替换功能的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/替换”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“替换”，或在弹出快捷菜单时按下<Ctrl>+<E>键。

替换是将查找的字符替换为用户指定的字符。选择替换后，系统弹出“替换”对话框，如图10.5所示。



10.5 替换对话框


- 首先将鼠标光标定位到编辑器中要开始查找的地方。打开替换对话框，在“查找内容”编辑框中输入要被替换掉的内容，在“替换为”编辑框中输入替换后的内容。
- 如果查找的内容是一个单字、词，则选择“全字匹配”选项，否则不选。不选时包含在其它字符串中的相同字符也会被查找出来。
- 如果希望在查找的过程中区分字符的大小写，则选择“区分大小写”选项，否则不选。不选时，与查找内容相同的无论大小写的字符都会被查找出来。当每查找到一个相匹配的字符时，编辑器中会将该位置的字符选中，以高亮度显示。
- 如果希望一个个的替换目标字符，首先单击“查找下一个”按钮，当查找到时，如果需要替换，单击“替换”按钮。则当前查找到的字符将被替换。
- 如果希望一次性将所有字符都替换掉，则直接单击“全部替换”按钮，

则当前编辑器中的符合条件的字符都将被替换。

- 替换完成后，单击“取消”，关闭对话框。


改变字符字体型号、大小

命令语言编辑器中的字体为默认字体，用户可以按照习惯修改编辑器中字符的字体、字号等。打开字体对话框的方法有以下几种：

- ◇ 选择编辑器菜单“编辑/字体”
- ◇ 单击工具条上的  按钮
- ◇ 在编辑器区界面上单击鼠标右键弹出如图 10.3 所示的快捷菜单中选择“字体”，或在弹出快捷菜单时按下<Ctrl>+<O>键。字体选择对话框如图 10.6 所示。该对话框是标准对话框，选择完成后，编辑器中的字体都变为所选择的字体、字体样式和大小等。



10.6 字体选择对话框

 注意：

在输入命令语言时，除汉字外，其它关键字，如标点符号必须以西文状态输入。

选择函数名称、画面名称等到编辑器中

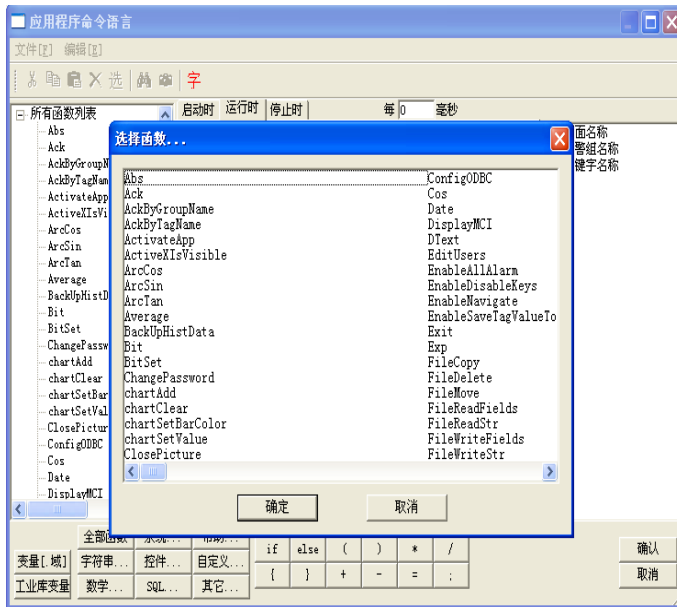
命令语言编辑器提供了众多的方便，使用户可以直接选择诸如画面名称、报警组名称、函数等到编辑器中，避免了烦杂的手工输入。



例如：在运行系统启动时希望显示一个画面。

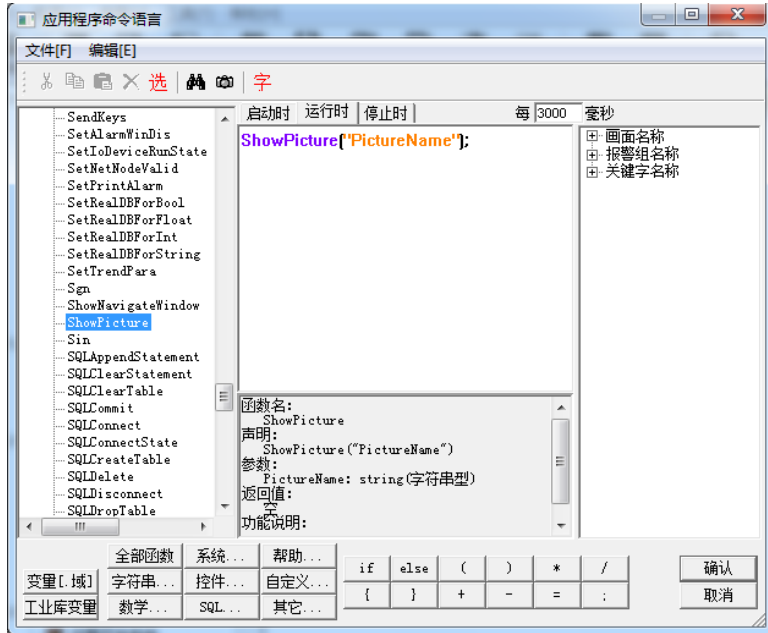
这个问题除了可以在系统配置—运行系统配置中配置主画面外，还可以使用应用程序命令语言—启动时—来打开该画面。

- 1) 打开应用程序命令语言编辑器，选择“启动时”标签。
- 2) 单击“全部函数”按钮，弹出“选择函数”对话框，如图10.7所示。



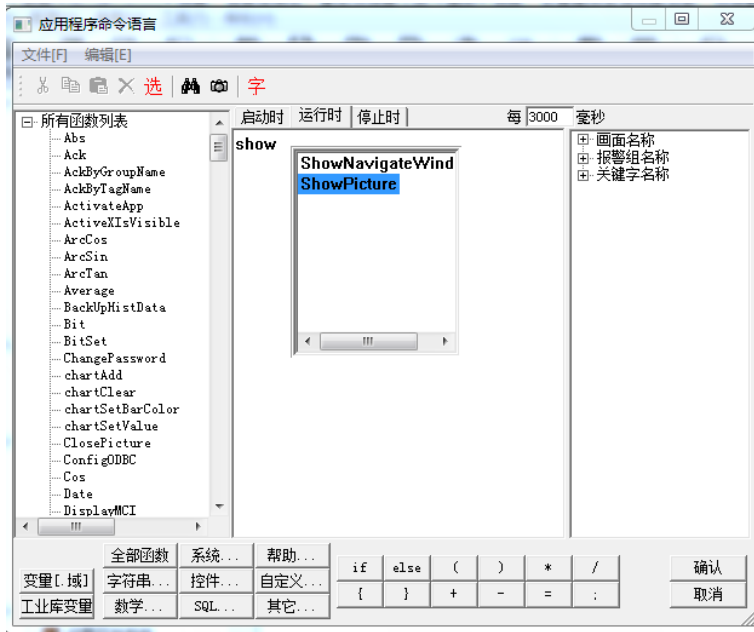
10.7 选择函数

- 3) 在“函数列表”找到“ShowPicture”函数，选择该项后，单击对话框上的“确定”按钮，后直接双击该函数名称，对话框被关闭，函数及其参数整体被选择到了编辑器中，如图10.8所示。



10.8 函数选择到编辑器中

或者在编辑框中输入函数前面字符，弹出函数快捷列表，从快捷列表中选择函数，快速的输入的编辑框中。



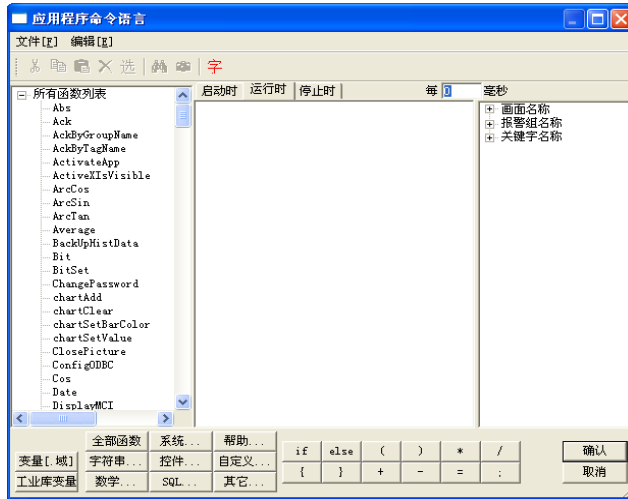
- 4) 函数ShowPicture中的参数为要显示的画面名称。选择函数默认的参数并删除（保留引号），保留光标位于函数参数位置处（引号之间）；单击编辑器右侧列表中的“画面名称”上的“+”，展开画面名称列表，显示当前工程中已有画面的画面名称。选择要显示的画面名称，并双击它。则该画面名称自动添加到了函数的参数位置。至此，该例中显示画面的程序编辑工作完成。

按照上面的例子，同样可以选择报警组名称、连接运算符、变量等到编辑器中。

10.1.1.3 应用程序命令语言的定义

组态王7.0应用程序命令语言只能定义一个。当选择“运行时”标签时，会有输入执

行周期的编辑框“每……毫秒”。输入执行周期，则组态王7.0运行系统运行时，将按照该时间周期性的执行这段命令语言程序，无论打开画面与否。选择“启动时”标签，在该编辑器中输入命令语言程序，该段程序只在运行系统程序启动时执行一次。选择“停止时”标签，在该编辑器中输入命令语言程序，该段程序只在运行系统程序退出时执行一次。



10.9 应用程序命令语言

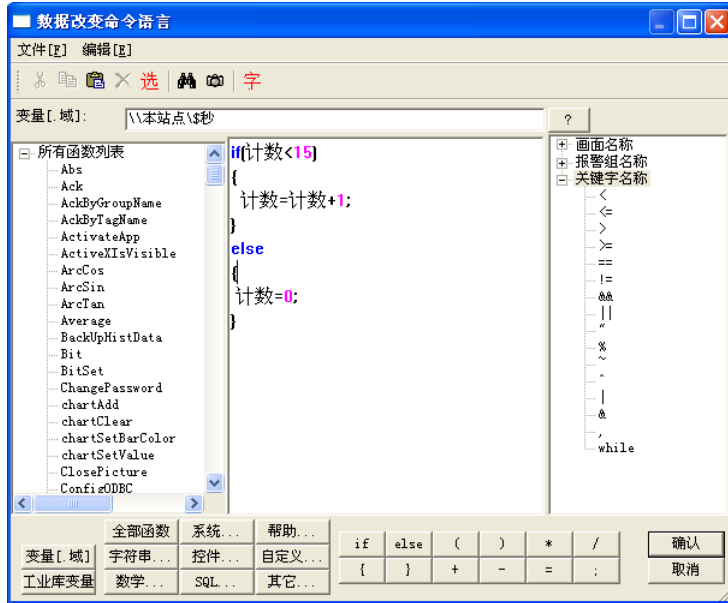
10.1.2 数据改变命令语言

数据改变命令语言触发的条件为连接的变量或变量的域的值发生了变化。

在工程浏览器中选择“命令语言\数据改变命令语言”，在浏览器右侧双击“新建……”，弹出数据改变命令语言编辑器。如图10.10所示。

在命令语言编辑器“变量[.域]”编辑框中输入或通过单击“？”按钮来选择变量名称（如：原料罐液位）或变量的域（如：原料罐液位.Alarm）。这里可以连接任何类型的

变量和变量的域，如离散型、整型、实型、字符串型等。当连接的变量的值发生变化时，系统会自动执行该命令语言程序。数据改变命令语言可以按照需要定义多个。



10.10 数据改变命令语言编辑器

 注意：

在使用“事件命令语言”或“数据改变命令语言”过程中要注意防止死循环。

例如，变量A变化引发数据改变命令语言程序中含有命令B=B+1，若用B变化再引发事件命令语言或数据改变命令语言的程序中不能再有类似A=A+1的命令。

数据改变命令语言和事件命令语言的条件如果引用远程变量，则下面的命令语言不执行。

10.1.3 事件命令语言

事件命令语言是指当规定的表达式的条件成立时执行的命令语言。如某个变量等于定值，某个表达式描述的条件成立。

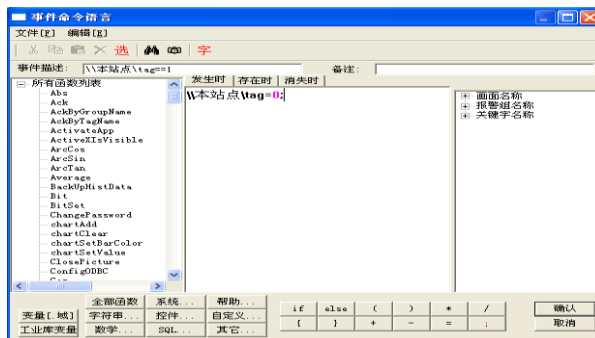
在工程浏览器中选择“命令语言\事件命令语言”，在浏览器右侧双击“新建……”，弹出事件命令语言编辑器，如图10.11所示。

◇ 事件命令语言有三种类型：

- **发生时：**事件条件初始成立时执行一次。
- **存在时：**事件存在时定时执行，在“每……毫秒”编辑框中输入执行周期，则当事件条件成立存在期间周期性执行命令语言。如图10.12所示。
- **消失时：**事件条件由成立变为不成立时执行一次。

◇ **事件描述：**指定命令语言执行的条件。

◇ **备注：**对该命令语言作一些说明性的文字。



10.11 事件命令语言编辑器

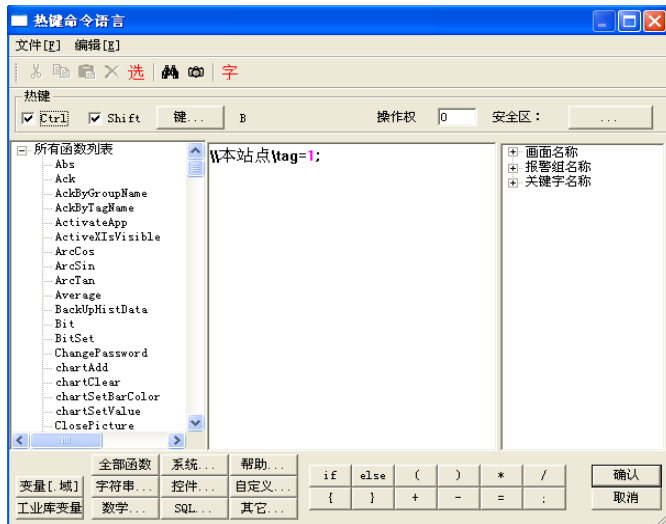
事件描述:	反应罐液位=10	备注:	反应罐液位等于10执行
发生时	存在时	消失时	每 3000 毫秒

10.12 事件命令语言 — 存在时

10.1.4 热键命令语言

“热键命令语言”链接到工程人员指定的热键上，软件运行期间，工程人员随时按下键盘上相应的热键都可以启动这段命令语言程序。热键命令语言可以指定使用权限和操作安全区。

输入热键命令语言时，在工程浏览器的目录显示区，选择“文件\命令语言\热键命令语言”，双击右边的内容显示区出现“新建...”图标，弹出热键命令语言编辑器，如图10.13所示。



10.13 热键命令语言编辑器

当Ctrl和Shift左边的复选框被选中时，表示此键有效。如图10.14所示



图 10.14 热键定义

热键定义区的右边为键按钮选择区，用鼠标单击此按钮，则弹出如图10.15所示的对话框：

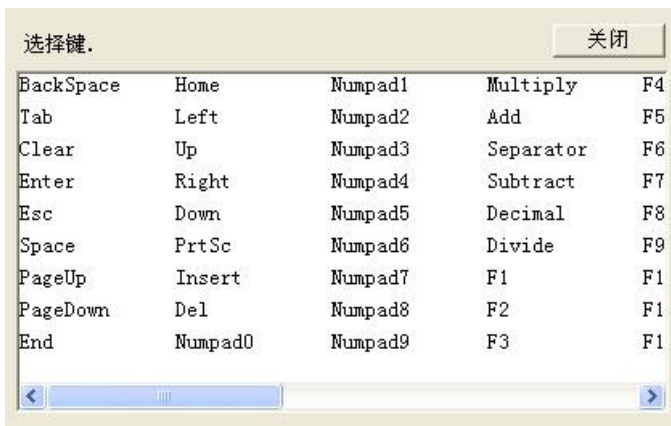
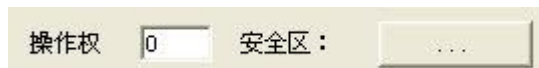


图 10.15 热键选择

在此对话框中选择一个键，则此键被定义为热键，还可以与Ctrl和Shift形成组合键。

热键命令语言可以定义安全管理，安全管理包括操作权限和安全区，两者可单独使用，也可合并使用。如图10.16所示。比如：设置操作权限为918。只有操作权限大于等于918的操作员登录后按下热键时，才会激发命令语言的执行。



10.16 热键的安全管理定义

10.1.5 用户自定义函数

如果组态王 7.0 提供的各种函数不能满足工程的特殊需要，组态王 7.0 还提供用户自定义函数功能。用户可以自己定义各种类型的函数，通过这些函数能够实现工程特殊的需要。如特殊算法、模块化的公用程序等，都可通过自定义函数来实现。

自定义函数是利用类似 C 语言来编写的一段程序，其自身不能直接被组态王 7.0 触发调用，必须通过其他命令语言来调用执行。

编辑自定义函数时，在工程浏览器的目录显示区，选择“文件\命令语言\自定义函数命令语言”，在右边的内容显示区出现“新建”图标，用左键双击此图标，将出现“自定义函数命令语言”对话框，如图 10.17 所示。

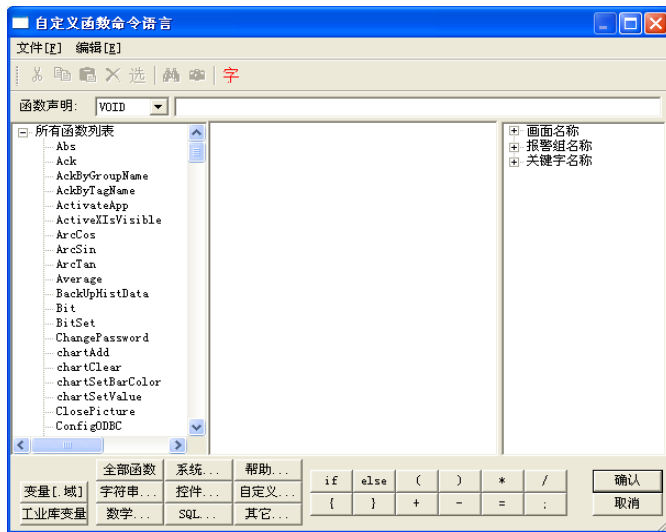


图 10.17 自定义函数命令语言编辑器

10.1.5.1 自定义函数里的概念

自定义函数里有六个关键字，分别是：LONG、FLOAT、STRING、BOOL、VOID、RETURN，大小写均可，语法含义和 C 语言类似：

LONG：表示数据/变量类型为整型；

FLOAT：表示数据/变量类型为实型；

STRING：表示数据/变量类型为字符型；

BOOL：表示数据/变量类型为布尔型；

VOID：表示函数无返回值或返回值类型为空 (NULL) 类型；

RETURN：表示函数的返回值，并且返回到主调函数中。

自定义函数的语法与 C 语言中定义子函数的格式类似。自定义函数命令语言是由变量定义部分和可执行语言组成的单独实体。

自定义函数定义的内容为：

- ✧ 自定义函数类型（函数返回值类型）
- ✧ 函数名和参数类型及名称
- ✧ 函数体内容

10.1.5.2 自定义函数的定义和使用

如图 10.17 所示，为自定义函数的编辑器。在“函数声明”后的列表框中选择函数返回值的数据类型，包括下面五种：VOID、LONG、FLOAT、STRING、BOOL，按照需要选择一种。如果函数没有返回值，则直接选择“VOID”。

在“函数声明”数据类型后的文本框中输入该函数的名称，不能为空。函数名称的命

名应该符合组态王 7.0 的命名规则，不能为组态王 7.0 中已有的关键字或变量名。函数名后应该加小括号“()”号，如果函数带有参数，则应该在括号内声明参数的类型和参数名称。参数可以设置多个。

在“函数体（执行代码）”编辑框中输入要定义的函数体的程序内容。在函数内容编辑区内，可以使用自定义变量。函数体内容是指自定义函数所要执行的功能。函数体中的最后部分是返回语句。如果该函数有返回值，则使用 Return Value（Value 为某个变量的名称）。对于无返回值的函数也可以使用 Return，但只能单独使用 Return，表示当前命令语言或函数执行结束。



注意：

自定义函数中的函数名称和在函数中定义的变量不能与组态王 7.0 中定义的变量、组态王 7.0 的关键字、函数名等相同。

具体使用方法，请参见下面的例子：



例 1： VOID 型函数，实现阶乘

返回类型为：VOID；

函数名为：jiechen(long Ref, long Ret)

函数体的内容为：

```
//本函数为无返回值型函数,实现阶乘运算,参加运算的变量均在函数的参数中
//Ref 为参加运算的变量, Ret 为计算结果
long a; //自定义变量,控制阶乘循环次数
```

```
long mul; //自定义变量, 存储阶乘运算结果

a=1;

mul=1;

if (Ref<=0)

mul=1;

else

{ while (a<=Ref)

{mul=mul*a;

a=a+1;

}}

Ret=mul;

return; //函数执行结束
```

定义完成后,在组态王 7.0 自定义函数内容区出现“VOID jiechen(long Ref, long Ret)”函数。如在按钮命令语言中调用,实现一个数的阶乘运算,在组态王 7.0 中定义整型变量为: 因数, 结果, 在按钮命令语言中输入: jiechen(因数, 结果)。则变量“结果”得到的值为计算结果。



例 2: 有返回值类型, LONG 型

返回类型为: LONG

函数名为: function2(long nTemp2)

函数体的内容为 (实现累加):

```
nTemp2=nTemp2+1; //累加;
return nTemp2; //返回 nTemp2 的值;
```

定义完成后, 在组态王 7.0 自定义函数内容区出现 “LONG function2(long nTemp2)” 函数。调用时, 如组态王 7.0 的整型变量为: 计数 2, 在命令语言中输入: function2(计数 2)。



例 3: 实现将给定的角度转换成弧度

返回类型为: float

函数名为: 弧度转换(float Ref)

函数体内容为:

```
float pai; //圆周率
float Ret;
pai=3.14159265;
Ret=(Ref/180)*pai;
return Ret; //返回结果
```

 注意:

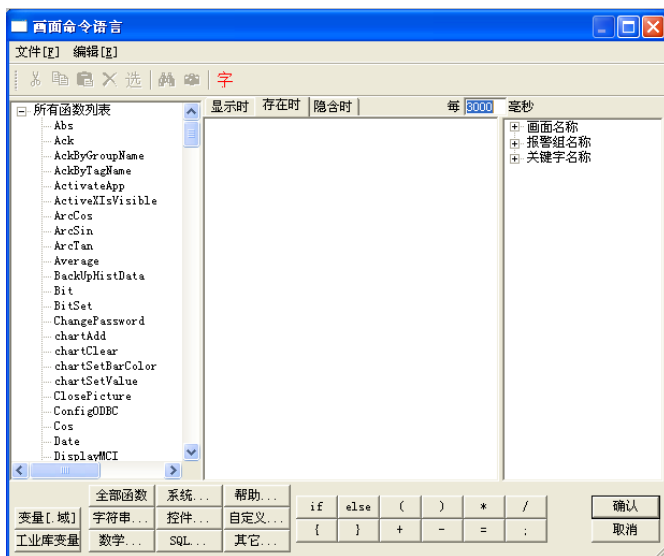
定义的函数支持有返回值和无返回值对应的是RETURN和VOID,函数传进的参数,若是变量在执行完函数后均保持(传址)。函数的返回值必须和被赋值的变量类型一致。

当很多的命令语言里需要一段同样的程序时,可以定义一个自定义函数,在命令语言里来调用,减少了手工的输入量,减小了程序的规模,同时也使得程序的修改和调试变得更为简明、方便。

除了用户自定义函数外,组态王7.0提供了三个报警预置自定义函数,利用这些函数,可以方便的在报警产生时做一些处理。关于报警预置自定义函数的具体使用方法请见“第九章 报警和事件系统”。Active X控件的事件函数编辑和使用与自定义函数相同,定义的事件函数的函数名称也会显示在工程浏览器中自定义函数目录下,但该类型函数的编辑只能通过是在控件属性界面进行,具体使用方法请参见“第十四章 控件”。

10.1.6 画面命令语言

画面命令语言就是与画面显示与否有关系的命令语言程序。画面命令语言定义在画面属性中。打开一个画面,选择菜单“编辑/画面属性”,或用鼠标右键单击画面,在弹出的快捷菜单中选择“画面属性”菜单项,或按下<Ctrl>+<W>键,打开画面属性对话框,在对话框上单击“命令语言...”按钮,弹出画面命令语言编辑器,如图10.18所示。



10.18 画面命令语言编辑器

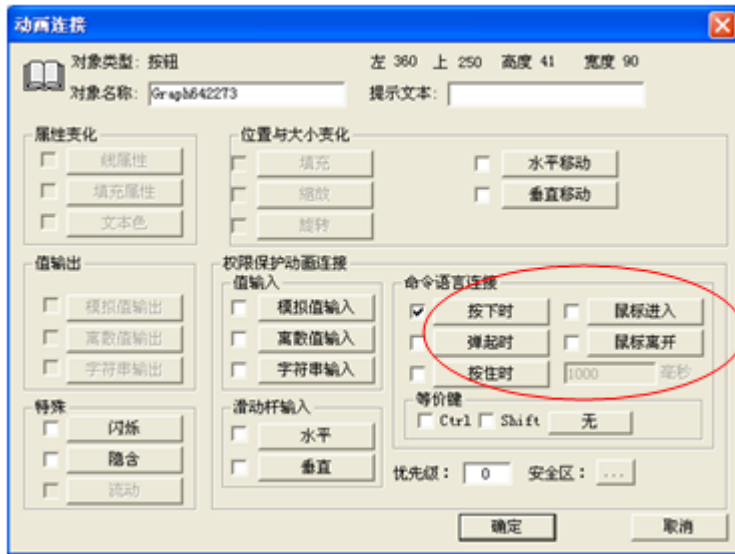
画面命令语言分为三个部分：显示时、存在时、隐含时。

- ◇ **显示时**：打开或激活画面为当前画面，或画面由隐含变为显示时执行一次。
- ◇ **存在时**：画面在当前显示时，或画面由隐含变为显示时周期性执行，可以定义指定执行周期，在“存在时”中的“每…毫秒”编辑框中输入执行的周期时间。
- ◇ **隐含时**：画面由当前激活状态变为隐含或被关闭时执行一次。

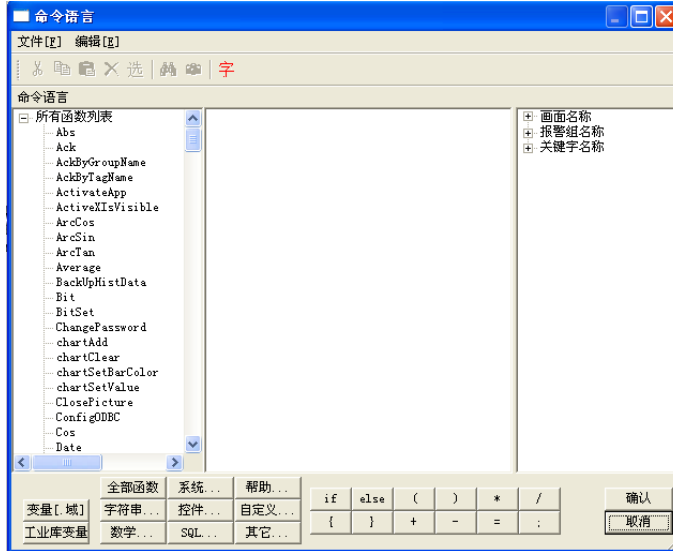
只有画面被关闭或被其它画面完全遮盖时，画面命令语言才会停止执行。只与画面相关的命令语言可以写到画面命令语言里——如画面上动画的控制等，而不必写到后台命令语言中——如应用程序命令语言等，这样可以减轻后台命令语言的压力，提高系统运行的效率。

10.1.7 动画连接命令语言

对于图素，有时一般的动画连接表达式完成不了工作，而程序只需要点击一下画面上的按钮等图素才执行，如点击一个按钮，执行一连串的动作，或执行一些运算、操作等。这时可以使用动画连接命令语言。该命令语言是针对画面上的图素的动画连接的，组态王7.0中的大多数图素都可以定义动画连接命令语言。如在画面上放置一个按钮，双击该按钮，弹出动画连接对话框，如图10.19所示。



10.19 图素动画连接动画框中的命令语言连接



10.20 图素动画连接命令语言编辑器

在“命令语言连接”选项中包含三个选项：

- ◇ **按下时**：当鼠标在该按钮上按下时，或与该连接相关联的热键按下时执行一次。
- ◇ **弹起时**：当鼠标在该按钮上弹起时，或与该连接相关联的热键弹起时执行一次。
- ◇ **按住时**：当鼠标在该按钮上按住，或与该连接相关联的热键按住，没有弹起时周期性执行该段命令语言。按住时命令语言连接可以定义执行周期，在按钮后面的“毫秒”标签编辑框中输入按钮被按住时命令语言执行的周期。
- ◇ **鼠标进入时**：当鼠标进入该按钮的矩形区域时，执行一次。

◇ **鼠标离开时**：当鼠标离开该按钮的矩形区域时，执行一次。

单击上述任何一个按钮都会弹出动画连接命令语言编辑器，如图10.20所示。其用法与其它命令语言编辑器用法相同。

动画连接命令语言可以定义关联的动作热键，如图10.19，单击“等价键”中的“无”按钮，可以选择关联的热键，也可以选择<Ctrl>、<Shift>与之组成组合键。运行时，按下此热键，效果同在按钮上按下鼠标键相同。定义有动画连接命令语言的图素可以定义操作权限和安全区，只有符合安全条件的用户登录后，才可以操作该按钮。



注意：

动作热键设置对鼠标进入和鼠标离开两种动画连接命令语言无效。

10.2 命令语言语法


命令语言程序的语法与一般C程序的语法没有大的区别，每一程序语句的末尾应该用分号“;”结束，在使用if…else…、while()等语句时，其程序要用花括号“{}”括起来。

10.2.1 运算符

用运算符连接变量或常量就可以组成较简单的命令语言语句，如赋值、比较、数学运算等。命令语言中可使用的运算符以及算符优先级与连接表达式相同。运算符有以下几种：

- ～ 取补码，将整型变量变成“2”的补码。
- * 乘法

/	除法
%	模运算
+	加法
-	减法（双目）
&	整型量按位与
	整型量按位或
^	整型量异或
&&	逻辑与
	逻辑或
<	小于
>	大于
<=	小于或等于
>=	大于或等于
==	等于（判断）
!=	不等于
=	等于（赋值）

 注意：

除上述运算符以外，还可使用：

— 取反，将正数变为负数（单目）。

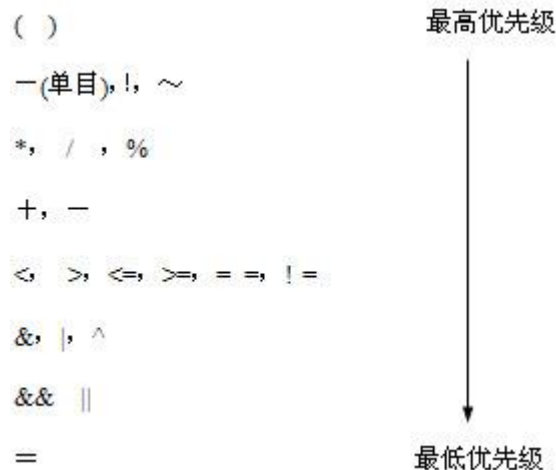
! 逻辑非

() 括号，保证运算按所需次序进行。

增强运算功能

运算符的优先级：

下面列出算符的运算次序，首先计算最高优先级的算符，再依次计算较低优先级的算符。同一行的算符有相同的优先级。



表达式举例：

复杂的表达式： 开关=1 液面高度>50&&液面高度<80(开关 1 || 开关 2)&&(液面高度. alarm)

10.2.2 赋值语句

赋值语句用得最多，语法如下：

变量（变量的可读写域）= 表达式；

可以给一个变量赋值，也可以给可读写变量的域赋值。



例如：

自动开关=1； 表示将自动开关置为开(1表示开, 0表示关)

颜色=2； 将颜色置为黑色(如果数字2代表黑色)

反应罐温度.priority=3； 表示将反应罐温度的报警优先级设为3

10.2.3 If-Else 语句

If-Else 语句用于按表达式的状态有条件地执行不同的程序，可以嵌套使用。语法为：

IF(表达式)

{

一条或多条语句；

}

ELSE

{

一条或多条语句；

}



注意：

if-else语句里如果是单条语句可省略花括弧“{}”，多条语句必须在一对花括弧“{}”中，ELSE分支可以省略。



例1:

```
if (step == 3)
    颜色="红色";
```

上述语句表示当变量step与数字3相等时，将变量颜色置为“红色”（变量“颜色”为内存字符串变量）



例2:

```
if (出料阀 == 1)
    出料阀=0;    //将离散变量“出料阀”设为0状态
else
    出料阀=1;
```

上述语句表示将内存离散变量“出料阀”设为相反状态。If-else里是单条语句可以省略“{}”。



例3:

```
if (step == 3)
{
    颜色="红色";
```


```
        反应罐温度.priority=1;
    }
else
    {
        颜色="黑色";
        反应罐温度.priority=3;
    }
```

上述语句表示当变量step与数字3相等时，将变量颜色置为“红色”（变量“颜色”为内存字符串变量），反应罐温度的报警优先级设为1；否则变量颜色置为“黑色”，反应罐温度的报警优先级设为3。

10.2.4 While（）语句

当while（）括号中的表达式条件成立时，循环执行后面“{ }”内的程序。语法如下：

```
WHILE(表达式)
{
    一条或多条语句(以; 结尾)
}
```

 注意:

同IF语句一样，WHILE里的语句若是单条语句，可省略花括弧“{ }”外，但若是多条语句必须在一对花括弧“{ }”中。这条语句要慎用，否则，会造成死循环。

 例1:

```
while (循环<=10)
{
    ReportSetCellvalue("实时报表",循环, 1, 原料罐液位);
    循环=循环+1;
}
```

当变量“循环”的值小于等于10时，向报表第一列的1~10行添入变量“原料罐液位”的值。应该注意使while表达式条件满足，然后退出循环。

10.2.5 命令语言程序的注释方法

命令语言程序添加注释，有利于程序的可读性，也方便程序的维护和修改。组态王7.0的所有命令语言中都支持注释。注释的方法分为单行注释和多行注释两种。注释可以在程序的任何地方进行。

单行注释在注释语句的开头加注释符“//”:

 例1:


```
//设置装桶速度
```

```
if(游标刻度>=10) //判断液位的高低
```

```
装桶速度=80;
```

多行注释是在注释语句前加“/*”，在注释语句后加“*/”。多行注释也可以用在单行注释上。



例 2:

```
if(游标刻度>=10) /*判断液位的高低*/
```

```
装桶速度=80;
```



例 3:

```
/*判断液位的高低
```

```
改变装桶的速度*/
```

```
if(游标刻度>=10)
```

```
{装桶速度=80;}
```

```
else
```

```
装桶速度=60;
```



注意:

多行注释不能嵌套使用。

10.3 命令语言执行中如何跟踪变量的值

命令语言一旦运行起来，往往看到的是最终的结果，如果结果出现差错，就需要查看命令语言的执行过程——调试命令语言。组态王7.0提供了一个函数——Trace（），该函数可以将规定的信息发送到组态王7.0信息窗口中，类似于程序的调试，根据这些信息，用户可以了解到命令语言执行的过程和期间变量的值。该函数可以添加到命令语言程序的任何需要跟踪的位置，当命令语言调试完成后，可以将其删除。函数的具体使用方法请参见组态王7.0函数手册。

10.4 在命令语言中使用自定义变量

自定义变量是指在组态王 7.0 的命令语言里单独指定类型的变量，这些变量的作用域为当前的命令语言，在命令语言里，可以参加运算、赋值等。当该命令语言执行完成后，自定义变量的值随之消失，相当于局部变量。自定义变量不被计算在组态王 7.0 的点数之中。适用于应用程序命令语言、事件命令语言、数据改变命令语言、热键命令语言、自定义函数、画面命令语言、动画连接命令语言、控件事件函数等。自定义变量功能的提供可以极大的方便用户编写程序。



注意：

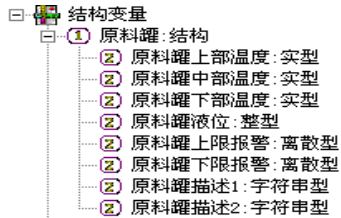
- 1、 自定义变量在使用之前必须要先定义
- 2、 自定义变量没有“域”的概念，只有变量的值

自定义变量的类型有 **BOOL**（离散型）、**LONG**（长整型）、**FLOAT**（实数型）、**STRING**（字符串型）和自定义结构变量类型。其在命令语言语言中的使用方法与组

态王 7.0 变量相同。



例：在结构变量中定义一个结构，如图 10.21 所示。设计一个求原料罐上、下平均温度的自定义函数。



10.21 结构变量

函数返回值类型为：FLOAT，

函数名称及参数表为：平均温度(原料罐 yuanliao1)

函数体程序为：

```
float 平均温度1;
```

```
平均温度1=(yuanliao1. 原料罐上部温度+yuanliao1. 原料罐下部温度)/2;
```

```
return 平均温度1;
```

其中“原料罐”为已定义的结构；“yuanliao1”为自定义结构变量，它继承原结构的所有成员作为自己的成员；“平均温度1”为自定义变量，作为函数的返回值。

10.5 命令语言函数及使用方法

“组态王7.0”支持使用内建的复杂函数，其中包括字符串函数、数学函数、系统函数、控件函数、报表函数、SQL函数、配方函数、报警函数及其它函数，具体见《组态王

7.0命令语言函数速查手册》。

第十一章 组态王运行系统

- ☞ 介绍 KingView 运行软件的相关知识
- ☞ 教您如何运行您的应用程序

“组态王”软件包由工程管理器 ProjectManage、工程浏览器 TouchExplorer 和画面运行系统 TouchView 三部分组成。其中工程浏览器内嵌组态王画面制作开发系统，生成人机界面工程。画面制作开发系统中设计开发的画面工程在 TouchView 运行环境中运行。TouchExplorer 和 TouchView 各自独立，一个工程可以同时被编辑和运行，这对于工程的调试是非常方便的。本章主要通过 TouchView 的菜单命令来介绍组态王画面运行系统。

11.1 配置运行系统

在运行组态王工程之前首先要在开发系统中对运行系统环境进行配置。在开发系统中单击菜单栏“配置\运行系统”命令或工具条“运行”按钮或工程浏览器“工程目录显示区\系统配置\设置运行系统”按钮后，弹出“运行系统设置”对话框，如图 11.1 所示。



图 11.1 运行系统设置_运行系统外观

“运行系统设置”对话框由三个配置属性页组成：

1、“运行系统外观”属性页

此属性页中各项的含义和使用介绍如下：

- ◇ **启动时—最大化：** TouchView 启动时占据整个屏幕。
- ◇ **启动时—缩成图标：** TouchView 启动时自动缩成图标。
- ◇ **窗口外观—标题条文本：** 此字段用于输入 TouchView 运行时出现在标题栏中的标题。

若此内容为空，则 TouchView 运行时将隐去标题条，全屏显示。

- ◇ **窗口外观—系统菜单：** 选择此选项使 TouchView 运行时标题栏中带有系统菜单框。
- ◇ **窗口外观—最小化按钮：** 选择此选项使 TouchView 运行时标题栏中带有最小化按钮。

- ◇ **窗口外观—最大化按钮:**选择此选项使 TouchView 运行时标题栏中带有最大化按钮。
- ◇ **窗口外观—可变大小边框:**选择此选项使 TouchView 运行时,可以改变窗口大小。
- ◇ **窗口外观—标题条中显示工程路径:**选择此项使当前应用程序目录显示在标题栏。
- ◇ **菜单:**选择此选项使 TouchView 运行时带有菜单。

2、“主画面配置”属性页

规定 TouchView 画面运行系统启动时自动调入的画面,如果几个画面互相重叠,最后调入的画面在前面。单击“主画面配置”属性页,则此属性页对话框弹出,同时属性页画面列表对话框中列出了当前应用程序所有有效的画面,选中的画面加亮显示。如图 11.2 所示。

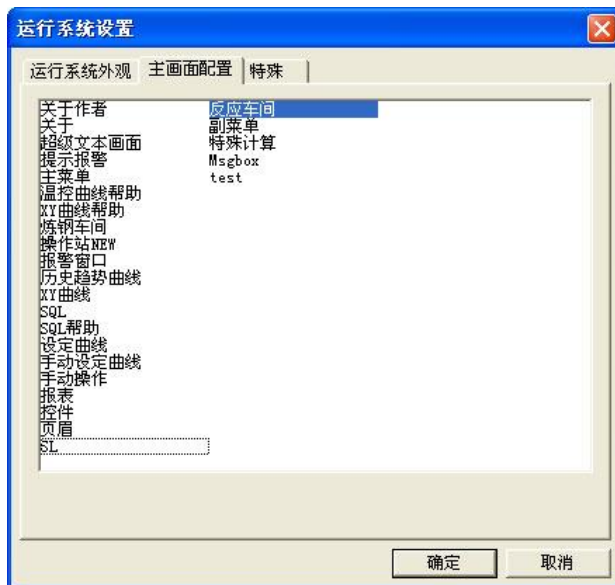


图 11.2 运行系统设置_主画面配置

3、“特殊”属性页

此属性页对话框用于设置运行系统的基准频率等一些特殊属性，单击“特殊”属性页，则此属性页对话框弹出，如图 11.3 所示。

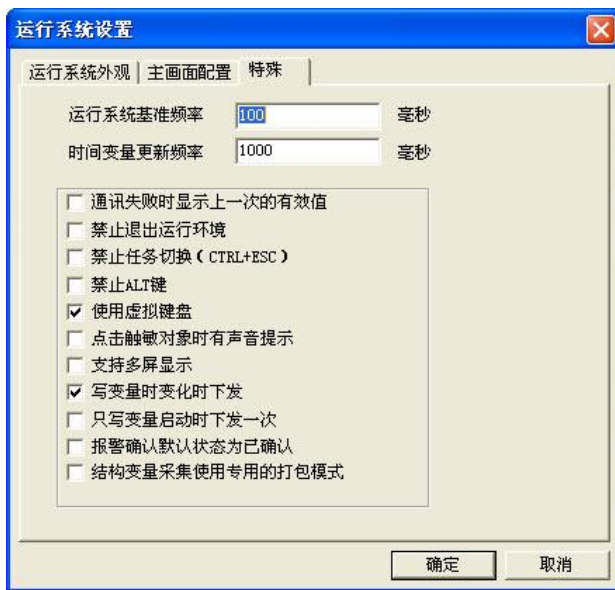


图 11.3 运行系统设置_特殊

- ◇ **运行系统基准频率**：是一个时间值。所有其他与时间有关的操作选项（如：有“闪烁”动画连接的图形对象的闪烁频率、趋势曲线的更新频率、后台命令语言的执行）都以它为单位，是它的整数倍。
- ◇ **时间变量更新频率**：用于控制 TouchView 在运行时更新数据库中时间变量（\$毫秒、\$秒、\$分、\$时等）。
- ◇ **通讯失败时显示上一次的有效值**：用于控制组态王中的 IO 变量在通讯失败后在画

面上的显示方式。选中此项后，对于组态王画面上 IO 变量的“值输出”连接，在设备通讯失败时画面上将显示组态王最后采集的数据值，否则将显示“???”。

- ◇ **禁止退出运行环境：**选择此选项后，其左边小方框内出现“√”号。选择此选项使 TouchView 启动后，除关机外不能退出。
- ◇ **禁止任务切换 (Ctrl—Esc)：**选择此选项后，其左边小方框内出现“√”号。选择此选项将禁止“Ctrl—Esc”键，用户不能作任务切换。
- ◇ **禁止 Alt 键：**选择此选项后，其左边小方框内出现“√”号。选择此选项将禁止“Alt”键，用户不能用 ALT 键调用菜单命令。



注意：

若将上述所有选项选中时，只有使用组态王提供的内部函数 Exit (Option) 退出。

-
- ◇ **使用虚拟键盘：**选择此选项后，其左边小方框内出现“√”号。画面程序运行中，当需要操作者使用键盘时，比如输入模拟值，则弹出模拟键盘窗口，操作者用鼠标在模拟键盘上选择字符即可输入。
 - ◇ **点击触敏对象时有声音提示：**选中此项后，其左边小方框内出现“√”号。则系统运行时，鼠标单击按钮等图素时，蜂鸣器发出声音。
 - ◇ **支持多屏显示：**选择此选项后，其左边小方框内出现“√”号。选择此选项支持多显卡显示，可以一台主机接多个显示器，组态王画面在多个显示器上显示。
 - ◇ **写变量时变化触发：**选择此选项后，如果变量的采集频率为 0，组态王写变量的时候，只有变量值发生变化才写，否则不写。

- ◇ 只写变量启动时下发一次：对于只写变量，选择此选项后，组态王运行系统启动时，将初始值向下写一次，否则不写。

11.2 运行系统菜单详解

配置好运行系统之后，就可以启动运行系统环境了。在开发系统中单击工具条“VIEW”按钮或快捷菜单中“切换到 View”命令后，进入组态王运行系统。如图 11.4 所示。



图 11.4 组态王运行系统

下面分别对运行系统菜单命令进行讲解。

11.2.1 画面菜单

单击“画面”菜单，弹出下拉式菜单，如图 11.5 所示。



图 11.5 画面菜单

◇ 画面\打开

选择此命令后，弹出“打开画面”对话框，如图 11.6 所示。



图 11.6 打开画面

对话框中列出当前路径下所有未打开画面的清单。用鼠标或空格键选择一个或多个

窗口后，单击“确定”按钮打开所有选中的画面，或单击“取消”按钮撤消当前操作。

◇ 画面\关闭

选择此命令后，弹出“关闭画面”对话框，如图11.7所示。

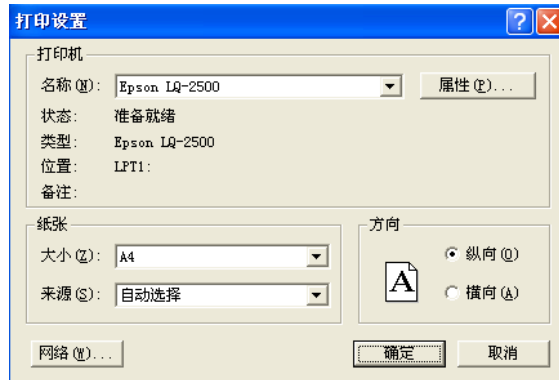


图11.7 关闭画面

对话框中列出所有已打开画面的清单。用鼠标或空格键选择一个或多个窗口后，单击“确定”按钮打开所有选中的画面，或单击“取消”按钮撤消当前操作。

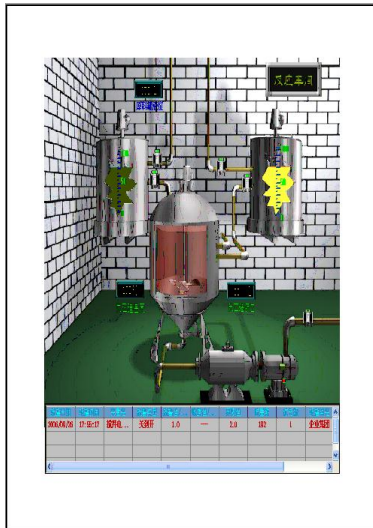
◇ 画面\打印设置

选择此命令后，弹出打印机标准设置对话框，如下图所示。用来设置画面打印时打印机的属性，如选择要使用的打印机、纸张大小、打印方向等。TouchView在运行时根据这些设置打印预览或打印画面。



例如:

分别设置打印方向为纵向和横向，组态王画面打印预览效果如下图所示。





◇ 画面\屏幕拷贝
暂时不支持此命令。

◇ 画面\退出(Alt+F4)
选择此命令后退出“组态王”运行程序。

11.2.2 特殊菜单

单击“特殊”菜单，弹出下拉式菜单，如图 11.8 所示。



图 11.8 特殊菜单

◇ **特殊\重新建立 DDE 连接**

TouchView先中断了已经建立的DDE连接，此命令用于重新建立DDE连接。

◇ **特殊\重新建立未成功的连接**

重新建立启动时未建立成功的DDE连接。已经建立成功的DDE连接不受影响。

◇ **特殊\重启报警历史记录**

此选项用于重新启动报警历史记录。在没有空闲磁盘空间时，系统自动停止报警历史记录。当发生此种情况时，将显示信息框，通知用户。为了重启报警历史记录，用户须清理出一定的磁盘空间，并选择此命令。

◇ **特殊\重启历史数据记录**

此选项用于重新启动历史数据记录。在没有空闲磁盘空间时，系统自动停止历史数据记录。当发生此种情况时，将显示信息框，通知用户。为了重启历史数

据记录，用户须清理出一定的磁盘空间，并选择此命令。

◇ **特殊\开始执行后台任务**

此选项只针对“应用程序命令语言”的“运行时”脚本，可通过选择该项控制“应用程序命令语言”的“运行时”脚本的启动。当系统检测到“应用程序命令语言”的“运行时”里有脚本时，系统运行后默认此项自动被勾选。

◇ **特殊\停止执行后台任务**

此选项只针对“应用程序命令语言”的“运行时”脚本，可通过选择该项控制“应用程序命令语言”的“运行时”脚本的停止。

◇ **特殊\登录开**

此选项用于用户进行登录。登录后，可以操作有权限设置的图形元素或对象。在TouchView运行环境下，当运行画面打开后，单击此选项，则弹出对话框，如图11.9所示。



图 11.9 用户登录对话框

- 选择已经定义了的用户名称。单击列表框右侧箭头，弹出的列表框中列出

了所有的用户名称,选择要登录的用户名称,用户配置请参见“第十六章 系统安全管理”。

- 输入选中的用户的登录密码。如果在开发环境中定义了使用软键盘,则单击该文本框时,弹出一个软键盘。也可以直接用外设键盘输入。
- 单击“确定”进行用户登录,如果用户密码错误,则会提示“登录失败”;单击“取消”则取消当前操作。在用户登录后,所有比此登录用户的访问权限级别低且在此操作员登录安全区内的图形元素或对象均变为有效。

◇ 特殊\修改口令

此选项用于修改已登录操作员的口令设置,在TouchView运行环境下,当运行画面打开后,单击此选项,则弹出对话框,如图11.10所示。



图 11.10 修改口令

- 在“旧口令”对话框中输入当前的用户密码
- 在“新口令”文本框中输入新的用户密码
- 在“校验新口令”文本框中输入新的用户密码,用于确认新密码的正确性。

- 输入完成后，单击“确定”确认口令修改；
- 单击“取消”取消当前操作。

◇ 特殊\配置用户

此选项用于重新设置用户的访问权限和口令以及安全区，当操作员的访问权限大于或等于 900 时，此选项有效。弹出“用户和安全区配置”对话框。如图 11.11 所示。



图 11.11 运行系统用户和安全区配置对话框

具体如何配置用户和安全区请参见“第十六章 系统安全管理”一章。当操作员的访问权限小于 900 时，此选项有效，会提示没有权限。

◇ 特殊\登录关

此选项用于使当前登录的用户退出登录状态，关闭有口令设置的图形元素或对象，则用户不可访问。

11.2.3 调试菜单

单击“调试”菜单，弹出下拉式菜单，如图 11.12 所示。



图 11.12 调试菜单

◇ 调试\通讯

此命令用于给出组态王与 I/O 设备通讯时的调试信息，包括通讯信息、读成功、读失败、写成功、写失败。当用户需要了解通讯信息时，选择“通讯信息”项，此时该项前面有一个符号“✓”，表示该选项有效，则组态王与 I/O 设备通讯时会在信息窗口中给出通讯信息，如图 11.13 所示。

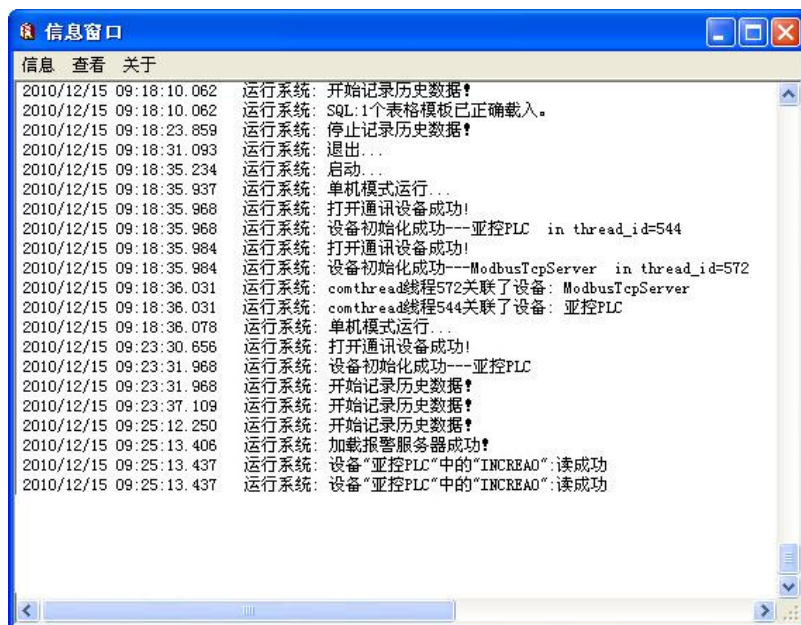


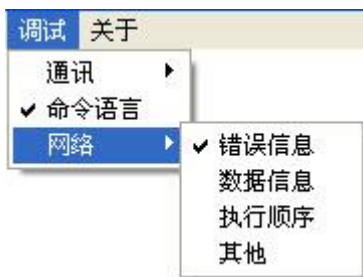
图 11.13 信息窗口通讯信息

- **通讯信息:** 在组态王信息窗口中显示/不显示组态王与设备的通讯信息。
- **读成功:** 在组态王信息窗口中显示/不显示组态王读取设备寄存器数据时成功的信息。
- **读失败:** 在组态王信息窗口中显示/不显示组态王读取设备寄存器数据时失败的信息。
- **写成功:** 在组态王信息窗口中显示/不显示组态王向设备寄存器写数据时成功的信息。
- **写失败:** 在组态王信息窗口中显示/不显示组态王向设备寄存器写数据时失败的信息。

◇ 调试\命令语言

该选项目前不起作用。

◇ 调试\网络



- **错误信息**：在组态王信息窗口中显示/不显示组态王与设备的通讯的错误信息。
- **数据信息**：在组态王信息窗口中显示/不显示组态王读取或回写的的数据信息。
- **执行顺序**：在组态王信息窗口中显示/不显示组态王的执行顺序。
- **其它**：在组态王信息窗口中显示/不显示组态王其它的信息。

以上选项只有选中时（菜单项有“✓”符号）有效。

11.2.4 导航菜单

单击“导航”菜单，弹出下拉式菜单，如图 11.14 所示。



图 11.14 导航菜单

◇ 导航\导航图

该选项用于导航图的显示或关闭。系统运行后，默认不显示导航图。选中“导航图”命令后在画面的右上方会出现矩形显示小窗口，该窗口就是导航画面，如图 11.15 所示。



图 11.15 运行系统导航图

有三种方法可以显示或关闭导航图。

- 1、使用上面提到的使用导航菜单。
- 2、使用鼠标右键点击运行系统画面，弹出快捷菜单，如图 11.16 所示，点击“导航图”命令也可显示或关闭导航图。

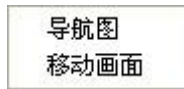


图 11.16 运行系统快捷菜单

3、使用导航图显示控制函数 EnableNavigate(is_enable)。函数参数说明如下：
is_enable: 整型,取值范围:0 或 1。is_enable 为 1 时,显示导航图;is_enable 为 0 时, 关闭导航图。使用 EnableNavigate 函数关闭导航图后, 除非使用该函数来显示导航图, 否则无法显示导航图。

在导航图中显示的始终是鼠标点击获得焦点的画面。运行画面显示窗口在整个画面中的位置, 在导航图中为一个标志矩形。画面中的图素在导航图中为缩小的图素, 报警窗口、报表、组态王控件、标准 ACTIVEX 控件不是图素, 是各自的标识符。导航图的大小是固定的, 当画面实际大小的长宽比例与导航图比例不一致时, 靠左或上为有效显示区域, 在导航图内鼠标的移动范围限制在有效区域内。按住鼠标左键点中导航图上面灰色标题条可以拖动导航图, 放置在屏幕任意的位置。

使用鼠标可以进行画面和导航图的互动操作。

- 1、 启动导航图时, 其内容为当前编辑的画面。
- 2、 当运行画面切换时, 如导航图在显示状态, 其内容也随之改变。
- 3、 当画面滚动时, 导航图中标志画面显示内容的矩形随之移动。
- 4、 当在导航图中鼠标单击指定位置时, 可将当前编辑画面滚动到以单击处坐标为中心位置。导航图中标志当前显示位置的矩形也随之移动, 但大小不变。

5、 当在导航图中单击位置在标志矩形内部时，可拖动鼠标到指定位置，放开鼠标后当前编辑画面自动滚动到相应位置。

6、 当画面没有滚动条时，显示导航图操作将不起作用。

◇ 导航\移动画面

该选项用于运行系统画面的移动。使用鼠标右键点击运行系统画面，弹出快捷菜单，如图 11.16 所示，点击“移动画面导航图”命令也可实现移动画面功能。选中此命令后该项前面有一个“✓”符号，同时鼠标变成小手的形状，按住鼠标左键移动鼠标可移动画面，但此状态下鼠标不能获得焦点。再次点击该命令则取消移动画面，该项前面“✓”符号消失。或是右键点击鼠标取消移动画面状态。

11.2.5 关于菜单

单击“关于”菜单，弹出下拉式菜单，如图 11.17 所示。



图 11.17 关于菜单

此菜单命令项用于显示“组态王”的版权信息和系统内存信息，对话框如图 11.18 所示。



图 11.18 组态王关于信息

第十二章 组态王 7.0 信息窗口

- ☞ 介绍组态王 7.0 信息窗口的功能
- ☞ 教您如何使用组态王 7.0 信息窗口

12.1 如何从信息窗口中获取信息

“组态王 7.0 信息窗口”是一个独立的 Windows 应用程序，用来记录、显示组态王 7.0 开发和运行系统在运行状态时的信息。信息窗口中显示的信息可以作为一个文件存于指定的目录中或是用打印机打印出来，供用户查阅。当工程浏览器、TouchView 等启动时，会自动启动信息窗口。一般情况下启动组态王 7.0 系统后，在信息窗口中可以显示的信息有：

- ✧ “组态王 7.0”系统的启动、关闭、运行模式；
- ✧ 历史记录的启动、关闭；
- ✧ I/O 设备的启动、关闭；
- ✧ 网络连接的状态；
- ✧ 与设备连接的状态；
- ✧ 命令语言中函数未执行成功的出错信息。
- ✧ 如果用户想要查看与下位设备通讯的信息，可以选择运行系统“调试”菜单下的“读

成功”、“读失败”、“写成功”、“写失败”等项，则 I/O 变量读取设备上的数据是否成功的信息也会在信息窗口中显示出来。组态王 7.0 的信息窗口如图 12.1 所示：

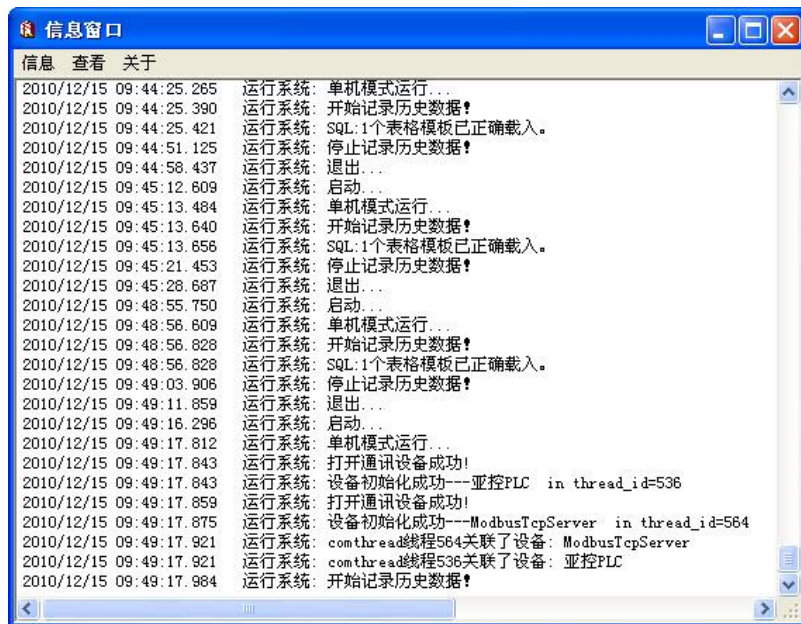


图 12.1 组态王 7.0 信息窗口

12.2 如何保存信息窗口中的信息

12.2.1 设置保存路径

用户可以将信息窗口中的信息以*.kv1 文件的形式保存到硬盘中，以供查阅。单击“信息”菜单下“设置存储路径”命令，弹出“设置保存路径”对话框，如图 12.2 所示：



图 12.2 信息窗口_设置保存路径

如果是第一次运行信息窗口，缺省保存路径为本机的临时目录“C:\Windows\Temp\”，用户可根据需要点击“浏览...”更改保存路径。一旦用户设置了新的路径后，信息窗口会自动在该路径下生成新的信息文件，以后生成的信息文件都保存到该路径下。信息文件命名方式为“年月日时分.kv1”。如用户是在 2002 年 12 月 30 日上午 10:29 保存信息文件到指定的路径下，则信息文件名称为 0212301029.kv1。

12.2.2 设置保存参数

除设置信息文件保存路径外，还可以设置保存参数。单击“信息”菜单下“设置保存参数”命令，弹出“设置保存参数”对话框，如图 12.3 所示：



图 12.3 信息窗口_设置保存参数

- ◇ **信息保存间隔：**信息文件每隔一定时间存盘一次，时间可由用户设置，缺省设置是 600 秒。在编辑输入框中设置信息文件保存到硬盘的时间间隔，也就是设定每隔多长时间将信息文件存入硬盘。例如：在编辑框中输入数值 3，则表示每隔 3 秒钟将信息文件存入硬盘。
- ◇ **信息文件保留：**指定信息文件在硬盘上的保存时间。信息窗口自动维护用户设置的保存天数内的信息，在保存期外的信息文件会被自动删除。例如：编辑框中输入数值 10，则表示信息文件在硬盘上保留 10 天，10 天之后将被自动清掉。
- ◇ **信息文件超过：**指定信息文件在硬盘上存储文件的大小，超过用户设置的大小后自动重新创建新文件。例如：在编辑框中输入数值 10，则表示保存一个日志文件最大为 10M，如果信息文件大小超过 10M，将自动重新创建新的信息文件。

12.2.3 信息窗口的内容保存为 txt 文件

用户可以将当前信息窗口显示的内容保存为 .txt 文件。单击“信息”菜单下的“另存为”，如下图 12.4 所示。



图12.4 另存为

执行“另存为”，弹出“选择信息文件”对话框，选择文件保存的路径和输入文件名，如图 12.5 所示。



图12.5 信息窗口文件保存为.txt 文件

点击“保存”按钮，如上图所示，将在 E 盘下生成一个名为 message.txt 的文件。

12.3 如何查看历史存储信息

上面一节介绍过组态王 7.0 的开发和运行系统信息以*.kvl 文件形式保存在硬盘上，形成历史信息记录。使用组态王 7.0 信息窗口可以浏览保存过的信息文件。单击“信息”菜单下“浏览信息文件”命令，弹出选择文件对话框，如图 12.6 所示：

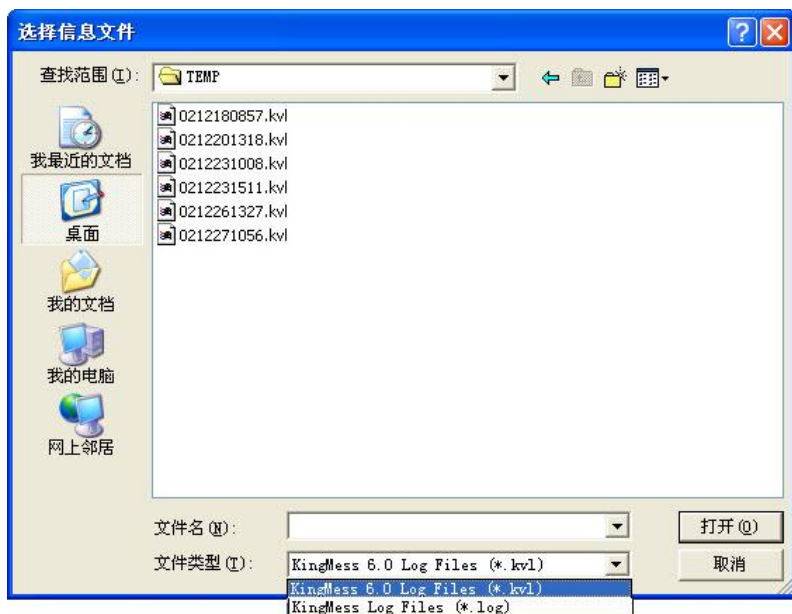


图 12.6 信息窗口_选择信息文件

 注意：

组态王 6.0 之前版本的信息文件名格式为*.log，6.0 及 6.0 以后的版本的信息

文件文件名格式为*.kv1”，例如：0212271056.kv1。用户可以选择所要浏览的信息文件，可以为老版本（6.0 之前版本）的*.log 格式或是新版本（6.0 版本）的*.kv1 格式。

选中所要浏览的信息文件，单击“打开”出现浏览信息窗口，如图 12.7 所示：

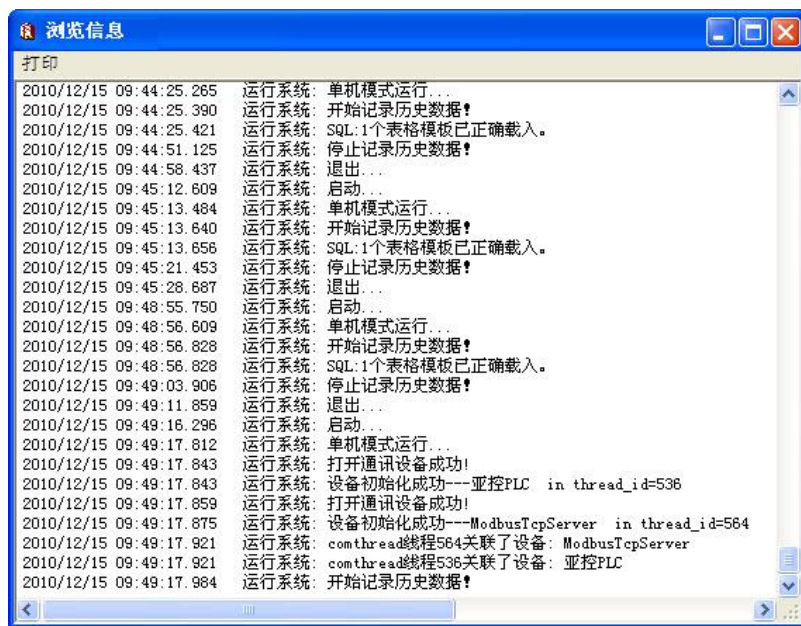


图 12.7 浏览信息

该信息窗口中的内容也可打印出来，单击“打印”菜单，出现“打印”和“打印设置”菜单命令，“打印”命令将打印信息窗口的内容；“打印设置”命令设置打印参数。具体如何打印请参见“12.4 如何打印信息窗口中的信息”一节。

12.4 如何打印信息窗口中的信息

组态王 7.0 信息窗口信息打印有两种：一种是打印当前信息窗口中的信息；另一种是打印浏览的历史信息文件。两种打印使用的方法大致相同，都是首先进行“打印设置”，然后执行“打印”，命令。下面以打印当前信息窗口中的信息为例，讲解打印功能。

12.4.1 信息\打印设置

单击“信息\打印设置”命令，弹出“设置打印参数”窗口，如图 12.8 所示：



图12.8 信息窗口_打印设置

通过“设置打印参数”窗口可对打印范围、页面设置和打印字体等参数设置。


◇ 打印范围

打印范围有四种方式选择：

- 全部（信息文件中所有信息）：打印信息窗口中显示的所有信息；
- 从…到…：输入包含在信息窗口中所显示的时间范围内的日期时间，将会打印

出所选择的时间段内的信息内容；

- 自…前：输入包含在信息窗口中所显示的时间范围内的日期时间，将会打印出所选择时间前的信息内容；
 - 自…后：输入包含在信息窗口中所显示的时间范围内的日期时间，将会打印出所选择时间后的信息内容。
-

 注意：

请正确填写打印时间范围，如果填写错误系统将会自动弹出“检查打印范围”提示框提示用户重新填写打印时间。

◇ 页面设置

对打印机及纸张进行设置。

◇ 打印字体

对打印的字体进行设置。

以上选项都设置完成后，单击“确定”按钮，完成打印设置。

12.4.2 信息\打印

单击“信息\打印”命令，弹出“打印”窗口，如图 12.9 所示：



图 12.9 信息窗口_打印

通过“打印”窗口可以对打印机的属性、打印布局和打印份数进行定义。单击“打印”按钮进行打印。

打印浏览的历史信息文件方法同上。

12.5 信息窗口其他菜单的使用

12.5.1 查看菜单

组态王 7.0 信息窗口内容可以修改显示的字体，单击组态王 7.0 信息窗口“查看\显示字体”命令，如图 12.10 所示：

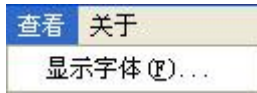


图12.10 显示字体

点击“显示字体”，弹出“字体”对话框，如图 12.11 所示：



图12.11 字体设置

在此对话框中用户可以选择信息窗口文本的各种字体、字体样式、字的大小、颜色等。

12.5.2 关于菜单

单击“关于”菜单中的“关于信息窗口”菜单，弹出组态王 7.0 信息窗口的有关信息，如图 12.12 所示：



图12.12 信息窗口

12.5.3 系统菜单

系统菜单在组态王 7.0 信息窗口左上角处，其位置如图 12.13 所示：

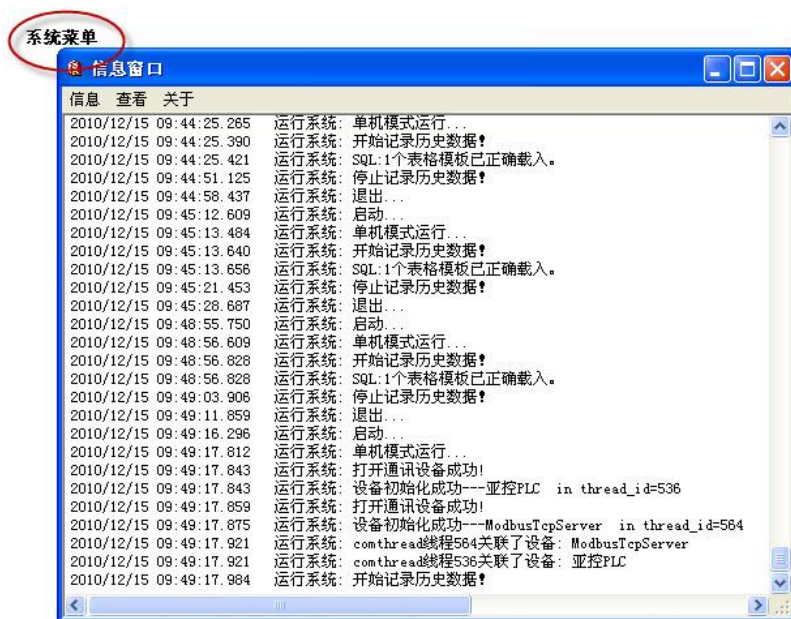


图12.13 信息窗口_系统菜单

右键单击组态王 7.0 信息窗口左上角的系统菜单，弹出的系统下拉式菜单如图 12.14 所示：



图12.14 信息窗口系统菜单

◇ 菜单命令“总在最前”的使用

此菜单命令选中有效时，则所有其它应用程序窗口总被组态王 7.0 信息窗口覆盖。如果想取消“总在最前”，则重新打开系统菜单，如图 12.15 所示：



图 12.15

单击“取消最前”菜单命令，则其它应用程序窗口可以覆盖组态王 7.0 信息窗口。当组态王 7.0 信息窗口最小化变为 WINDOWS 操作系统“开始”菜单条上的小图标时，

如图 12.16 所示：

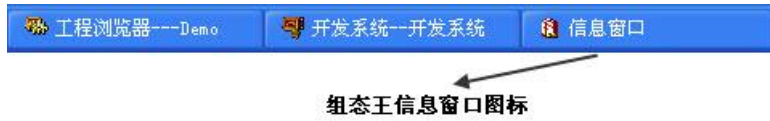


图12.16 信息窗口图标

用右键单击组态王 7.0 信息窗口图标，则也弹出系统菜单如图 12.17 所示：



图 12.17

第十三章 图库

- ☞ 介绍组态王图库的相关知识
- ☞ 教您如何使用图库管理器
- ☞ 教您如何添加图库精灵

13.1 图库概述

图库是指组态王中提供的已制作成型的图素组合。图库中的每个成员称为“图库精灵”。

使用图库开发工程界面至少有三方面的好处：一是降低了工程人员设计界面的难度，使他们能更加集中精力于维护数据库和增强软件内部的逻辑控制，缩短开发周期；二是用图库开发的软件将具有统一的外观，方便工程人员学习和掌握；最后，利用图库的开放性，工程人员可以生成自己的图库元素，“一次构造，随处使用”，节省了工程人员投资。

组态王为了便于用户更好地使用图库，提供图库管理器，图库管理器集成了图库管理的操作，在统一的界面上，完成“新建图库”，“更改图库名称”“加载用户开发的精灵”“删除图库精灵”如图 13.1 所示。



图 13.1 图库管理器

13.2 认识图库精灵

图库中的元素称为“图库精灵”。之所以称为“精灵”，是因为它们具有自己的“生命”。图库精灵在外观上类似于组合图素，但内嵌了丰富的动画连接和逻辑控制，工程人员只需把它放在画面上，做少量的文字修改，就能动态控制图形的外观，同时能完成复杂的功能。

用户可以根据自己工程的需要，将一些需要重复使用的复杂图形做成图库精灵，加入到图库管理器中。组态王提供两种方式供用户自制图库。一种是编程序方式，即用户利用亚控公司提供的图库开发包，自己利用 VC 开发工具和组态王开发系统中生成的精灵描述文本制作，生成*.dll 文件。关于该种方式，详见亚控公司提供的图库开发包。另一种

是利用组态王开发系统中建立动画连接并合成图素的方式直接创建图库精灵。本章将对第二种方式做详细说明。

 例如：

画面上需要一个按钮，代表一个开关，开关打开时按钮为绿色，开关关闭后变为红色，并且可以定义按钮为“置位”开关、“复位”开关或“切换”开关。如果没有图库，首先要绘制一个绿色按钮和一个红色按钮，用一个变量和它们连接，设置隐藏属性，最后把它们叠在一起——把这些复杂的步骤合在一起，这就是“按钮精灵”。利用组态王定义好的“按钮精灵”，工程人员只要把“按钮精灵”从图库拷贝到画面上，它就具有了“打开为绿色，关闭为红色”，也可以根据用户具体需求改变颜色，并且可以设置开关类型的功能。



图 13.2 图库精灵的组成

图库中的几乎每个精灵都有类似的已经定义的动画连接，所以使用图库精灵将极大地提高设计界面的效率。例如使用第一种方式即编制程序制作的图库精灵具有自动控制图形外观的向导功能。双击该类图形，弹出类似如下的向导界面：



图 13.3 编程序图库精灵的动画连接向导

组态王提供的系统图库中的所有精灵均是用这种方式制作的。

利用第二种方式制作的图库精灵同样具有属性定义界面，双击该类图形，弹出类似如下的向导界面：



图 13.4 动画连接方式创建的图库精灵的属性

工程人员可以在画面上任意地缩放图库精灵。

13.3 如何创建图库精灵

关于第一种编制程序创建图库精灵的方式，请见亚控公司提供的图库开发包。本节详细讲述第二种创建图库的方式，即为图形建立动画连接并合成图素。下面以一开关按钮的制作为例，教您如何创建图库精灵：

- 1、在画面上创建两个按钮，一个按钮用字符串替换使文本显示为“开”，另一为“关”。
- 2、在数据词典中定义一内存离散变量，如：0nOff
- 3、添加动画连接。

双击“开”按钮，弹出“动画连接”对话框，选择“隐含连接”，键入：0nOff==1，如图 13.5 所示。



图 13.5 建立隐含动画连接

写入正确的条件表达式后，单击“确定”，回到“动画连接”对话框，再选择“弹起时连接”，键入：0n0ff=0；确定，如图 13.6 所示。

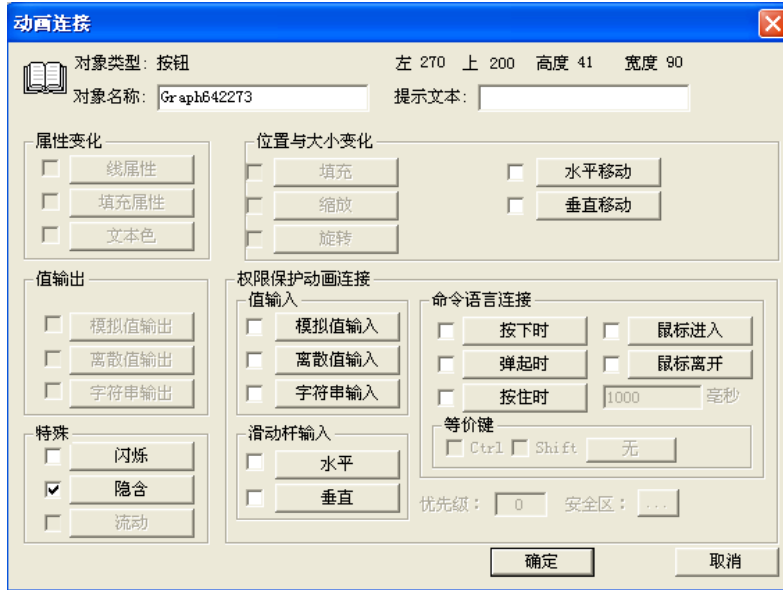


图 13.6 建立按钮动画连接

确定即可。同样，在“开”按钮的“隐舍连接”属性对话框中键入：0n0ff=0；在“弹起时连接”属性对话框中键入：0n0ff=1；确定即可，此时您已基本大功告成了。

4、组合图素单元

将两个按钮放在一起，“关”按钮在上，“开”按钮在下，选中两个按钮，选择工具箱中的“合成单元”图标，如图 13.7 所示。



图 13.7 合成单元

5、按上述步骤，您已创建了一个图库精灵，选择菜单图库/创建图库精灵，弹出“输入新的图库图素名称”，如图 13.8 所示。



图 13.8 输入加入图库的图素名

6、输入精灵名称，按确定后，弹出图库管理器，光标在图库管理器的左边确定该精灵要放的图库下，在管理器右边单击后即可，如把变色按钮放在自己创建的专用图库下，

如图 13.9 所示。

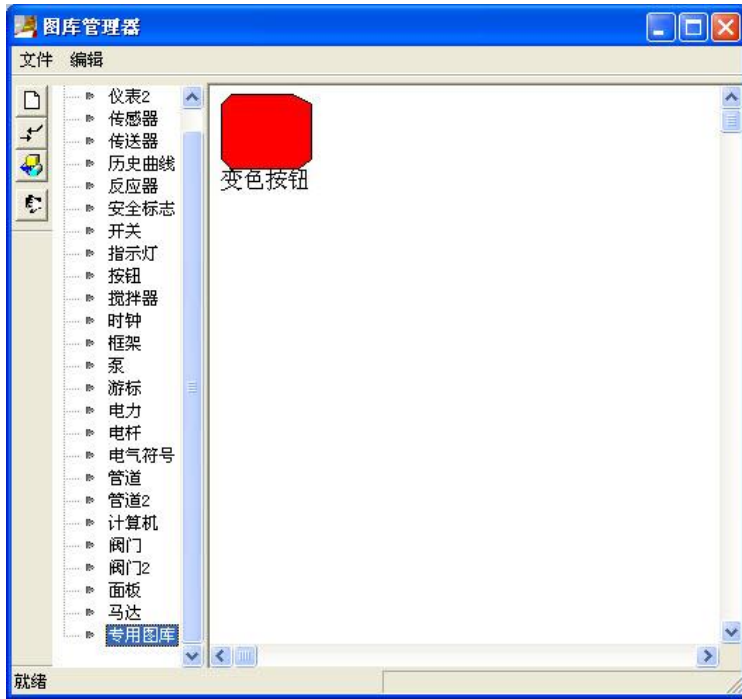


图 13.9 加入图库中的自定义图素

13.4 如何使用图库精灵

在图库管理器中选择需要的精灵。如果在开发过程中图库管理器被隐藏，请选择菜单“图库/打开图库”或按 F2 键激活图库管理器。

◇ 在画面上放置图库精灵

选择图库精灵并放到画面上是很容易的。在图库管理器窗口内用鼠标左键双击所需要的精灵（如果图库窗口不可见，请按<F2>键激活它），鼠标变成直角形。移动鼠标到

画面上适当位置，单击左键，图库精灵就复制到画面上了。可以任意移动、缩放精灵，如同处理一个单元一样。

◇ 修改图库精灵

- 对于第一种方式——编制程序制作的图库精灵具有个性化外观，双击画面上的图库精灵，将弹出改变图形外观和定义动画连接的“属性向导”对话框。对话框中包含了图库精灵的外观修改、动作、操作权限、与动作连接的变量等各项设置，对于不同的图库精灵，具有不同的属性向导界面。用户只需要输入变量名，合理调整各项设置，就可以设计出符合自己使用要求的个性化图形。如图 13.10 所示。



图 13.10 编制程序的图素——按钮

在属性界面向导中，“变量名”一项要求输入工程人员实际使用的变量名即可，该变量必须是已经在数据库中定义过的。为减少文字输入量，可单击“？”按

钮，在弹出的“变量选择”对话框中选择所需的变量名。需要注意，“变量名”使用的变量必须符合图库精灵已经定义好的变量类型。

- 对于第二种方式——直接通过动画连接并合成图素的方式制作的图库精灵同样具有可修改的属性界面。双击画面上的图库精灵，将弹出动画连接的“内容替换”对话框。对话框中记录了图库精灵的所有动画连接和连接中使用的变量。单击“变量名”，将在对话框中显示精灵使用到的所有变量；单击“动画连接”就可以看到动画连接的内容。



图 13.11 建立动画连接的图素——开关

一般情况下，该类图库精灵使用的变量名都是示意性的，不一定适合工程人员的需要，修改变量名请单击按钮“变量名”，然后在对话框中双击需要修改的变量名，则弹出“替换变量名”对话框：



图 13.12 替换变量名对话框

在对话框中输入工程人员实际使用的变量名即可，该变量必须是已经在数据库中定义过的。为减少文字输入量，可单击“？”按钮，在弹出的“变量选择”对话框中选择所需的变量名。需要注意，新变量和图库精灵原来使用的变量必须是同一类型，否则系统提示：



图 13.13 错误提示

修改完成后，图库精灵所有的动画连接中的变量名都已更改了。

工程人员也可以根据自己的需要修改任一动画连接。在“内容替换”对话框中单击按钮“动画连接”，然后在对话框中双击需要修改的栏目，弹出“动画连接设置”对话框：

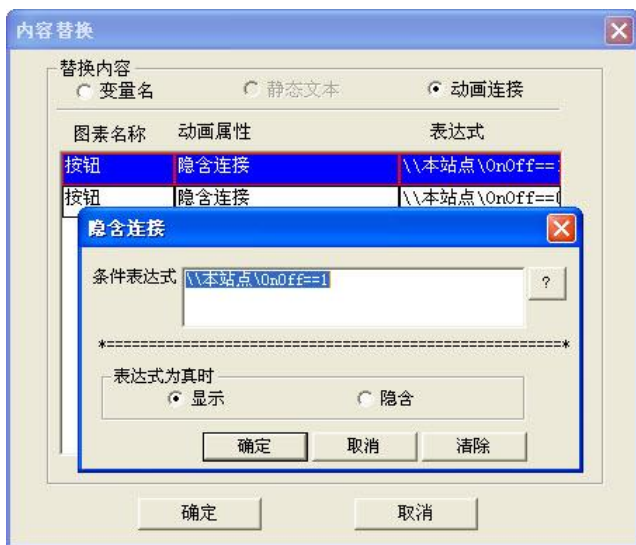


图 13.14 动画连接

图库精灵的动画连接属性的设置方法与普通图素的动画连接一样。如果组成图库精灵的图素中有静态文本，“内容替换”对话框中按钮“静态文本”加亮。单击此按钮，将在对话框中显示图库精灵中所有的静态文本。修改方法与其他图素相同。修改完成后，单击“内容替换”对话框的“确定”按钮以修改图库精灵的动画连接，或单击“取消”按钮以取消修改。

13.5 如何管理图库

图库的管理是依靠组态王提供的图库管理器完成的。图库管理器集成了图库管理的操作，在统一的界面上完成“新建图库”、“更改图库名称”、“加载用户开发的精灵”、“删除图库精灵”。

如果在开发过程中图库管理器被隐藏，请选择菜单“图库/打开图库”或按 F2 键激活

图库管理器。

在画面菜单上选择“打开图库”弹出“图库管理器” 如下图 13.15 所示。



图 13.15 图库管理器

13.5.1 图库管理器菜单条

通过弹出菜单方式管理图库。

◇ 文件菜单

- 文件\关闭

在图库管理器中单击文件菜单，如下图 13.16 所示。



图 13.16 图库管理器文件菜单

用于关闭图库管理器。如果在开发过程中图库管理器被隐藏，请选择菜单“图库 /打开图库”或按 F2 键激活图库管理器

◇ **编辑菜单**

在图库管理器中单击编辑菜单，如图 13.17 所示。



图 13.17 图库管理器编辑菜单

● **编辑\创建新图库**

单击“创建新图库”，弹出“定义新图库”对话框，如图 13.18 所示。

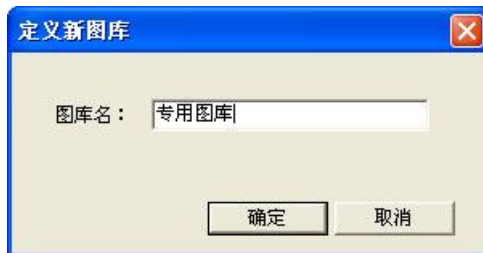


图 13.18 定义新图库

在对话框中输入名称，图库名称不超过 8 个字符（四个汉字），确定后，图库名显示在图库管理器的左边的树形中，如图 13.19 所示。

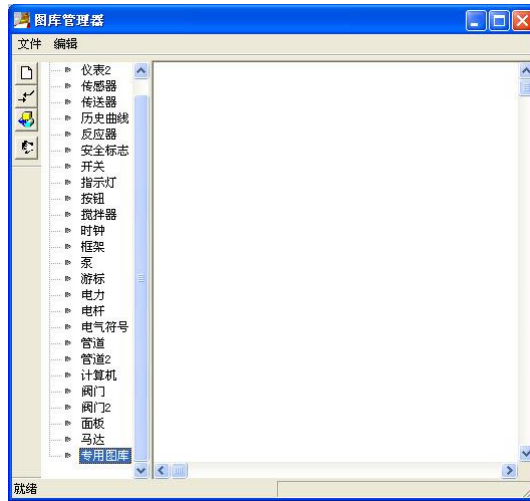


图 13.19 定义好的新图库 - “专用图库”

- 编辑\更改图库名称

在左侧选择图库名称后，单击“更改图库名称”，弹出对话框，如图 13.20 所示。

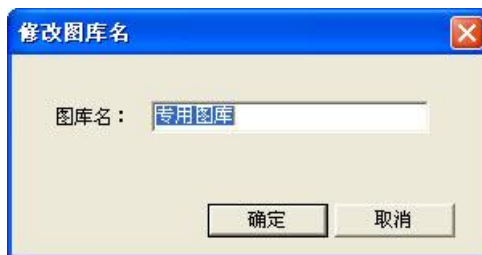


图 13.20 修改图库名

输入新名称即可（注：名称不允许相同）如把“专用图库”改名成“特别图库”如图 13.21 所示。



图 13.21 修改图库名后的图库管理器

- 编辑\删除图库精灵

选中要删除的精灵，单击“删除图库精灵”，弹出对话框。



图 13.22 确认是否删除图素精灵


- 编辑\加载用户图库精灵

当用户使用图库开发包开发出专用图形时（文件格式为*.dll），通过该项选择将自己编制的图形加入到组态王的图库管理器中来。单击“加载用户开发的精灵”，弹出对话框。



图 13.23 加载图库精灵

该菜单只对第一种方式编制的图库精灵有效。

图库文件名：要加载的图库名，单击  按钮，可以选择系统中的图库精灵，系统默认路径是组态王的当前路径下的 Dynamos 路径，用户自己开发的图库精灵文件 (*.dll)，均放在该路径下。单击按钮时会显示 Dynamos 路径下的所有 dll 文件，用户选择其中的一个。

要加载的精灵序号：一个图库程序中可以包含多个图库精灵，这些精灵都有一个序号，从 0 开始

“加载库中全部精灵” 选择该选项，可以一次加载该图库程序中的全部精灵。

例如加载用户自己定义的泵：

- 1) 在对话框中单击  按钮，弹出对话框，如图 13.24 所示。

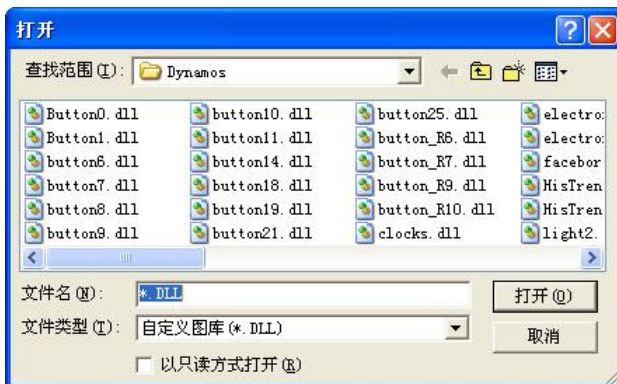


图 13.24 Dynamos 路径下存放的所有图库程序

- 2) 选择要加载的图库名，确定加载索引号，如图 13.25 所示。



图 13.25 加载泵.dll 中的第 5 个图库精灵

- 3) 按“确定”后，鼠标变成“L”型，选择一个图库后，在图库管理器右侧精灵窗口内点击鼠标即可。

13.5.2 图库管理器工具条

在图库管理器的左上方是图库管理器工具条，即通过快捷图标方式管理图库管理器。如图：



创建新图库：与工程管理器中菜单“编辑\创建新图库”效果相同



更改图库名称：与工程管理器中菜单“编辑\更改图库名称”效果相同



加载用户图库精灵：与工程管理器中菜单“编辑\加载用户图库精灵”效果相同



删除图库精灵：与工程管理器中菜单“编辑\删除图库精灵”效果相同

13.5.3 图库显示区

显示图库管理器中所有的图库。

13.5.4 精灵显示区

显示图库里的精灵。

13.6 如何将图库精灵转换成普通图素

如果要对组成图库精灵的图素作调整，如：需要改变图库精灵的颜色属性，首先把图库精灵转换成普通图素。具体操作是：

- 1、选取某图库精灵（精灵周围出现 8 个小矩形），拖动到画面上。如“特别图库”（用户自己定义的）图库下的“指示灯_7”。
- 2、选中该图库精灵，在组态王开发系统菜单中选择“图库/转换成普通图素”。如图：



- 3、在工具箱中选择“分裂单元”图标，选中“指示灯_7”，如图：



- 4、移动鼠标，将“指示灯_7”拆分为：



- 5、若想继续拆分，选中图素，在工具箱中选择“分裂组合图素”图标，继续拆分，然后工程人员就可以对这些图素做任意的修改。

注意:

对于编程方式创建的图库精灵，在转换成普通图素后，其各个图素含有的动画连接将不再存在；利用组态王的图素创建的图库精灵在被转换为普通图素后，其各个图素含有的动画连接将被保留下来。

13.7 关于图库开发包

为了方便用户自己开发适用的图库，亚控公司为用户提供了“组态王开发工具”，其中之一为“图库开发包”。利用该开发包用户可以自己通过编程制作动态连接库类型的图库精灵。关于组态王开发工具具体情况可以与亚控公司及个分支机构的技术支持或销售人员联系。

利用图库开发包开发图库精灵时，需要用程序语言描述图素外观及其属性。为了方便用户，组态王在画面开发系统中提供了一个“精灵描述文本”的工具。

精灵描述文本是指对利用组态王的绘图工具绘制出的图素进行描述的文本文件，其内容包括各个图素的线形、颜色、动画连接、操作权限、命令语言等信息，是一段类似 C 程序的文本，用户可以利用该段描述文本，用 C 等编程语言来制作自己的图库精灵。操作方法为：

- 1) 绘制图素，并做必要的动画连接等。如图 13.26 所示。创建一个小的矩形，然后创建一个椭圆，定义过渡色类型。定义椭圆的动画连接为旋转，连接的变量为整型变量“旋转”。

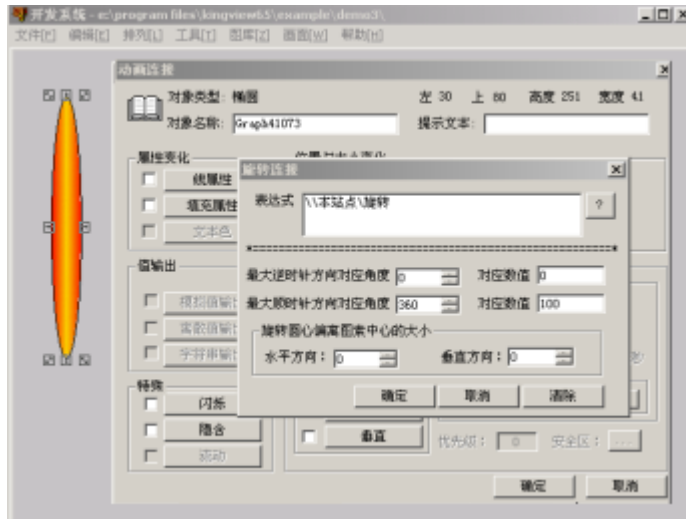


图 13.26 创建图素及动画连接

2) 选中要进行描述的图素，可以为多个，选择菜单“图库\生成精灵描述文本”命令，则系统自动并弹出生成精灵的描述文本。本例中全部选中创建的图素，选择“生成精灵描述文本”命令。

下面是本例中生成的描述文本：

```
/* 绘制矩形框 */  
  
{  
  
DRAWRECT graphic;  
  
graphic.x1 = 0 * fact;  
  
graphic.x2 = 7 * fact;  
  
graphic.y1 = 56 * fact;
```

```
    graphic.y2 = 64 * fact;

    graphic.brushColor = 255;

    graphic.brushStyle = 0;

    graphic.brushHatch = 0;

    graphic.fillStyle = 21;

    graphic.fromColor = 255;

    graphic.toColor = 10079487;

    graphic.penColor = 65535;

    graphic.penStyle = 0;

    graphic.penWidth = 1;

    KVDDrawRoundRect( &graphic);

}

/* 绘制椭圆 */

{

    DRAWRECT graphic;

    graphic.x1 = 0 * fact;

    graphic.x2 = 7 * fact;

    graphic.y1 = 0 * fact;

    graphic.y2 = 61 * fact;

    graphic.brushColor = 255;
```

```
    graphic.brushStyle = 0;
    graphic.brushHatch = 0;
    graphic.fillStyle = 3;
    graphic.fromColor = 255;
    graphic.toColor = 10079487;
    graphic.penColor = 65535;
    graphic.penStyle = 0;
    graphic.penWidth = 1;
    KVDrawEllipse( &graphic);
}
/* 设置旋转动画连接 */
{
    LINKROTATE links;
    links.minVal = 0.000000;
    links.maxVal = 100.000000;
    links.minAngle = 0;
    links.maxAngle = 360;
    links.CenterPt.x = 0;
    links.CenterPt.y = 96;
    char Express[14];
```



```
strcpy(Express, "\\本站点\旋转");  
  
links.pExpress = Express;  
  
KVLinkRotate( &links);  
  
}
```

如果图素有安全限制，定义图素的动画连接时定义了优先级或安全区，在描述文本中都可以体现出来。

图库精灵描述文本为用户自制图库精灵提供了很大的方便和帮助，而且也避免了用户编程时出现错误。其详细使用方法见“组态王图库开发包”。

第十四章 控件

- ☞ 介绍各种控件的相关知识
- ☞ 介绍组态王 7.0 内置控件的设置及使用
- ☞ 介绍 Active X 控件的设置及使用
- ☞ 介绍组态王 7.0 中提供的 Active X 控件的使用方法
- ☞ 介绍温控曲线控件的设置及使用

14.1 控件简介

14.1.1 什么是控件

控件实际上是可重用对象,用来执行专门的任务。每个控件实质上都是一个微型程序,但不是一个独立的应用程序,通过控件的属性、方法等控制控件的外观和行为,接受输入并提供输出。例如,Windows 操作系统中的组合列表框就是一个控件,通过设置属性可以决定组合列表框的大小,要显示文本的字体类型,以及显示的颜色。组态王 7.0 的控件(如棒图、温控曲线、X-Y 轴曲线)就是一种微型程序,它们能提供各种属性和丰富的命令语言函数用来完成各种特定的功能。

14.1.2 控件的功能

控件在外观上类似于组合图素, 工程人员只需把它放在画面上, 然后配置控件的属性, 进行相应的函数连接, 控件就能完成复杂的功能。

当所实现的功能由主程序完成时需要制作很复杂的命令语言, 或根本无法完成时, 可以采用控件。主程序只需要向控件提供输入, 而剩下的复杂工作由控件去完成, 主程序无需理睬其过程, 只要控件提供所需要的结果输出即可。另外, 控件的可重用性也提供了方便。比如画面上需要多个二维条图, 用以表示不同变量的变化情况, 如果没有棒图控件, 则首先要利用工具箱绘制多个长方形框, 然后将它们分别进行填充连接, 每一个变量对应一个长方形框, 最后把这些复杂的步骤合在一起, 才能完成棒图控件的功能。而直接利用棒图控件, 工程人员只要把棒图控件拷贝到画面上, 对它进行相应的属性设置和命令语言函数的连接, 就可实现用二维条图或三维条图来显示多个不同变量的变化情况。


总之, 使用控件将极大地提高工程人员工程开发和工程运行的效率。

14.1.3 组态王 7.0 支持的控件

组态王 7.0 本身提供很多内置控件, 如列表框、选项按钮、棒图、温控曲线、视频控件等, 这些控件只能通过组态王 7.0 主程序来调用, 其它程序无法使用, 这些控件的使用主要是通过组态王 7.0 相应控件函数或与之连接的变量实现的, 其使用方法请参见 14.2 节。

随着 Active X 技术的应用, Active X 控件也普遍被使用。组态王 7.0 支持符合其数据类型 Active X 标准控件。这些控件包括 Microsoft Windows 标准控件和任何用户制作的标准 Active X 控件。这些控件在组态王 7.0 中被称为“通用控件”, 本手册及组态王

7.0 程序中但凡提到“通用控件”，既是指 Active X 控件。Active X 控件在组态王 7.0 中的使用方法请参见 14.3 节。

 注意：

在运行系统中使用控件的函数、属性、方法等时，应该打开含有控件的画面（不一定是当前画面），否则会造成操作失败，这时，信息窗口中应该有相应的提示。

14.2 组态王 7.0 内置控件

组态王 7.0 内置控件是组态王 7.0 提供的、只能在组态王 7.0 程序内使用的控件。它能实现控件的功能，组态王 7.0 通过内置的控件函数和连接的变量来操作、控制控件，从控件获得输出结果。其它用户程序无法调用组态王 7.0 内置控件。这些控件包括：棒图控件、温控曲线、X—Y 曲线、列表框、选项按钮、文本框、超级文本框、AVI 动画播放控件、视频控件、开放式数据库查询控件、历史曲线控件等。在组态王 7.0 中加载内置控件，可以单击工具箱中的“插入控件”按钮，如图 14.1 所示。



图 14.1 工具箱—插入控件按钮

或选择画面开发系统中的“编辑/插入控件”菜单。系统弹出“创建控件”对话框，如图 14.2 所示。



图 14.2 创建控件对话框

对话框左侧的“种类”列表中列举了内置控件的类型，选择每一项，在右侧的内容显示区中可以看到该类中包含的控件。选择控件图标，单击按钮“创建”，则创建控件；单击“取消”按钮，则取消创建。

14.2.1 立体棒图控件

使用棒图控件，需先在画面上创建控件。单击工具箱中的“插入控件”按钮，或选择画面开发系统中的“编辑/插入控件”菜单。系统弹出“创建控件”对话框，在种类列表中选择“趋势曲线”，在右侧的内容中选择“立体棒图”图标，单击对话框上的“创建”

按钮，或直接双击“立体棒图”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如下图 14.3 所示。

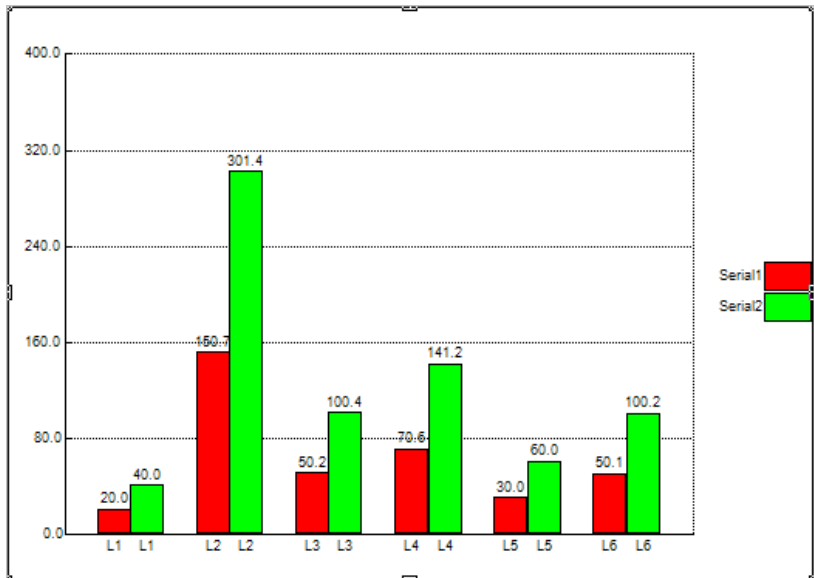


图 14.3 棒图控件

控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

棒图每一个条形图下面对应一个标签 L1、L2、L3、L4、L5、L6。这些标签分别和组态王 7.0 数据库中的变量相对应，当数据库中的变量发生变化时，则与每个标签相对应的条形图的高度也随之动态地发生变化，因此通过棒图控件可以实时地反应数据库中变量的

变化情况。另外，工程人员还可以使用三维条形图和二维饼形图进行数据的动态显示。

14.2.1.2 设置棒图控件的属性

用鼠标双击棒图控件，则弹出棒图控件属性页对话框，如图 14.4 所示：

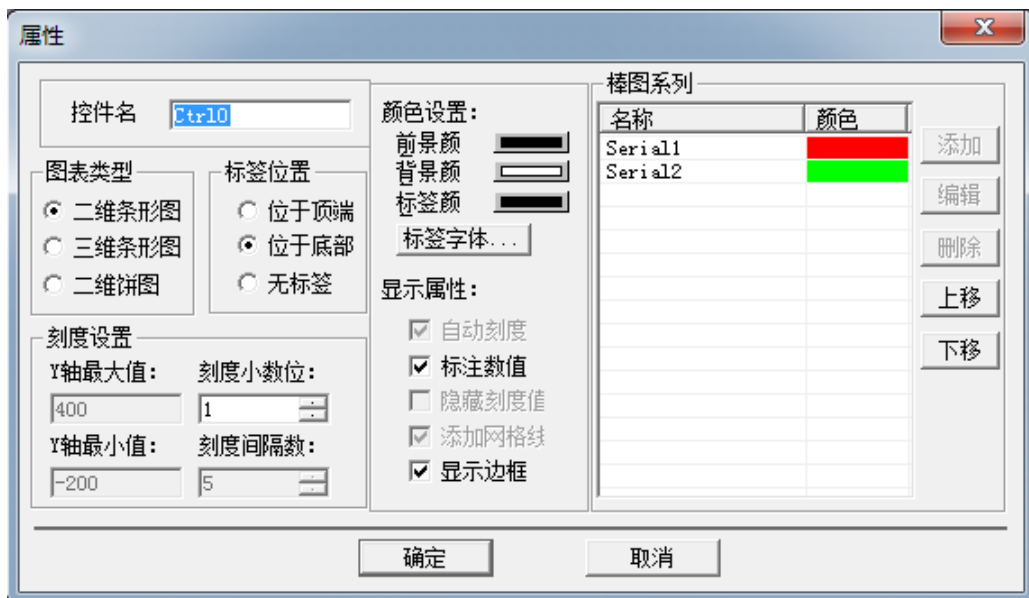
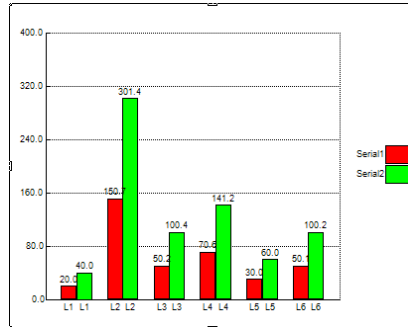


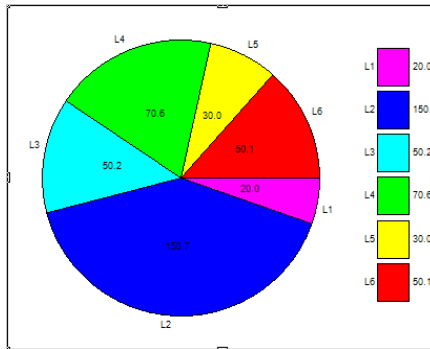
图 14.4 棒图控件属性设置

此属性页用于设置棒图控件的控件名称、图表类型、标签位置、颜色设置、刻度设置、字体型号、显示属性等各种属性，分别介绍如下：

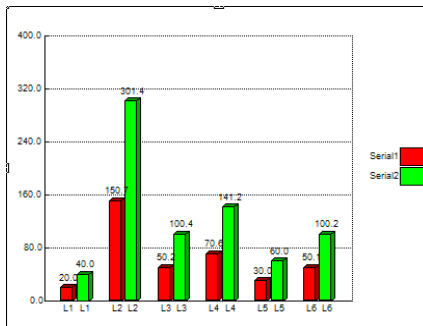
- ◇ **图表类型：**提供二维条形图、三维条形图和二维饼形图三种类型，三种类型显示效果如下图例所示。



二维条形图



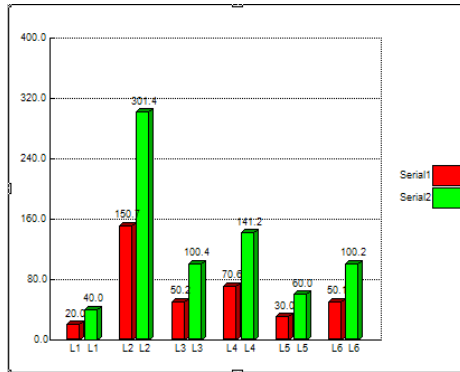
二维饼图



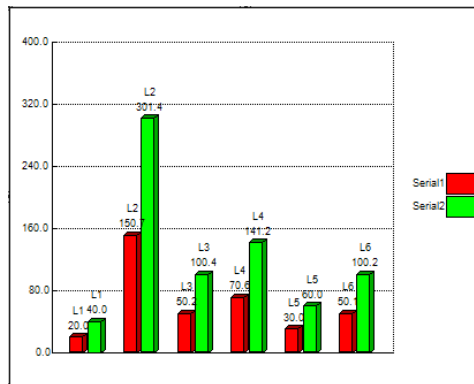
三维条形图

◇ **标签位置：**用于指定变量标签放置的位置，提供**位于顶端**、**位于底部**、**无标签**三种类型，对于不同的图表类型，位于顶端、位于底部两种类型的含义有所不同：

当工程人员将图表类型设置为二维条形图、三维条形图时，则位于顶端是指变量标签处于条形图的顶部，位于底部是指变量标签处于条形图和横坐标的下面。如下图所示。

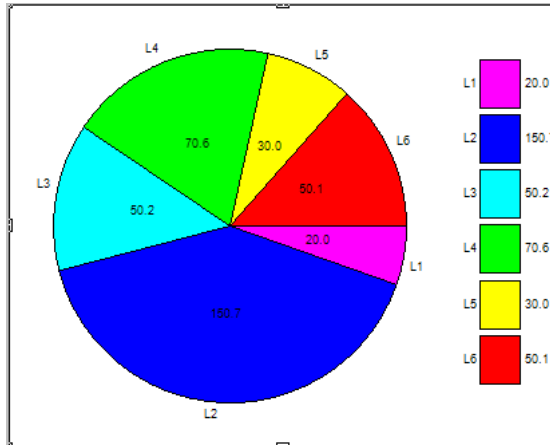


变量标签 L1、L2、L3、L4、L5、L6 位于底部

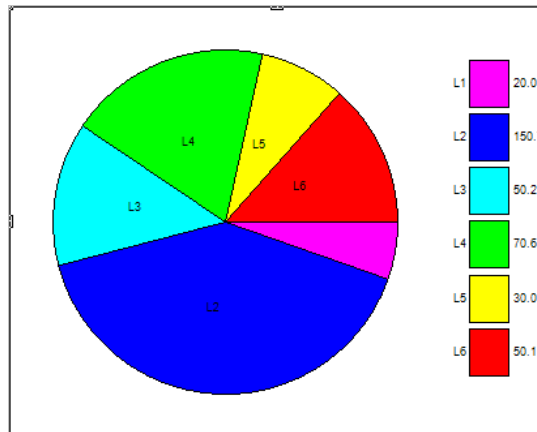


变量标签 L1、L2、L3、L4、L5、L6 位于顶部

当工程人员将图表类型设置为二维饼形图时，则位于顶端是指标签处于饼形图的外部，位于底部是指标签处于饼形图的内部。如下图所示。



标签处于饼形图的外部



标签处于饼形图的内部

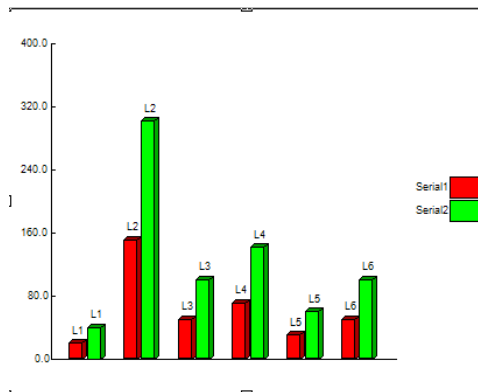
◇ **前景颜色：**此按钮用于设置棒图外框、坐标轴、网格的显示颜色。单击“前景”按

钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可使用。

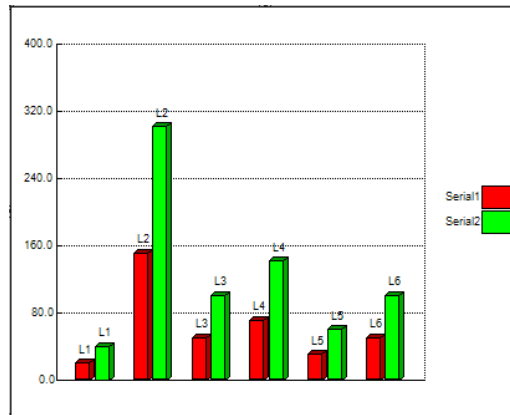
- ◇ **背景颜色：**此按钮用于设置棒图的背景显示颜色。单击“背景”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可使用。
- ◇ **标签颜色：**此按钮用于设置棒图上的所带文字的显示颜色。单击“棒图”按钮，则弹出下拉式颜色列表框供工程人员选择，有多种颜色可使用。
- ◇ **标签字体：**此按钮用于设置变量标签的字体大小、字体样式。单击“标签字体”按钮，则
弹出“字体”对话框。
- ◇ **Y 轴最大值：**用于设置 Y 轴的最大坐标值。当“显示属性中”的“自动刻度”不选择时此项有效。
- ◇ **Y 轴最小值：**用于设置 Y 轴的最小坐标值。当“显示属性中”的“自动刻度”不选择时此项有效。
- ◇ **刻度小数位：**用于设置 Y 轴坐标刻度值的有效小数位。
- ◇ **刻度间隔数：**用于指定 Y 轴的最大坐标值和最小坐标值之间的等间隔数，通常默认值为 5 等份间隔。比如，如果 Y 轴的最大坐标值为 300，最小坐标值为 100，设定刻度间隔数为 20，则最小坐标值和最大坐标值之间有 20 等份，每一个等份代表的值为 10。设定的刻度间隔数不同，则每一个等份代表的值也不相同。当“显示属性中”的“自动刻度”不选择时此项有效。
- ◇ **自动刻度：**此选项用于自动/手动设置 Y 轴坐标的刻度值，当此选项有效时，此选项前面有一个对勾符号“✓”，Y 轴最大值和 Y 轴最小值编辑输入框变灰无效，则 Y

轴坐标的刻度将根据当前条形图的最大值进行自适应调整，一般 Y 轴坐标的最大刻度值取比当前最大值大的整数，即留一定余量。如果“自动刻度”选项无效时，则需要设定 Y 轴坐标刻度的最大值和 Y 轴坐标刻度的最小值。

- ◇ **标注数值：** 此选项用于显示/隐藏棒图上的标注数值。
- ◇ **隐藏刻度值：** 此选项用于显示/隐藏 Y 轴坐标的刻度值，当此选项有效时，此选项前面有一个对勾符号“✓”，同时刻度小数位和刻度间隔数编辑输入框变灰无效。
- ◇ **添加网格线：** 此选项用于添加/删除网格线，当此选项有效时，此选项前面有一个对勾符号“✓”，网格线用于标识 Y 轴坐标刻度值的大小。有网格线和无网格线的棒图如下所示。



无网格线的棒图



有网格线的棒图

- ◇ **显示边框**: 此选项用于显示/隐藏棒图的边框。
- ◇ **棒图系列**: 通过添加、删除、编辑按钮可以添加、删除或编辑一个系列。添加或编辑时可以设置系列的名称和颜色。可以通过上移、下移按钮调整系列的显示顺序。

14.2.1.3 如何使用棒图控件

设置完棒图控件的属性后，就可以准备使用该控件了。棒图控件与变量关联，以及棒图的刷新都是使用组态王 7.0 提供的棒图函数来完成的。组态王 7.0 的棒图函数有以下这些：

- ◇ **chartAdd ("ControlName", Value, "label");**

此函数为第一个系列添加数据值和标签。语法格式如下：

```
chartAdd( "ControlName" , Value, "label" );
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

Value: 设定条形图的初始值, 整形数据, 实型数据。

label: 设定条形图的标签值, 默认值=索引值 Index, Index 的取值范围是 1-16。

例如:

```
chartAdd( "XYChart", 1, "L6" );
```

此语句将在棒图控件 XYChart 中为第 1 个系列增加一个标签为 L6 的条形图, 其初始值 1。

✧ **chartAddSerial("ControlName", "SerialName" , redValue, greenValue, blueValue);**

此函数用于在指定的棒图控件中增加一个新的系列, 仅用于条形图, 不适用于饼图。

语法格式如下:

```
chartAddSerial( "ControlName", "SerialName" , redValue, greenValue,  
blueValue );
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

SerialName: 新增系列的名称。

redValue: 新增系列颜色 RGB 值的 R 值。

greenValue: 新增系列颜色 RGB 值的 G 值。

blueValue : 新增系列颜色 RGB 值的 B 值。

例如:

```
chartAddSerial( "XYChart", "Serial3" , 255, 0, 0);
```

此语句将在棒图控件 XYChart 中增加一个颜色为红色，名称为 Serial3 的系列。

◇ **chartAddSerialValue ("ControlName", nSerialIndex,Value, "Label");**

此函数为添加一个系列并为其设置数据值和标签。语法格式如下：

```
chartAddSerialValue ("ControlName", nSerialIndex,Value, "Label" );
```

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

nSerialIndex：所选系列的索引，从 0 开始计。用于饼图时，nSerialIndex 值为 1。

Value：设定条形图的初始值，整形数据，实型数据。

label：设定条形图的标签值，默认值=索引值 Index，Index 的取值范围是 1-16。

例如：

```
chartAddSerialValue ( "XYChart", 1, 1, "L6" );
```

此语句将在棒图控件 XYChart 中添加系列 1，并为其设置标签为 L6，初始值 1 的条形图。

◇ **chartClear("ControlName");**

此函数用于在指定的棒图控件中删除第一个系列的数据值和标签。语法格式如下：

```
chartClear( "ControlName" );
```

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

例如：

```
chartClear( "XYChart" );
```

此语句将在棒图控件 XYChart 中删除第一个系列的所有数据。

◇ **chartClearSerialData("ControlName", nSerialIndex)**

此函数用于清除 nSerialIndex 系列的所有数据值（包括标签）。

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

nSerialIndex：所选系列的索引，从 0 开始计。用于饼图时，nSerialIndex 值为 1。

例如：

```
chartClearSerialData ( " XYChart ", 1);
```

此语句将在棒图控件 XYChart 中清除系列 1 的所有数据。

◇ **chartClearAll("ControlName");**

此函数用于在指定的棒图控件中删除所有数据，用于条形图或者饼图。语法格式如下：

```
chartClearAll( "ControlName");
```

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

例如：

```
chartClearAll( " XYChart ");
```

此语句将在棒图控件 XYChart 中删除所有的数据。

◇ **chartSetPieColor("ControlName", nValueIndex, nColorIndex);**

此函数用于在指定的棒图控件中设置饼图的颜色。仅用于饼图。颜色由颜色索引表取得。语法格式如下：

```
chartSetPieColor("ControlName", nValueIndex, nColorIndex);
```

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

nValueIndex: 整型变量, 表示饼状图索引号, 用于设定指定的饼状图, 从 0 开始计。

nColorIndex: 颜色索引值。

索引	英文名	颜色值	中文名
1	Black		黑
2	AliceBlue	#F0F8FF	爱丽丝蓝
3	AntiqueWhite	#FAEBD7	古董白
4	Aqua	#00FFFF	水绿
5	Aquamarine	#7FFFD4	碧绿
6	Azure	#F0FFFF	天蓝
7	Beige	#F5F5DC	米色
8	Bisque	#FFE4C4	汤黄
9	Black		黑
10	BlanchedAlmond	#FFEBCD	布兰奇杏黄
11	Blue	#0000FF	蓝
12	BlueViolet	#8A2BE2	蓝紫
13	Brown	#A52A2A	棕褐
14	BurlyWood	#DEB887	实木
15	CadetBlue	#5F9EA0	军校蓝
16	Chartreuse	#7FFF00	查特酒绿
17	Chocolate	#D2691E	巧克力色
18	Coral	#FF7F50	珊瑚色
19	CornflowerBlue	#6495ED	矢车菊蓝
20	Cornsilk	#FFF8DC	玉米黄
21	Crimson	#DC143C	深红
22	Cyan	#00FFFF	青
23	DarkBlue	#00008B	暗蓝
24	DarkCyan	#008B8B	暗青
25	DarkGoldenrod	#B8860B	暗一枝黄花
26	DarkGray	#A9A9A9	暗灰
27	DarkGreen	#006400	暗绿
28	DarkKhaki	#BDB76B	暗黄褐
29	DarkMagenta	#8B008B	暗品红
30	DarkOliveGreen	#556B2F	暗橄榄绿
31	DarkOrange	#FF8C00	暗橙色
32	DarkOrchid	#9932CC	暗淡紫
33	DarkRed	#8B0000	暗红
34	DarkSalmon	#E9967A	暗鲜肉色
35	DarkSeaGreen	#8FBC8B	暗海绿
36	DarkSlateBlue	#483D8B	暗岩蓝
37	DarkSlateGray	#2F4F4F	暗岩灰

第十四章 控件

38	DarkTurquoise	#00CED1	暗青绿
39	DarkViolet	#9400D3	暗紫
40	DeepPink	#FF1493	深粉红
41	DeepSkyBlue	#00BFFF	深天蓝
42	DimGray	#696969	淡灰
43	DodgerBlue	#1E90FF	传单蓝
44	Firebrick	#B22222	耐火砖色
45	FloralWhite	#FFFAF0	花卉白
46	ForestGreen	#228B22	森林绿
47	Fuchsia	#FF00FF	灯笼海棠红
48	Gainsboro	#DCDCDC	浅灰色
49	GhostWhite	#F8F8FF	幽灵白
50	Gold	#FFD700	金黄
51	Goldenrod	#DAA520	一枝黄花
52	Gray	#808080	灰
53	Green	#008000	绿
54	GreenYellow	#ADFF2F	黄绿
55	Honeydew	#F0FFD0	蜜色
56	HotPink	#FF69B4	热粉红
57	IndianRed	#CD5C5C	印度红
58	Indigo	#4B0082	靛蓝
59	Ivory	#FFFFF0	象牙色
60	Khaki	#F0E68C	卡其黄
61	Lavender	#E6E6FA	薰衣草紫
62	LavenderBlush	#FFF0F5	紫红
63	LawnGreen	#7FC000	草坪绿
64	LemonChiffon	#FFFACD	柠檬薄绸
65	LightBlue	#ADD8E6	亮蓝
66	LightCoral	#F08080	亮珊瑚色
67	LightCyan	#E0FFFF	亮青
68	LightGoldenrodYellow	#FAFAD2	亮一枝黄
69	LightGray	#D3D3D3	亮灰
70	LightGreen	#90EE90	亮绿
71	LightPink	#FFB6C1	亮粉红
72	LightSalmon	#FFA07A	亮鲜肉色
73	LightSeaGreen	#20B2AA	亮海绿
74	LightSkyBlue	#87CEFA	亮天蓝
75	LightSlateGray	#778899	亮岩灰
76	LightSteelBlue	#B0C4DE	亮钢蓝
77	LightYellow	#FFFFE0	亮黄
78	Lime	#00FF00	浅绿
79	LimeGreen	#32CD32	浅绿

第十四章 控件

80	Linen	#FAF0E6	亚麻色
81	Magenta	#FF00FF	品红
82	Maroon	#800000	栗色
83	MediumAquamarine	#66CDAA	中碧绿
84	MediumBlue	#0000CD	中蓝
85	MediumOrchid	#BA55D3	中淡紫
86	MediumPurple	#9370DB	中紫
87	MediumSeaGreen	#3CB371	中海绿
88	MediumSlateBlue	#7B68EE	中岩蓝
89	MediumSpringGreen	#00FA9A	中春绿
90	MediumTurquoise	#48D1CC	中青绿
91	MediumVioletRed	#C71585	中紫红
92	MidnightBlue	#191970	午夜蓝
93	MintCream	#F5FFFA	薄荷乳
94	MistyRose	#FFE4E1	雾玫瑰红
95	Moccasin	#FFE4B5	鹿皮色
96	NavajoWhite	#FFDEAD	纳瓦霍白
97	Navy	#000080	海军蓝
98	OldLace	#FDF5E6	旧鞋带
99	Olive	#808000	橄榄
100	OliveDra	#6B8E23	干橄榄
101	Orange	#FFA500	橙
102	OrangeRed	#FF4500	橙红
103	Orchid	#DA70D6	兰花紫
104	PaleGoldenrod	#EEE8AA	白黄
105	PaleGreen	#98FB98	白绿
106	PaleTurquoise	#AFEEEE	白青绿
107	PaleVioletRed	#DB7093	白紫红
108	PapayaWhip	#FFD5D5	番木瓜鞭色
109	PeachPuff	#FFDAB9	桃粉
110	Peru	#CD853F	秘鲁色
111	Pink	#FFC0CB	粉红
112	Plum	#DDA0DD	梅红
113	PowderBlue	#B0E0E6	粉蓝
114	Purple	#800080	紫
115	Red	#FF0000	红
116	RosyBrown	#BC8F8F	玫瑰褐
117	RoyalBlue	#4169E1	皇家蓝
118	SaddleBrown	#8B4513	鞍褐
119	Salmon	#FA8072	鲜肉色
120	SandyBrown	#F4A460	沙褐
121	SeaGreen	#2E8B57	海绿

122	SeaShell	#FFF5EE	海贝色
123	Sienna	#A0522D	赭色
124	Silver	#C0C0C0	银白
125	SkyBlue	#87CEEB	天蓝
126	SlateBlue	#6A5ACD	岩蓝
127	SlateGray	#708090	岩灰
128	Snow	#FFFAFA	雪白
129	SpringGreen	#00FF7F	春绿
130	SteelBlue	#4682B4	钢蓝
131	Tan	#D2B48C	茶色
132	Teal	#008080	水鸭青
133	Thistle	#D8BFD8	薊色
134	Tomato	#FF6347	番茄红
135	Transparent	#FFFFFF	透明
136	Turquoise	#40E0D0	宝石绿
137	Violet	#EE82EE	紫罗兰
138	Wheat	#F5DEB3	小麦色
139	White	#FFFFFF	白
140	WhiteSmoke	#F5F5F5	烟白
141	Yellow	#FFFF00	黄
142	YellowGreen	#9ACD32	黄绿

◇ `chartSetValue("ControlName", nValueIndex, Value);`

此函数用于在指定的棒图控件中为第 1 个系列设定/修改索引值为 Index 的饼图或条形图的数据值。语法格式如下：

```
chartSetValue("ControlName", nValueIndex, Value);
```

参数：ControlName：工程人员定义的棒图控件名称，可以为中文名或英文名。

nValueIndex：选定要设置的条形图或扇区索引，从 0 开始计。

Value：设定条形图或扇区的数据，整形数据，实型数据。

例：要在画面上棒图显示变量“原料罐温度”和“反应罐温度”的值的变化。则可以按照系列步骤进行。

1) 在画面上创建棒图控件，定义控件的属性，如图 14.5 所示，棒图名称为“温度棒图”，图标类型选择“三维条形图”，其它选项为默认值。定义完成后，单击“确定”

按钮，关闭属性对话框。



图 14.5 定义棒图属性

2) 在画面上单击右键，在弹出的快捷菜单中选择“画面属性”，在弹出的画面属性对话框中选择“命令语言”按钮，单击“显示时”标签，在命令语言编辑器中，添加如下程序：

```
chartAdd(“温度棒图”， \\本站点\原料罐温度，“原料罐”);
```

```
chartAdd(“温度棒图”， \\本站点\反应罐温度，“反应罐”);
```

该段程序将在画面被打开为当前画面时执行，在棒图控件上添加两个棒图，一个棒图与变量“原料罐温度”关联，标签为“原料罐”；第二个棒图与变量“反应罐温度”关联，标签为“反应罐”。

3) 单击画面命令语言编辑器的“存在时”标签，定义执行周期为 1000 毫秒。在命令语言编辑器中输入如下程序：

```
chartSetValue( "温度棒图", 1, \\本站点\原料罐温度);
```

```
chartSetValue ( "温度棒图", 1, \\本站点\反应罐温度);
```

这段程序将在画面被打开为当前画面时每 1000 毫秒用相关变量的值刷新一次控件。

4) 关闭命令语言编辑器，保存画面，则运行时打开该画面如图 14.6 所示。每 1000 毫秒系统会用相关变量的值刷新一次控件，而且控件的数值轴标记随绘制的棒图中最大的一个棒图值的变化而变化（这就是自动刻度）。

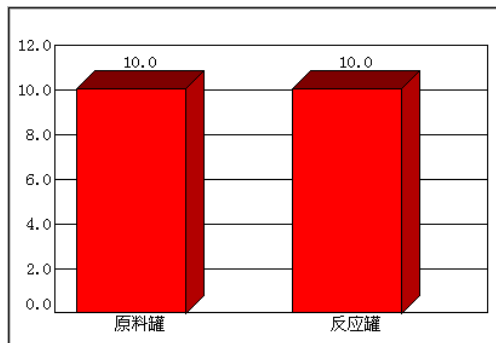


图 14.6 运行时的棒图控件

当画面中的棒图不再需要时，可以使用 `chartClearAll()` 函数清除当前的棒图，然后再用 `chartAdd()` 函数重新添加。

◇ **`chartSetSerialValue ("ControlName", nSerialIndex, nValueIndex, Value);`**

此函数为指定系列、指定索引的条形图设置数据值。语法格式如下：

```
chartSetSerialValue ("ControlName", nSerialIndex, nValueIndex, Value );
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

nSerialIndex: 所选系列的索引, 从 0 开始计。用于饼图时, nSerialIndex 值为 1。

nValueIndex: 整型变量, 表示饼状图索引号, 用于设定指定的饼状图, 从 1 开始计。

Value: 设定条形图的数据, 整形数据, 实型数据。

例如:

```
chartSetSerialValue ("XYChart", 1,1, \\本站点\原料罐温度);
```

此语句将在棒图控件 XYChart 中为第 1 个系列的索引为 1 的柱形图设置一个\\本站点\原料罐温度的值。

◇ **chartGetSerialNum("ControlName");**

此函数用于在指定的棒图控件中获得系列的数量, 仅用于条形图。语法格式如下:

```
chartGetSerialNum("ControlName");
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

例如:

```
chartGetSerialNum("XYChart");
```

此语句将获得在棒图控件 XYChart 中系列的数量, 返回值为整型。

◇ **chartGetSerialName("ControlName", nSerialIndex);**

此函数用于在指定的棒图控件中获取指定系列索引的系列名。语法格式如下:

```
chartGetSerialName("ControlName", nSerialIndex);
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

nSerialIndex: 所选系列的索引, 从 0 开始计。用于饼图时, nSerialIndex 值为 1。

例如:

```
chartGetSerialName("XYChart", 1);
```

此语句在棒图控件 XYChart 中获得指定系列的名称, 返回值为字符串。

◇ **chartGetDataNum("ControlName", nSerialIndex);**

此函数用于在指定的棒图控件中获得所选系列中的数据的数据的数量。语法格式如下:

```
chartGetDataNum("ControlName", nSerialIndex);
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

nSerialIndex: 所选系列的索引, 从 0 开始计。用于饼图时, nSerialIndex 值为 1。

例如:

```
chartGetDataNum("XYChart", 1);
```

此语句将获得在棒图控件 XYChart 中所选系列 1 中的数据的数据的数量, 返回值为整型。

◇ **chartDeletSerial("ControlName", nSerialIndex);**

此函数用于在指定的棒图控件中删除一个系列。语法格式如下:

```
chartDeletSerial("ControlName", nSerialIndex);
```

参数: ControlName: 工程人员定义的棒图控件名称, 可以为中文名或英文名。

nSerialIndex: 删除系列的索引, 从 0 开始计。

例如：

```
chartDeletSerial("XYChart", 1);
```

此语句将在棒图控件 XYChart 中删除索引为 1 的系列。

14.2.2 温控曲线控件

温控曲线反映出实际测量值按设定曲线变化的情况。在温控曲线中，纵轴代表温度值，横轴对应时间的变化，同时将每一个温度采样点显示在曲线中，另外还提供两个游标，当用户把游标放在某一个温度的采样点上时，该采样点的注释值就可以显示出来。主要适用于温度控制，流量控制等等。温控曲线的具体使用方法请参见本手册第八章 趋势曲线和其它曲线。

14.2.3 X-Y 轴曲线控件

X-Y 轴曲线控件可用于显示两个变量之间的数据关系，如电流—转速曲线等形式的曲线。组态王 7.0 提供了超级 X-Y 曲线控件，建议用户使用该控件。超级 X-Y 曲线控件的具体使用方法请参见本手册第八章 趋势曲线和其它曲线。

14.2.4 列表框和组合框控件

在列表框中，可以动态加载数据选项，当需要数据时，可以直接在列表框中选择，使与控件关联的变量获得数据。组合框是文本框与列表框的组合，可以在组合框的列表框中直接选择数据选项，也可以在组合框的文本框中直接输入数据。组态王 7.0 中列表框和组

合框的形式有：普通列表框、简单组合框、下拉式组合框、列表式组合框。它们只是在外观形式上不同，其它操作及函数使用方法都是相同的。列表框和组合框中的数据选项可以依靠组态王 7.0 提供的函数动态增加、修改，或从相关文件（.csv 格式的列表文件）中直接加载。

14.2.4.1 如何创建列表框控件

创建列表框控件的步骤如下：

- ◇ 单击工具箱中的“插入控件”按钮，如图 14.1，或选择画面开发系统中的“编辑/插入控件”菜单。系统弹出“创建控件”对话框，如图 14.2 所示。
- ◇ 在种类列表中选择“窗口控制”，在右侧的内容中选择“列表框”图标，单击对话框上的“创建”按钮，或直接双击“列表框”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如图 14.7 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

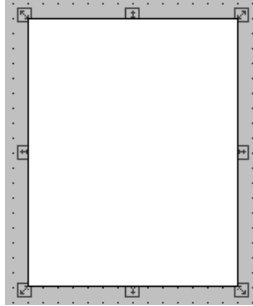


图 14.7 列表框控件

从外观上看，画面上放置的列表框控件与普通的矩形图素相似，但在进行动画连接和运行环境中是不同的。

14.2.4.2 设置列表框控件的属性

在使用列表框控件之前，需要先对控件的属性进行设置，设置控件名称、关联的变量和操作权限等。操作步骤如下：

- ◇ 用右键单击列表框控件，弹出浮动式菜单，如图 14.8 所示。



图 14.8 浮动式菜单

选择菜单命令“动画连接”，弹出“设置控件”对话框，或用左键双击列表框控件，弹出“设置控件”对话框，如图 14.9 所示。



图 14.9 列表框控件设置

- 1) **控件名称:** 定义控件的名称，一个列表框控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。
- 2) **变量名称:** 指定与当前列表框控件关联的变量，该变量为组态王 7.0 数据字典中已

定义的字符串型变量。

- 3) **访问权限:** 设置访问该列表框的操作级别, 权限级别从 1-999。
- 4) **排序:** 此选项有效时列表框中的内容按字母顺序排列。
- ✧ 列表框属性定义完成后, 单击“确认”按钮关闭对话框。

14.2.4.3 如何使用列表框控件

对于列表框控件中数据项的添加、修改、获取或删除等操作都是通过列表框控件函数实现的。首先认识一下列表框控件的函数:

listLoadList("ControlName", "Filename")

此函数用于将 CSV 格式文件“Filename”中的列表项调入指定的列表框控件“ControlName”中, 并替换列表框中的原有列表项。列表框中只显示列表项的成员名称(字符串信息), 而不显示相关的数据值。

listSaveList("ControlName", "Filename")

此函数用于将列表框控件“ControlName”中的列表项信息存入 CSV 格式文件“Filename”中。如果该文件不存在, 则直接创建。

listAddItem("ControlName", "MessageTag")

此函数将给定的列表项字符串信息“MessageTag”增加到指定的列表框控件“ControlName”中并显示出来。组态王 7.0 将增加的字符串信息作为列表框中的一个成员项——Item, 并自动给这个成员项定义一个索引号——ItemIndex, 索引号 ItemIndex 从 1 开始由小到大

自动加 1。

listClear("ControlName")

此函数将清除指定列表框控件“ControlName”中的所有列表成员项。

listDeleteItem("ControlName", ItemIndex)

此函数将在指定的列表框控件“ControlName”中删除索引号为 ItemIndex 的成员项。

listDeleteSelection("ControlName")

此函数将删除列表框控件“ControlName”中当前选定的成员项。

listFindItem("ControlName", "MessageTag", IndexTag)

此函数用于查找指定控件“ControlName”中与给定的成员字符串信息“MessageTag”相对应的索引号，并送给整型变量 IndexTag。

listGetItem("ControlName", ItemIndex, "StringTag")

此函数用于获取指定控件“ControlName”中索引号为 ItemIndex 的列表项成员字符串信息，并送给字符串变量 StringTag。

listGetItemCount("ControlName")

此函数用于获取指定控件“ControlName”中列表项的数目。

listGetCurSel("ControlName")

此函数用于获取指定控件“ControlName”当前选中列表项的 ID 号(从 0 开始)，返回值为 -1 说明当前控件没有选中项目。

listSetCurSel("ControlName", ItemIndex)

此函数用于将控件中索引号为 ItemIndex 的列表项设置为当前选中项，返回值为控件中当前选中项的索引号(从 0 开始)。如果返回值为-1，说明没有设置成功 (ItemIndex 为负数或超过项目数)。

listGetItemData("ControlName", ItemIndex, NumberTag)

此函数用于获取指定控件“ControlName”中索引号为 ItemIndex 的列表项中的数据值，并送给整型变量 NumberTag。

listInsertItem("ControlName", ItemIndex, "StringTag")

此函数将字符串信息 StringTag 插入到指定控件“ControlName”中列表项索引号为 ItemIndex 所指示的位置。如果 ItemIndex=-1，则字符串信息 StringTag 被插入到列表项的最尾端。

`listSetItemData("ControlName", ItemIndex, Number)`

此函数用于将变量 Number 的值设置到指定控件“ControlName”中索引号为 ItemIndex 的列表项中。

`ListLoadFileName("CtrlName", "*.ext")`

此函数将“*.ext”指示的文件名显示在指定控件“ControlName”列表框中。函数的具体参数及用法请参见《组态王 7.0 函数手册》。



例 1:

制作一个动态的列表，可以向列表框中动态添加数据，添加完成后，需要保存列表为文件，文件保存在当前工程路径下（如 D:\Test），在以后使用。需要时要从文件中读出列表信息。操作步骤如下：

- 1) 在组态王 7.0 数据词典中定义变量“列表数据”字符串变量。
- 2) 在画面上创建列表框控件，定义控件属性如图 14.10 所示。



图 14.10 定义列表框属性

- 3) 在画面上创建三个按钮，如图 14.11 所示。按钮的作用和连接的动画连接命令语

言分别为:

- 按钮 1——“增加”: 增加数据项:

```
listAddItem("列表框 1", 列表数据);
```

- 按钮 2——“保存”: 保存列表框内容:

```
listSaveList("列表框 1", "D:\Test\list1.csv");
```

- 按钮 3——“加载”: 将指定 csv 文件中的内容加载到列表框中来:

```
listLoadList("列表框 1", "D:\Test\list1.csv");
```

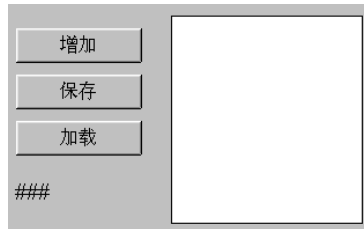


图 14.11 创建列表框和操作按钮

- 4) 在画面上创建一个文本图素, 定义动画连接为字符串值输入和字符串值输出, 连接的变量为“列表数据”。
- 5) 保存画面, 切换到组态王 7.0 运行系统, 在文本图素中输入数据项的字符串值, 如“数据项 1”。如图 14.12 所示。单击“增加”按钮, 则变量的内容增加到了列表框中。

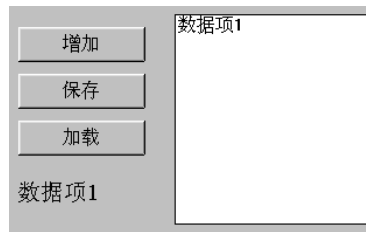


图 14.12 向列表框中增加数据项

按照上面的方法，可以向列表框中增加多个数据项。当在列表框中选中某一项时，与列表框关联的变量可以自动获得当前选择的数据项的字符串值。如图 14.13 所示。

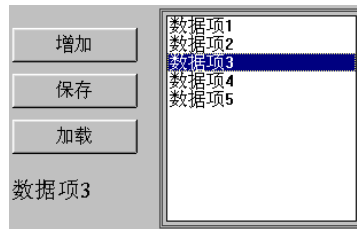


图 14.13 在列表框中选择数据项

可以将列表框中的数据项保存起来。单击“保存”按钮。当需要将保存的数据加载到列表框时，单击“加载”按钮，原保存的列表数据就被加载到当前列表框中来。



例 2:

将指定路径下 (C:\Program Files\Kingview) 的扩展名为 “.exe” 的文件名列到列表框中来。可以在命令语言中使用函数: ListLoadFileName ()。操作步骤如下:

- 在画面上增加按钮，定义为“可执行文件”，如图 14.14 所示。双击按钮，定

义其动画连接——命令语言连接——弹起时为：

```
ListLoadFileName( "列表框 1", " C:\Program Files\Kingview\*.exe" );
```

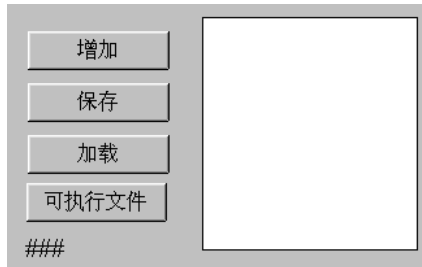


图 14.14 增加调用按钮

- 保存画面，切换到运行系统，单击该按钮，可以将指定目录下扩展名为“*.exe”的文件名全部列到列表框中来，如图 14.15 所示。

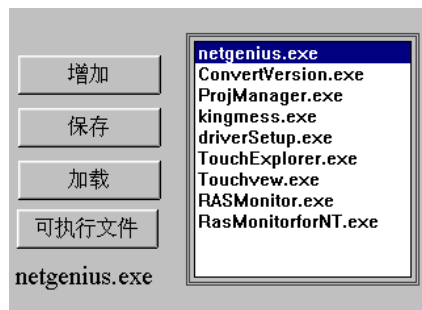


图 14.15 执行函数结果

14.2.4.4 如何使用组合框控件

组合框的创建与列表框的创建过程、方法相同。组合框是由列表框和文本编辑框组合而成的。组合框有三种类型：简单组合框，如图 14.16 所示；下拉式组合框如图 14.17

所示；列表式组合框，如图 14.18 所示。组合框属性的定义方法与列表框的定义方法相同。

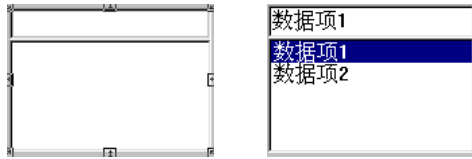


图 14.16 简单组合框

◇ 简单组合框：

简单组合框创建后，其列表框的大小已经为创建时的大小。当列表项超出列表框显示时，列表框会自动加载垂直滚动条。将鼠标光标置于文本编辑框中时，可以直接输入不在当前列表中的数据项。

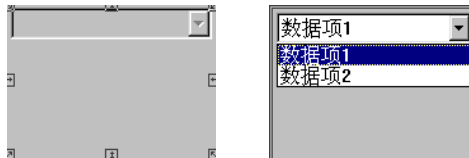


图 14.17 下拉式组合框

◇ 下拉式组合框：

下拉式组合框创建后，其文本编辑框是灰色无效的，表示该文本编辑框在运行中是禁止添加数据的。当用户在运行系统中单击该文本编辑框时，会弹出列表框。单击下拉箭头也会弹出列表框。通常情况下，下拉式组合框的列表框是隐藏的，除非单击文本编辑框或单击下拉箭头。表示只能从列表中选择数据项。



图 14.18 列表式组合框

◇ 列表式组合框：

列表式组合框兼有简单组合框和下拉式组合框的功能。通常组合框的列表框是隐藏的，当单击下拉箭头时，才弹出列表框。选择完数据项后，列表框自动隐藏。在列表式组合框的文本框中可以直接输入数据项。

对于简单组合框控件和列表式组合框控件，在文本框中输入字符串时，控件关联的字符串变量的值也随之改变。例如，放置简单组合框控件，控件属性如下图所示，在“组合框 1”控件的文本框内输入字符串时，变量“数据选项”的值随之改变。



组合框的操作也是通过函数实现的，所使用的函数和使用方法与列表框完全相同。请参见 14.2.4.3 节中列表框使用的例程。

14.2.5 复选框控件

复选框控件可以用于控制离散型变量，如用于控制现场中的各种开关，做各种多选选项的判断条件等。复选框一个控件连接一个变量，其值的变化不受其它同类控件的影响，当控件被选中时，变量置为 1，不选中时，变量置为 0。

14.2.5.1 如何创建复选框控件

在画面开发系统的工具箱中选择“插入控件”按钮，或选择菜单“编辑/插入控件”命令，在弹出的如图 14.2 所示的“创建控件”对话框中，在种类列表中选择“窗口控制”，在右侧的内容中选择“复选框”图标，单击对话框上的“创建”按钮，或直接双击“复选框”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如图 14.19 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。



图 14.19 创建复选框控件

14.2.5.2 如何设置复选框控件的属性

在使用复选框控件前，需要先对控件的属性进行设置，在画面上双击控件，弹出“复选框控件属性”对话框，如图 14.20 所示。



图 14.20 复选框控件属性对话框

- ◇ **控件名称:** 定义控件的名称，一个列表框控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。如：“switch”。
- ◇ **变量名称:** 与控件关联的变量名称，一般为离散型变量。如离散型变量：“开关”。当复选框被选中时，该变量的值为 1；否则为 0。
- ◇ **访问权限:** 定义控件的安全级别——访问权限，范围为 1~999。定义了访问权限后，运行时，只有符合该安全级别的用户登录后，才能操作控件，否则，修改不了控件的值。如 100。
- ◇ **标题文本:** 控件在画面上显示的提示文本、说明性的文本，如：“电源开关”。定义完成的控件如图 14.21 所示。



图 14.21 复选框属性定义及结果

14.2.5.3 如何使用复选框控件

复选框控件没有控件命令语言函数，只需要使用“设置控件”对话框中的变量即可。

如图 14.21 所示定义控件属性与变量相关联。



例如：

用复选框控件控制一个开关。定义步骤如下：

1. 在画面上创建复选框控件，定义控件属性如图 14.21 所示。
2. 在画面上创建文本图素，定义文本的动画连接——离散值输出动画连接，如图 14.22 所示。动画连接的变量为与控件关联的变量“开关”。
3. 保存画面，切换到运行系统。



图 14.22 创建动画连接

运行系统中单击该复选框控件时，变量值的变化与控件选择关系的变化如图 14.23 所示。



图 14.23 运行时用复选框控制变量的值

复选框控件除了上述用法外，还可以作为条件选项来使用，如多选选项等。

14.2.6 编辑框控件

控件用于输入文本字符串并送入指定的字符串变量中。输入时不会弹出虚拟键盘或其它的对话框。

14.2.6.1 如何创建编辑框控件

在画面开发系统的工具箱中选择“插入控件”按钮，或选择菜单“编辑/插入控件”命令，在弹出的如图 14.2 所示的“创建控件”对话框中，在种类列表中选择“窗口控制”，在右侧的内容中选择“编辑框”图标，单击对话框上的“创建”按钮，或直接双击“编辑框”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如图 14.24 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

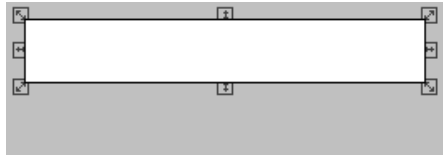


图 14.24 创建后的编辑框控件

14.2.6.2 如何定义编辑框控件属性

控件创建后，要定义其属性，才能使用。双击控件，或选择控件，然后在控件上单击鼠标右键，在弹出的快捷菜单上选择“动画连接”命令，弹出如图 14.25 所示的控件属性对话框。



图 14.25 编辑框控件属性对话框

- ◇ **控件名称：**定义控件的名称，一个列表框控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。
- ◇ **变量名称：**指定与当前编辑框控件关联的变量，该变量为组态王 7.0 数据字典中已定义的字符串型变量。
- ◇ **访问权限：**设置访问该列表框的操作级别，权限级别从 1-999。

- ◇ **风格：**定义编辑框控件的使用风格。编辑框控件的风格有以下几类：
 - **多行文字：**允许在编辑框中显示多行文字，该选项一般与“接收换行”一起使用。
 - **密码显示：**当在编辑框中输入字符时，实际的输入字符不会显示，而只显示为“*”号。
 - **接收换行：**在编辑框中输入字符时，如果按下回车键_<Enter>，则可以在编辑框中换行输入。一般与“多行文字”选项一起使用。
 - **全部大写：**当在编辑框中输入英文字符时，无论输入的是大写还是小写，都转换为大写显示。
 - **全部小写：**当在编辑框中输入英文字符时，无论输入的是大写还是小写，都转换为小写显示。

14.2.6.3 如何使用编辑框控件

编辑框控件没有控件命令语言函数，只需要定义其属性与字符串变量连接即可。因为组态王 7.0 中的字符串长度为 127 个字符，所以组态王 7.0 的编辑框控件只接收 127 个字符的输入。编辑框控件可以用于在画面上直接输入字符，或输入密码等使用。



例如：要求画面上输入的字符显示为“*”，不想被其他人看到输入内容，如密码输入。

- 在画面上创建编辑框控件。在组态王 7.0 中定义字符串变量——“密码”。
- 定义控件属性如图 14.26 所示。在“风格”选项中选择“密码显示”。定

义完成后，单击“确认”按钮，关闭对话框。保存画面，切换到运行系统。



图 14.26 定义编辑框属性

- 在运行系统中打开该画面，在编辑框中输入字符时，显示如图 14.27 所示。当在编辑框中输入字符时，全部显示为“*”，看不到实际输入内容。



图 14.27 密码显示输入

14.2.7 单选按钮控件

当出现多选一的情况时，可以使用单选按钮来实现。单选按钮控件实际是由一组单个的选项按钮组合而成的。在每一组中，每次只能选择一个选项。

14.2.7.1 如何创建单选按钮控件

在画面开发系统的工具箱中选择“插入控件”按钮，或选择菜单“编辑/插入控件”

命令，在弹出的如图 14.2 所示的“创建控件”对话框中，在种类列表中选择“窗口控制”，在右侧的内容中选择“单选按钮”图标，单击对话框上的“创建”按钮，或直接双击“单选按钮”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如图 14.28 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

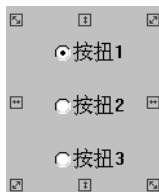


图 14.28 创建单选按钮控件

14.2.7.2 如何定义单选按钮控件属性

控件创建后，要定义其属性，才能使用。双击控件，或选择控件，然后在控件上单击鼠标右键，在弹出的快捷菜单上选择“动画连接”命令，弹出如图 14.29 所示的控件属性对话框。



图 14.29 单选按钮控件属性

- ◇ **控件名称：**定义控件的名称，一个单选按钮控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。如“optionbutton1”。
- ◇ **变量名称：**与控件关联的变量名称，一般为整型变量。如整型变量：“单选项”。每选择一个单选按钮时，该整型变量将得到不同的数值。
- ◇ **访问权限：**定义控件的安全级别——访问权限，范围为 1~999。定义了访问权限后，运行时，只有符合该安全级别的用户登录后，才能操作控件，否则，修改不了控件的值。如 100。
- ◇ **标题数据：**控件在画面上显示时每个单选按钮的标题文本，如：“选项 1”、“选项 2”等。标题数据定义项由一个组合列表框和三个功能按钮组成。选择列表框中的某一项，可以“修改”当前选中的项的标题文本，也可以在当前位置

的前边“插入”一项，或“删除”选择的项。

✧ **按钮数**：表示当前总共定义的单选按钮的个数。按钮数最多定义为 100 个，最少不少于 2 个。

✧ **对齐选项**：定义单选按钮的排列方式。“横向对齐”表示单选按钮将横向排列为一行；“纵向对齐”表示单选按钮将纵向排列为一系列。定义完成的控件如图 14.30 所示。

14.2.7.3 如何使用单选按钮控件

单选按钮控件没有控件命令语言函数，只需要使用“设置控件”对话框中的变量即可。

如图 14.30 所示定义控件属性与变量相关联。

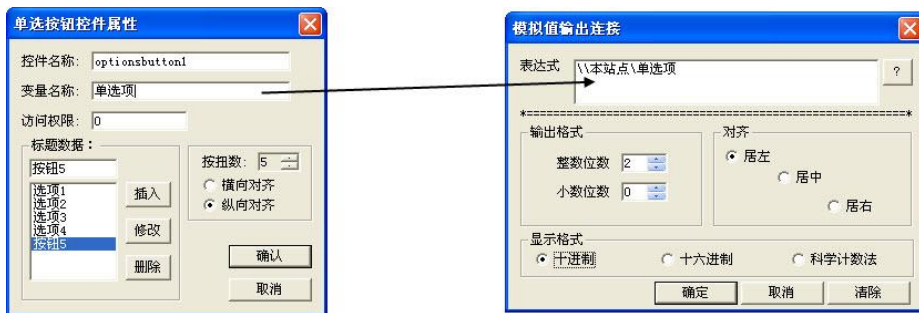


图 14.30 定义单选按钮控件属性



例：用复选框控件控制一个开关。定义步骤如下：

- 在画面上创建单选按钮控件，定义控件属性如图 14.30 所示。
- 在画面上创建文本图素，定义图素的动画连接属性为“模拟值输出”，关

联的变量为单选按钮中关联的变量。

- 定义完成后，保存画面，切换到组态王 7.0 运行系统，打开该画面。用鼠标左键单击不同的按钮选项时，得到的变量的值不相同。如图 14.31 所示。

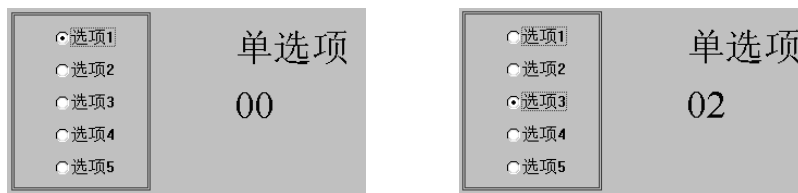


图 14.31 单选按钮的使用

可以根据关联的变量的值的不同来判断用户选择了哪一项。

14.2.8 超级文本显示控件

组态王 7.0 提供一个超级文本显示控件，用于显示 RTF 格式或 TXT 格式的文本文件，而且也可在超级文本显示控件中输入文本字符串，然后将其保存成指定的文件，调入 RTF、TXT 格式的文件和保存文件通过超级文本显示控件函数来完成。

14.2.8.1 如何创建超级文本显示控件

在画面开发系统的工具箱中选择“插入控件”按钮，或选择菜单“编辑/插入控件”命令，在弹出的如图 14.2 所示的“创建控件”对话框中，在种类列表中选择“超级文本显示”，在右侧的内容中选择“显示框”图标，单击对话框上的“创建”按钮，或直接双击“显示框”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松

开鼠标左键，控件在画面上显示出来。如图 14.32 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

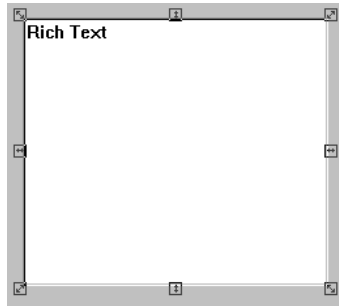


图 14.32 创建后的超级文本显示框控件

14.2.8.2 如何定义超级文本显示控件的属性

控件创建完成后，需要定义空间的属性。用鼠标双击控件，弹出超级文本显示框控件属性对话框，如图 14.33 所示。

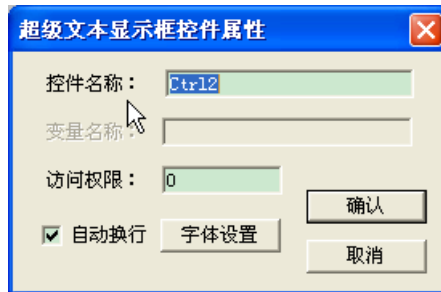
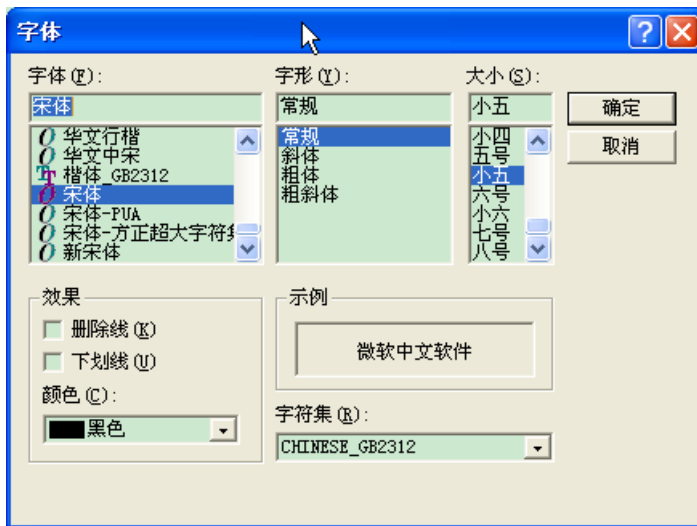


图 14.33 超级文本显示框控件属性对话框

- ◇ **控件名称：**定义控件的名称，一个显示框控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。如“Richtxt1”。

- ◇ **访问权限:** 定义控件的安全级别——访问权限, 范围为 1~999。定义了访问权限后, 运行时, 只有符合该安全级别的用户登录后, 才能操作控件, 否则, 修改不了控件的值。如 100。
- ◇ **自动换行:** 选中该选项后, 文本框显示的内容超出当前行范围时自动换行。
- ◇ **字体设置:** 可以设置字体颜色及大小, 如下图所示:



属性定义完成后, 单击“确认”按钮关闭对话框。

14.2.8.3 如何使用超级文本显示控件

超级文本显示框的作用是显示 RTF 格式或 TXT 格式的文本文件的内容, 或在显示框中输入文本字符串, 将其保存为 RTF 格式或 TXT 格式的文本文件。实现以上这些要依靠组态王 7.0 提供的两个函数实现:

- ◇ **LoadText() 函数:** 将指定 RTF 格式或 TXT 格式文件的内容加载到文本显示框里。

◇ **SaveText() 函数：** 将显示框里的内容保存为指定的 RTF 格式或 TXT 格式文件。

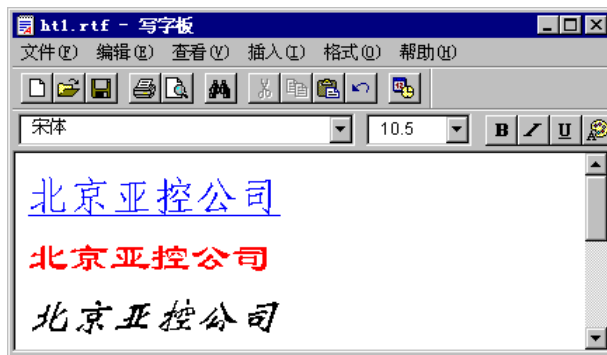
下面以两个具体例子说明超级文本显示控件的使用及操作步骤



例 1: 编写 RTF 格式的文件

第一步：

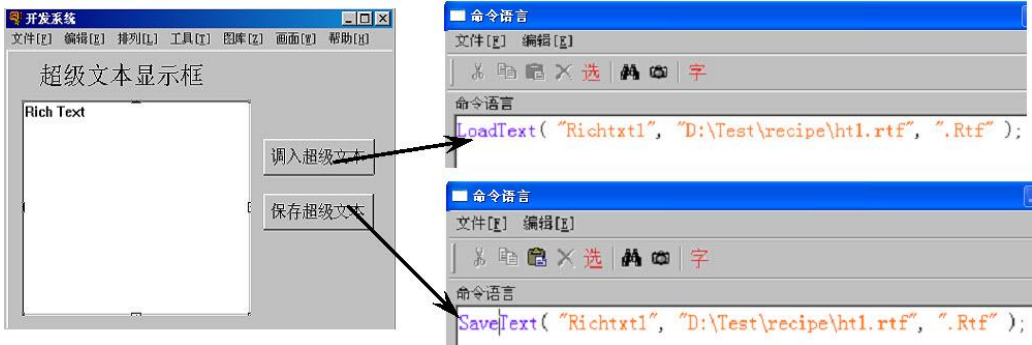
用 WINDOWS 操作系统的写字板编写一个 RTF 文件 ht1.rtf，其内容如下



将文件保存在指定的目录下，比如目录 D:\Test\recipe 下。

第二步：

在组态王 7.0 画面开发系统放置超级文本显示控件以及相应的操作按钮放置超级文本显示控件，控件名设为“Richtxt1”，然后再放置两个命令按钮，并将这两个按钮分别进行命令语言连接，如下图所示。



按钮“调入超级文本”的命令语言为

```
LoadText( "Richtxt1", "D:\Test\recipe\htl.rtf", ".Rtf" );
```

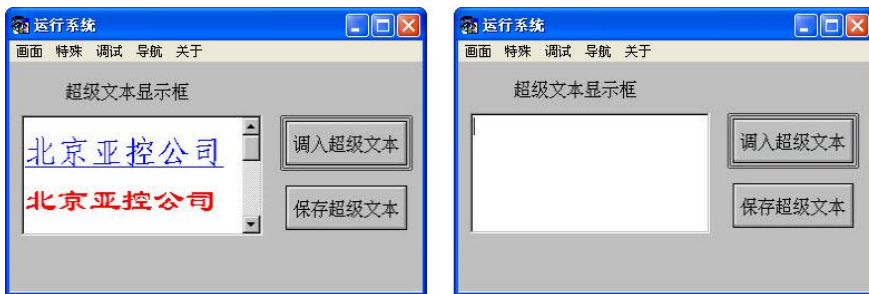
按钮“保存超级文本”的命令语言为

```
SaveText( "Richtxt1", "D:\Test\recipe\htl.rtf", ".Rtf" );
```

将画面文件全部保存。

第三步：

运行画面。启动组态王 7.0 运行系统，单击“调入超级文本”按钮，其结果如下。



执行“调入超级文本”按钮前

执行“调入超级文本”按钮后

如果修改显示框中的内容，然后单击“保存超级文本”按钮，可以将显示框中的内容保存

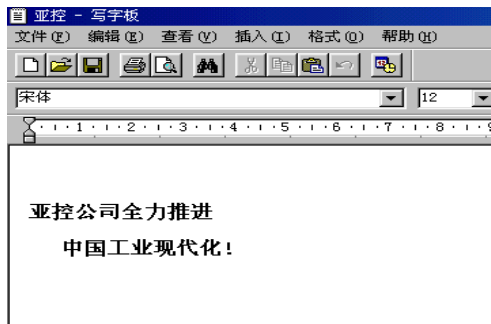
到指定的文件中。



例 2: 编写 Txt 格式的文件

第一步:

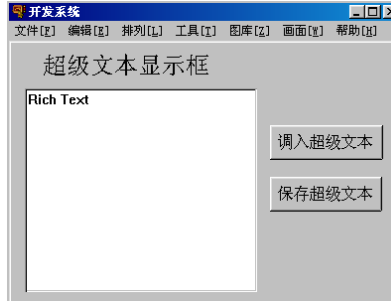
用 WINDOWS 操作系统的写字板编写一个 Txt 文件 ht2.txt，其内容如下：



将文件保存在指定的目录下，比如目录 D:\Test\recipe 下。

第二步:

在组态王 7.0 画面开发系统放置超级文本显示控件以及相应的操作按钮放置超级文本显示控件，控件名设为“Richtxt2”，然后再放置两个命令按钮，并将这两个按钮分别进行命令语言连接，如下所示：



按钮“调入超级文本”的命令语言为

```
LoadText( " Richtxt2", "D:\Test\recipe\ht2.txt", ".Txt" );
```

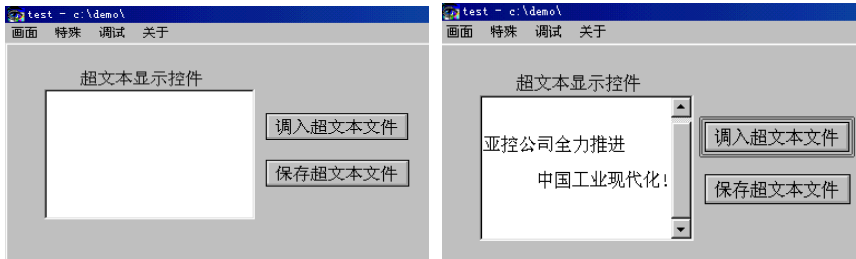
按钮“保存超级文本”的命令语言为

```
SaveText( " Richtxt2", "D:\Test\recipe\ht2.txt", ".Txt" );
```

将画面文件全部保存。

第三步：

运行画面。启动组态王 7.0 运行系统，单击“调入超级文本”按钮，其结果如下：



执行“调入超级文本”按钮前 执行“调入超级文本”按钮后

如果修改显示框中的内容，然后单击“保存超级文本”按钮，可以将显示框中的内容保存到指定的文件中。

14.2.9 多媒体控件

组态王 7.0 提供的多媒体控件有：动画播放控件（播放*.avi 文件）和视频输出控件。其中视频输出控件组态王 7.0 内部控件有两个：视频控件和 TDM 视频控件，另外是 ACTIVE X 视频控件，该控件的使用介绍请参见 14.3.3 节的介绍。

14.2.9.1 AVI 动画播放控件

AVI 动画播放控件是专门用来播放 AVI 格式的动画文件的。

14.2.9.1.1 如何创建 AVI 动画播放控件

在画面开发系统的工具箱中选择“插入控件”按钮，或选择菜单“编辑/插入控件”命令，在弹出的如图 14.2 所示的“创建控件”对话框中，在种类列表中选择“多媒体”，在右侧的内容中选择“AVI 动画”图标，单击对话框上的“创建”按钮，或直接双击“显示框”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来。如图 14.34 所示。控件周围有带箭头的小矩形框，鼠标挪到小矩形框上，鼠标箭头变为方向箭头时，按下鼠标左键并拖动，可以改变控件的大小。当鼠标在控件上变为双“十”字型时，按下鼠标左键并拖动，可以改变控件的位置。

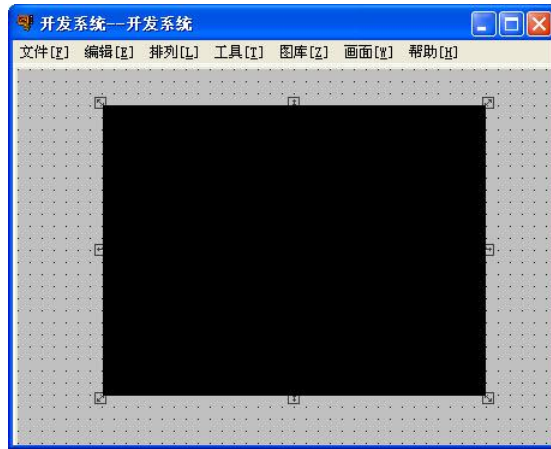


图 14.34 创建后的 AVI 控件

14.2.9.1.2 如何定义 AVI 动画播放控件的属性

在使用 AVI 控件之前，必须对其属性进行设置。用鼠标左键双击控件，弹出控件属性对话框如图 14.35 所示。



图 14.35 AVI 控件属性定义对话框

控件名称: 定义控件的名称，一个单选按钮控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。如“AVICtrl”。

14.2.9.1.3 如何使用 AVI 动画播放控件

AVI 动画控件的驱动是靠组态王 7.0 提供的一个函数实现的，该函数为：

```
PlayAvi( "CtrlName", filename, option );
```

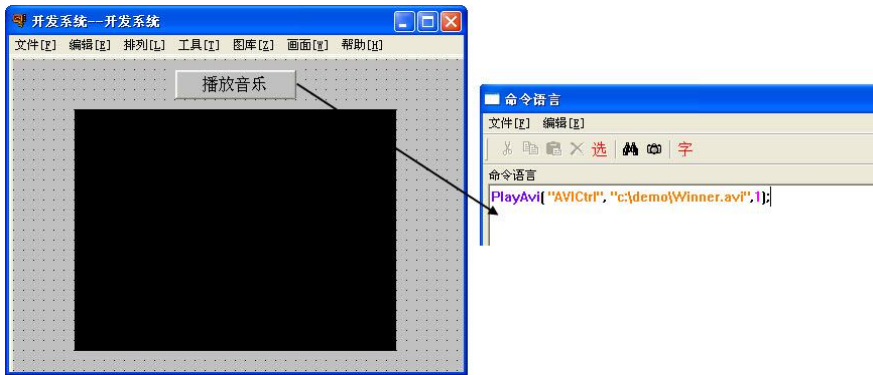
- ✧ **CtrlName**：用于播放播放 AVI 动画的控件的名称。
- ✧ **filename**：代表要播放的动画文件的字符串或字符串变量。
- ✧ **option**：可为下述之一：
 - 0：停止播放 AVI 动画
 - 1：播放一遍 AVI 动画
 - 2：连续播放 AVI 动画，直到接受到停止播放的信息为止。

下面以一个具体例子说明 AVI 动画控件函数的使用及操作步骤。



例：设定一个按钮，将 avi 文件 c:\demo\Winner.avi 调入控件名为“AVICtrl”的 AVI 动画控件中，操作步骤如下：

- ✧ 第一步：在画面上画出文本按钮“播放动画”
- ✧ 第二步：将此按钮进行控件命令语言连接，双击该按钮，弹出“命令语言连接”对话框，文本按钮和命令语言连接的对应关系如下：



- ◇ 第三步：将画面文件全部保存后，进入组态王 7.0 运行系统，单击“播放动画”按钮，则播放 winner.avi 中的内容如下所示



⚠ 注意：

当动画播放完后，画面停止在 AVI 文件的最后一帧。

14.2.9.2 视频动画控件

该视频动画控件是一个比较简单的控件，它只能针对一路视频输入，无须用户做过多的配置。这里只做简单的介绍。因为没有输入通道的选择，一般该控件适用于单通道视频

输入，即整个系统只监视一个视频图像的情况。否则请选用 ACTIVE X 视频控件（14.3.3 节介绍）。

1、在画面上放置视频输出控件

操作步骤如下：

- ◇ 第一步：单击工具箱中的“插入控件”按钮或选择菜单命令“编辑\插入控件”，则弹出“创建控件”对话框
- ◇ 第二步：在“创建控件”对话框内从控件种类为多媒体中选择视频控件图标
- ◇ 第三步：用鼠标左键单击“创建”按钮，鼠标变成十字形。然后在画面上画一个矩形框，视频动画控件就放到画面上了。可以任意移动、缩放视频输出控件，如同处理一个单元一样。

在画面上放置的视频动画控件如图 14.36 所示。

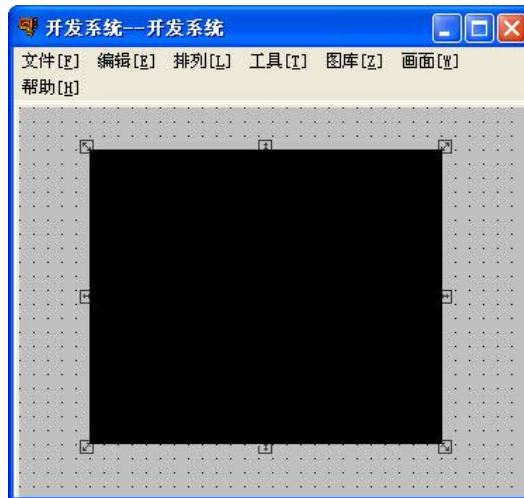


图 14.36 视频动画控件

2、视频动画控件的设置

用左键双击视频控件，弹出“设置控件”对话框，或者用右键单击视频动画控件选择菜单命令“动画连接”，则也弹出“设置控件”对话框如图 14.37 所示：



图 14.37 视频动画控件设置

- ◇ 控件名：一个视频动画控件对应一个控件名称，而且是唯一的，不能重复命名，控件名由工程人员设定，例如设为“监控”。

3、视频控件的使用

首先用户将视频输入卡插入计算机，启动计算机并安装视频卡的驱动程序，一般视频卡都自带视频使用软件，使用该软件对视频卡做必要的配置，并检查视频卡工作是否正常。一切正常后，关闭该软件程序，在组态王 7.0 中创建视频控件，切换到组态王 7.0 运行系统。就可以看到视频图像了。



例：在大厅内安装一个探头，在屏幕上设置一个视频控件，则探头所拍摄下来的内容全部在屏幕上显示出来，如下图所示



⚠ 注意:

在组态王 7.0 运行系统中，不能用两个视频控件同时显示同一视频信息。

14.2.9.3 AVICapture 视频动画控件

AVICapture 控件只能针对一路视频输入，具有拍摄、摄像、回放等功能。用户可以抓拍瞬间的一个画面保存为图片，或录制一段视频信息保存为 AVI 格式的文件，通过“回放”功能来查看，回放时可以将画面进行重放、放大或缩小。

14.3 Active X 控件

14.3.1 Active X 控件概述

组态王7.0支持Windows标准的Active X控件，包括Microsoft提供的标准Active X控件和用户自制的Active X控件。Active X控件的引入在很大程度上方便了用户，用户可以

灵活地编制一个符合自身需要的控件，或调用一个已有的标准控件，来完成一项复杂的任务，而无须在组态王7.0中做大量的复杂的工作。一般的Active X控件都具有控件属性、控件方法、控件事件，用户在组态王7.0中通过调用控件的这些属性、事件、方法来完成任务。

14.3.2 如何在组态王 7.0 中使用 Active X 控件

14.3.2.1 创建 Active X 控件

如图 14.40 所示，在组态王 7.0 工具箱上单击“插入通用控件”或选择菜单“编辑\插入通用控件”命令。弹出“插入控件”对话框，如图 14.41 所示。



图 14.40 插入通用控件按钮

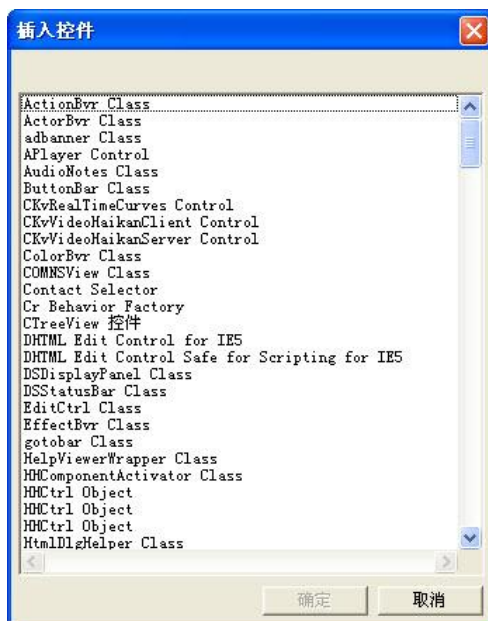




图 14.41 插入通用控件对话框

在对话框的列表框中列出了本机上已经注册到 Windows 的 Active X 控件名称，用户从中通过单击鼠标左键来选择所需的控件，在列表框的下方的标签文本显示当前选中的 Active X 控件所对应的文件。单击“取消”按钮取消插入控件操作；选中控件名称后单击“确定”按钮或用鼠标左键双击该列表项，则插入控件对话框自动关闭，鼠标箭头变为小“十”字型，在画面上选择要插入控件的位置，按下鼠标左键，然后拖动，直到拖动出的矩形框大小满足所需，放开鼠标左键，则创建的控件便出现在画面上。

 注意：

有些控件本身不是很大，而且有着自身的尺寸大小，所以创建后，有可能恢复

到其本身大小

 例：需要在画面上显示一个日历，供操作人员来选择所需的日期。操作步骤如下：

1、 在工具箱上单击“插入通用控件”或选择菜单“编辑\插入通用控件”命令。会弹出如图14.41所示的“插入控件”对话框。

2、 在对话框的列表中找到“Microsoft Date and Time Picker control”项，选中它，然后单击“确定”按钮，或直接鼠标双击该项。“插入控件”对话框自动关闭，按照本节开始所描述的方法在画面上创建控件。创建后的控件如图14.42所示。

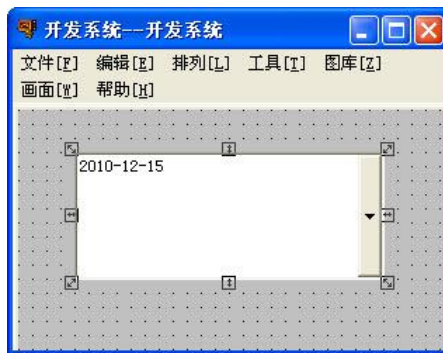


图14.42 创建后的日历控件

有些特殊的 Active X 控件在组态王 7.0 中无法支持，所以当用户在创建控件时，会有如下图所示的提示框。表明该控件无法在组态王 7.0 中创建的使用。



14.3.2.2 设置 Active X 控件的固有属性

根据控件的特点，有些控件带有固定的属性设置界面——这些属性界面在组态王7.0里称为控件的“固有属性”。通过这些固有属性，可以设置控件的操作状态、控件的外观、颜色、字体或其它一些的属性等。设置的固有属性一般为控件的初始状态。每个控件的固有属性页都各不相同。

设置固有属性的方法为，首先选中控件，在控件上单击鼠标右键，系统弹出快捷菜单，选择“控件属性”命令。如果用户创建的控件有属性页的话，则会直接弹出控件的属性页。

如图14.43所示，以上节中创建的日历控件为例，在控件上单击鼠标右键，选择弹出的快捷菜单中的“控件属性”项，弹出如图14.44所示的日历控件的固有属性页。在这个属性页中，可以设置日历控件的初始值、日期范围、格式、字体、各部分显示颜色等属性。大多数固有属性可以在运行时通过控件的属性来修改。

控件属性页的使用与一般编程程序中使用相同，这里不再详细介绍，用户可以查看相应的控件厂家的帮助说明来了解和设置控件的属性页。



图14.43 选择“控件属性”菜单命令



图14.44 日历控件固有属性

控件的固有属性按照实际使用情况，可以设置也可以不设置。主要看用户的使用需求。

14.3.2.3 设置 Active X 控件的动画连接属性

在组态王 7.0 中使用 Active X 控件，最重要的是要设置它的动画连接属性。动画连接属性是控件与组态王 7.0 相联系的关键。在动画连接属性中要定义控件在组态王 7.0 中的标记名称，安全级别等。

1、控件动画连接属性——常规页

以上节创建的日历控件为例，用鼠标双击控件，系统弹出控件“动画连接属性”对话框。如图14.45所示。动画连接属性页由三页组成：常规、属性和事件。首先显示的是常规页。

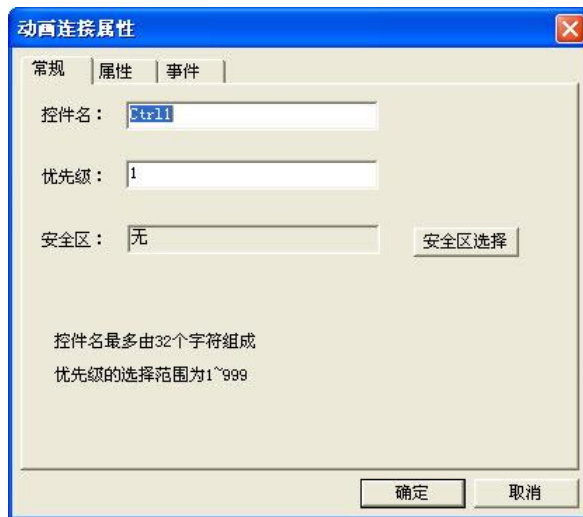


图14.45 控件动画连接属性对话框——常规页

- ◇ **控件名：**定义控件的名称，一个控件对应一个控件名称，而且是唯一的，不能重复命名，控件的命名应该符合组态王 7.0 的命名规则。如“DatTimCtrl”。

- ◇ **优先级、安全区：**定义控件的安全访问级别。优先级的输入范围为 1~999，单击“安全区选择”按钮，弹出如图 14.46 所示的“选择安全区”对话框，单击中间的按钮，可以选入和选出安全区，当鼠标位于某个按钮上时，在对话框的底部有文字标签显示按钮的作用。可以选择已定义的安全区，也可以多选。运行时，只有符合该安全级别的用户登录后，才能操作控件，否则，操作不了控件。

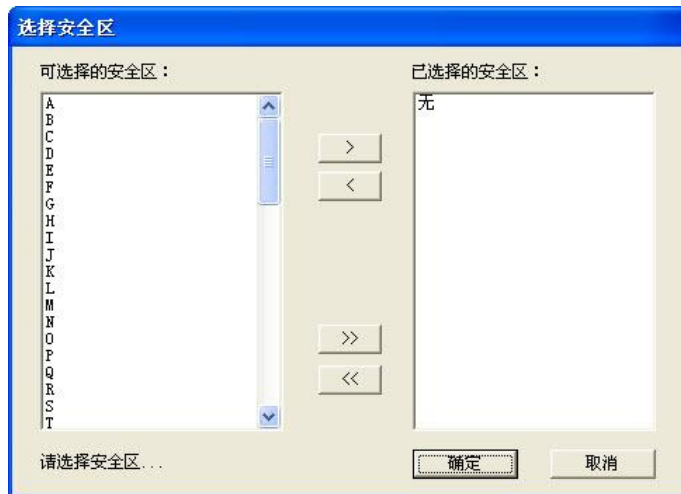


图14.46 控件动画连接属性中安全区的选择

2、动画连接属性页——属性页

单击动画连接属性的“属性”标签，显示“属性”页。如图14.47所示。



图14.47 控件动画连接属性对话框——属性页

在“属性”页的列表中，列出了控件的所有属性。该属性页的主要作用是提供控件属性与组态王7.0变量相关联的界面。其中列表共分为三列：属性列、类型列和关联变量列。属性列列举了控件的所有属性；类型列标明了相应属性的数据类型；可以在动画连接属性中直接将相关属性与组态王7.0的变量相关联——关联变量。按照实际需要，使用控件属性时，可以关联变量也可以不关联。



注意：

控件中有组态王7.0不支持的数据类型的属性将在对话框中显示为“UNKNOWN”，其关联变量的表格显示为灰色，表明该属性在组态王7.0中不能使用。以下相同。

关联变量时，在允许关联属性的“关联变量”的表格中单击鼠标右键，弹出快捷菜单，如图14.48所示。菜单项共有三项内容：

添加：添加变量与控件属性相关联

编辑：编辑当前关联变量的属性，可以更改关联方向或更换变量

删除：删除当前属性关联的变量



图14.48 关联变量的快捷菜单

如果选择的属性没有关联变量，则“编辑”、“删除”项无效。如给日历的“Value”属性关联一个组态王7.0变量。在“Value”属性关联变量表格中单击右键，选择添加后，弹出变量浏览器，如图14.49所示。

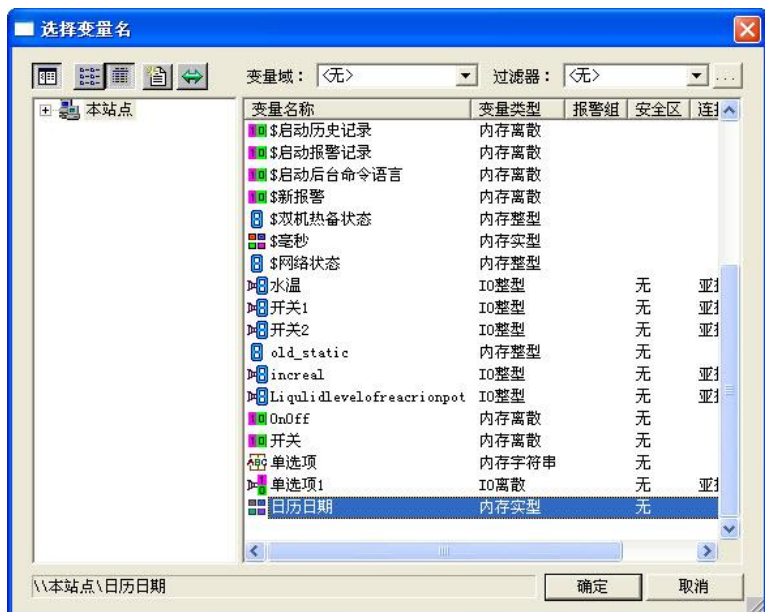



图14.49 关联变量——变量浏览器

首先在变量浏览器中选择要关联的变量，然后选择属性和变量的关联方向。在关联变量选择的变量浏览器中有一个特殊的按钮 。单击该按钮弹出一个选择关联方向的对话框，如图14.50所示。



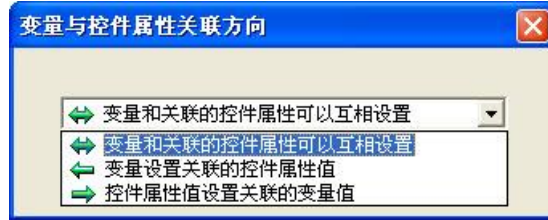


图14.50 变量与控件属性关联方向对话框

关联方向共有三种选择：

- **变量和关联的控件属性可以互相设置：**无论变量的值或控件属性值发生变化，都可以设置对方的值
- **变量设置关联的控件属性值：**变量的值变化时，可以设置控件属性值，但控件属性值变化时，变量的值不会发生变化
- **控件属性值设置关联的变量值：**控件属性值变化时，可以设置变量的值，但变量的值变化时，控件属性的值不会发生变化

用户根据实际需要设置关联方向。如在日历控件中选择“控件属性值设置关联的变量值”。设置完成后，关闭变量浏览器，在控件动画连接对话框中的“Value”各种出现了如图 14.51 所示的内容。关联变量的前面出现的箭头“—>”标明关联方向。切换到组态王 7.0 运行系统，当改变控件的属性“Value”值时，可以得到变化了的变量“日历日期”的值；但修改变量的值时，控件的属性值并不变化。



图14.51 关联变量或的动画连接属性

3、动画连接属性页——事件页


单击动画连接属性的“事件”标签，显示“事件”页。如图14.52所示。



图14.52 控件动画连接属性对话框——事件页

在事件属性页中，列举出了该控件的所有事件。如单击控件事件——“Click”，控件数据变化事件——“Change”等。控件的各个事件的具体含义请参见控件的说明文档（如本例中的日历控件可以查看MSDN等资料）。

事件属性页列表共分为两列：第一列为事件的名称；第二列为与该事件关联的组态王7.0自定义函数，即可以定义一个组态王7.0自定义函数，当控件的事件产生时，组态王7.0可以自动调用该函数。

 注意：

控件事件的关联函数必须在控件属性页中添加和编辑。

在相应事件的“关联函数”表格中单击鼠标右键，弹出快捷菜单，如图14.53所示。




图14.53 控件事件函数编辑快捷菜单

在快捷菜单中共有三个菜单项：

- ◇ **添加**：对控件事件添加一个函数。选择“添加”命令后，弹出“控件事件函数”命令语言编辑器，如图14.54所示。控件事件函数命令语言编辑器与通用的命令语言编辑器的使用方法相同，唯一特殊的是图中划圈的地方：

- “**事件**”标签后的内容表示当前编辑的控件事件的名称，如“Change ()”事件，如果事件带有参数，则会在事件名称内的括号中直接给出。
- “**函数声明**”：控件事件函数的类型都为“VOID”型，该项不可修改。控件事件函数同一般组态王7.0自定义函数相同，都需要定义一个函数名称，如“DatTimChange()”，函数声明中可以带有参数。

 注意：

函数声明中函数名称不管是否有参数，都要加小括号。命令语言编辑器和自定义函数的定义、使用方法请参见本手册第十章 创建命令语言程序。

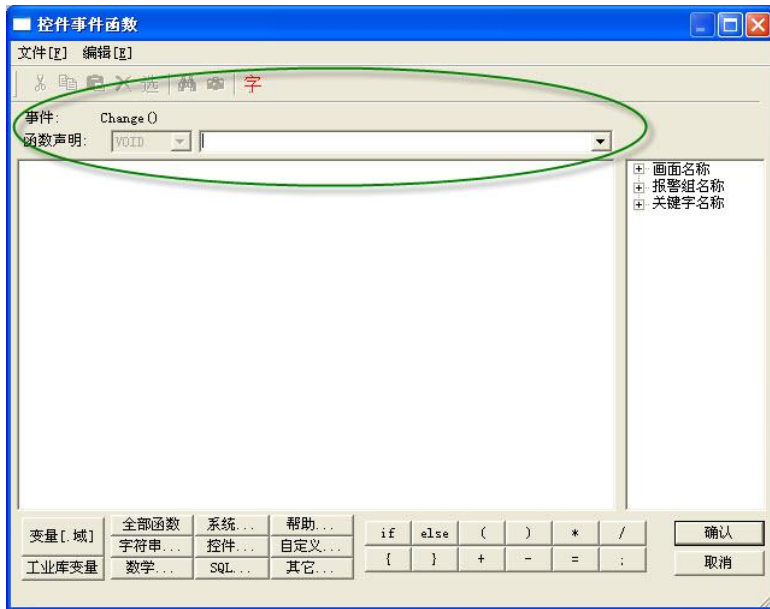


图14.54 控件事件函数编辑器

- ◇ **编辑：**编辑控件属性已关联的事件函数。当控件事件关联有事件函数时，单击鼠标右键，该项有效。
- ◇ **删除：**删除控件属性已关联的事件函数。当控件事件关联有事件函数时，单击鼠标右键，该项有效。



例1：在日历控件的“Change”事件中添加一个控件事件函数，用组态王7.0的变量读取控件的属性值。操作方法如下：

- 1、 创建日历控件，定义控件的动画连接属性中的名称，如“DatTim”。
- 2、 在控件的事件属性页中的“Chang”事件关联函数格中，单击右键，选择“添加”菜单命令。
- 3、 在函数声明的编辑框中输入函数名称“DatTimChange()”。
- 4、 在命令语言编辑区输入命令语言程序，如图14.55所示。

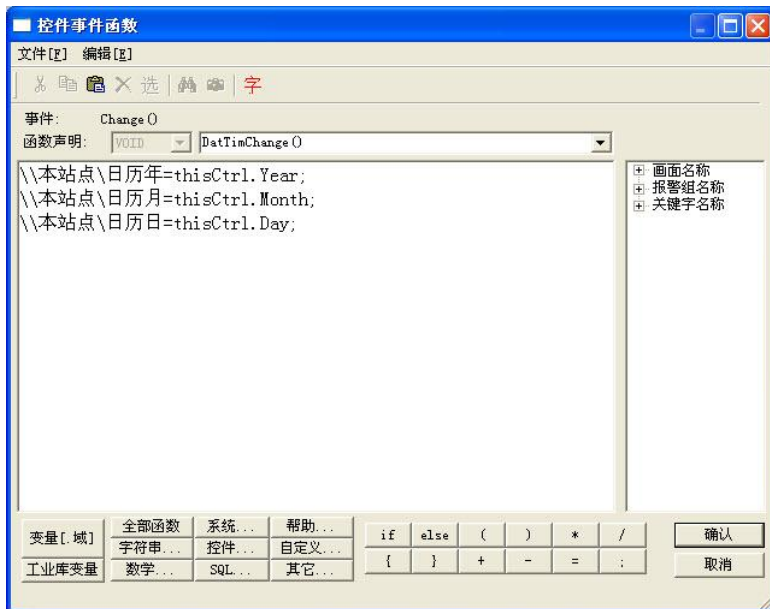


图14.55 控件事件函数1

5、在命令语言编辑区的最后一行输入 `ActiveXIsVisible(“DatTimChange”,0)`。运行时，单击控件的下拉按钮，在下拉框中选择设定的日期后，窗体消失。



注意：

在控件事件函数中引用控件的属性时：可以用“`thisCtrl.`”，再加上控件的属性。控件属性的引用区分大小写。



例2：

组态王7.0提供了一个表格控件“KVADODBGrid”。在工具箱上选择插入通用控件，在控件列表中选择“KVADODBGrid Class”，创建控件。定义控件的动画连接属性中控件名称为“KVADODBGridCtrl”。如图14.56所示，添加其“MouseDown”事件函数。定义当鼠标左键按下时，表格的行数加1；鼠标右键按下时，表格的列数加1。定义的命令语言程序如图14.57所示。



图14.56 添加控件的MouseDown事件函数

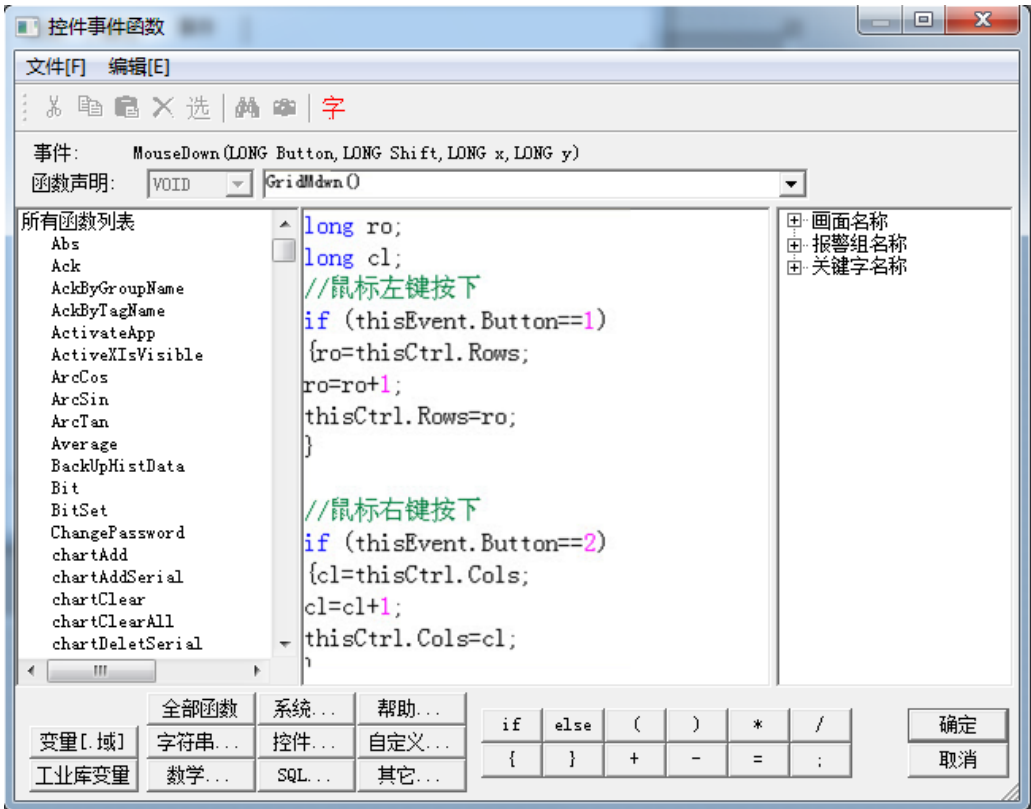



图14.57 控件的MouseDown事件函数

 注意:

在控件事件函数中引用控件事件的参数时: 用“thisEvent.”, 再加上控件事件的参数。参数名称的引用区分大小写。

14.3.2.4 在命令语言中调用 Active X 控件的属性和方法

控件与组态王7.0关联除了在控件属性中关联组态王7.0变量和定义控件事件函数外,

还可以在组态王7.0的任何命令语言程序中调用控件的属性和方法。如画面命令语言、动画连接命令语言、应用程序命令语言等。



注意：

当在运行系统中命令语言满足条件执行时，含有控件的画面在运行系统中应该是运行着的（不一定是当前画面），否则会出现调用控件属性或方法失败的情况，即得不到控件返回的值。

当控件创建成功，并定义了名称后，就可以在命令语言中调用其属性和方法了。下面以在按钮命令语言中调用日历控件的属性和方法为例介绍如何在命令语言中使用控件的属性和方法。



例1：通过一个按钮将日历控件的时间值赋值给组态王7.0变量。实现步骤为

- 1、按照14.3.2.1节所述，在画面上创建日历控件，定义动画连接属性，控件名称为“DatTimCtrl”。
- 2、在画面上创建按钮，双击按钮，弹出按钮动画连接对话框，选择“弹起时”命令语言，弹出命令语言编辑器，在命令语言编辑器的底部按钮区中单击“控件”按钮，弹出“控件属性和方法”对话框，如图14.58所示。该对话框由三个列表组成：

- **控件名称：**为下拉列表框，该列表框中列出当前工程中创建的所有Active X控件的名称。选择需要引用的控件的名称，如“DatTimCtrl”。“查

看类型”列表框将变为有效。

- **查看类型：**从列表框中选择查看的类型：控件属性或控件方法。
- **属性或方法：**当选择“查看类型”后，在属性或方法列表中将列出相应类型的列表。如选择控件属性，则在属性或方法列表中列出日历控件的所有属性。如图14.59所示。



图14.58 控件属性和方法对话框

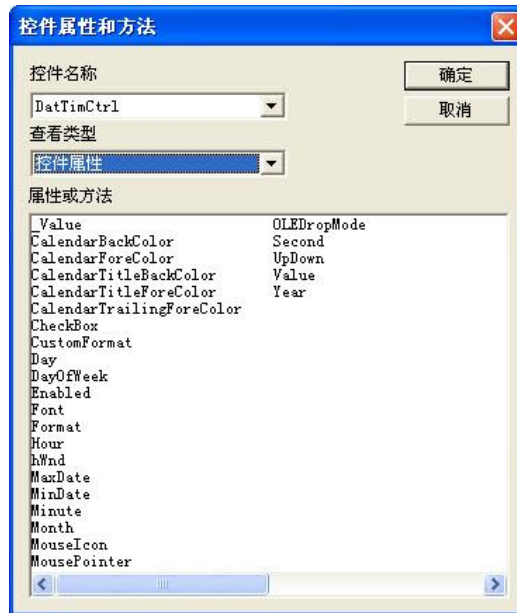


图14.59 日历控件的控件属性

3、 在属性列表中选择控件的属性，如“Year”，单击对话框的“确定”按钮，或在列表中直接双击该属性，控件属性和方法列表框自动关闭，控件的属性自动添加到命令语言编辑器中。如图14.60所示。

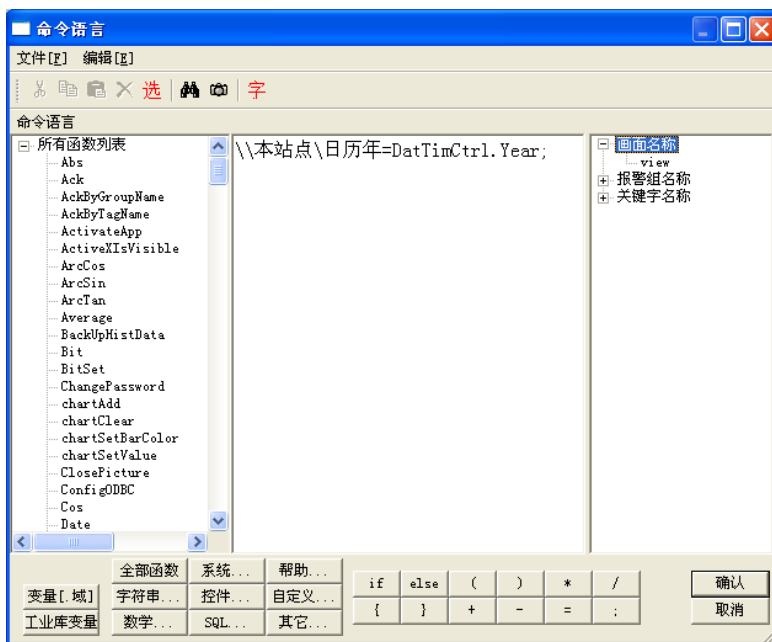


图14.60 在命令语言中引用控件的属性

 注意：

在命令语言里引用控件的属性时，应该使用控件的名称加控件属性名称，这与控件事件函数中使用有所不同。



例2：通过按钮命令语言调用控件的方法。操作步骤如下：

- 1、使用例1中的日历控件。在画面上创建按钮，在按钮命令语言编辑器中单击“控件”按钮。
- 2、在弹出的“控件属性和方法”对话框中，选择控件名称“DatTimCtrl”，在

查看类型中选择“控件方法”。如图14.61所示。如在本例中选择控件的关于——“AboutBox”。选择后，单击“确定”，关闭对话框。如图14.62所示，为在命令语言中调用的控件的方法。



图14.61 控件方法列表

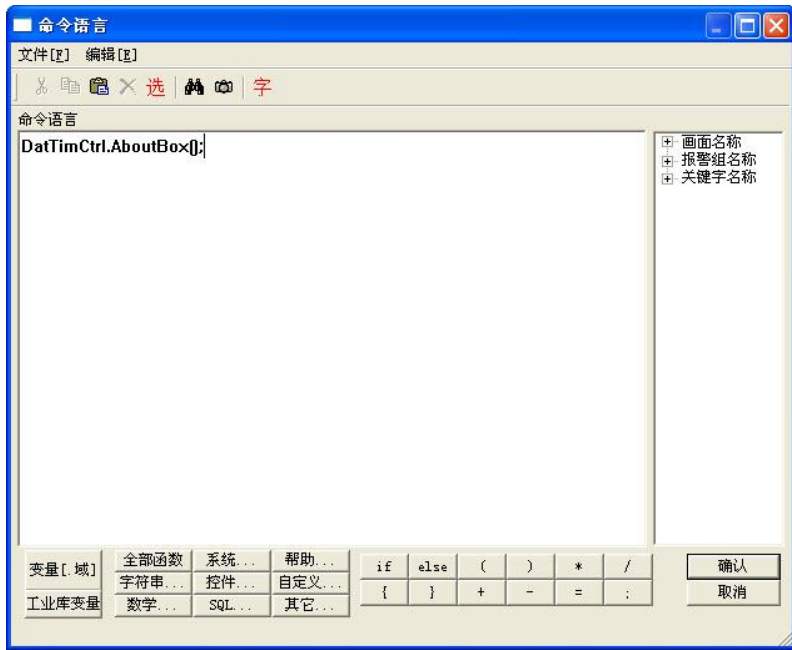


图14.62 调用控件的方法

按照上述例程中的方法，可以在组态王 7.0 的其它命令语言中调用控件的方法和属性。

14.3.2.5 一个 Active X 控件的使用例程

Active X 控件的使用极大的方便了用户，使用户很多在组态软件中无法使用的功能，通过 Active X 控件轻而易举的实现了。下面详细举例说明 Active X 控件在组态王 7.0 中如何使用。

下面使用 Microsoft Chart Control 控件为例说明如何利用该控件作一个步骤图。

1. 新建画面，单击工具箱上的“插入通用控件”按钮，或选择菜单“编辑/插入通用控

件”命令，弹出插入控件对话框，如下图 14.63 所示。

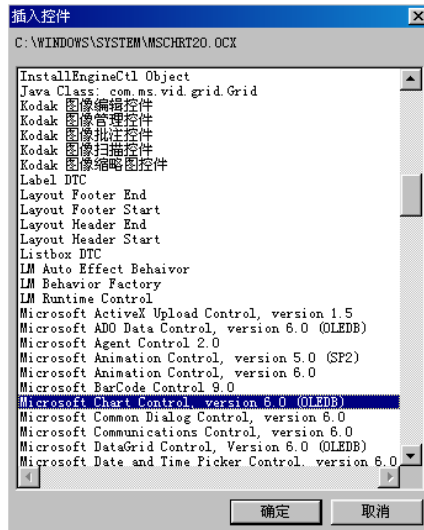


图 14.63 通用控件菜单

2. 插入控件：在对话框中选择“Microsoft Chart Control”项，然后单击“确定”按钮。对话框自动消失，鼠标箭头变为小“十”字型，在画面中绘制图表的区域按下鼠标左键，然后拖动鼠标，拖动出的矩形框为图表的区域，松开鼠标键后，图表创建成功，显示在画面上，如图 14.64 所示。

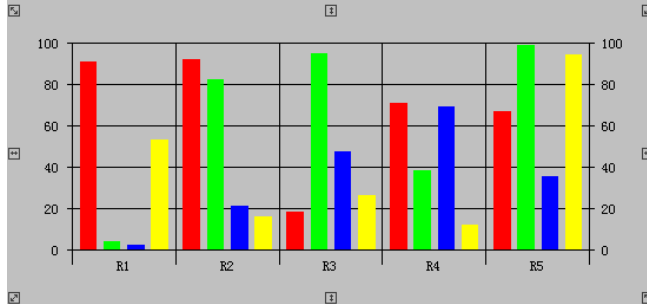


图 14.64 插入控件

3. 设置控件固有属性：在控件上单击右键，在弹出的快捷菜单上选择“控件属性”命令，弹出控件固有属性对话框，如下图 14.65 所示。

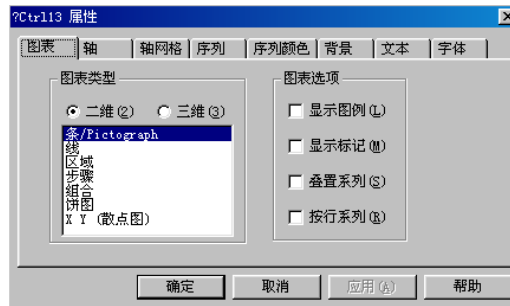


图 14.65 控件属性 1

- ✧ 在图表属性页中的图表类型中选择“二维”、“步骤”选项。在图表选项中选择“显示图例”选项，该选项选中后，在图表的右侧将出现关于图表中各种颜色与曲线对应情况的标识。



图 14.66 控件属性 2

- ◇ 在文本属性页中定义图表各个标签的标题。在“属性名称”列表框中选择对象，如标题，为图表的标题，在“文本”编辑框中输入标题名称，如“步骤图”。然后可以选择“X轴”、“Y轴”定义其标签名称。然后可以通过“对齐”、“方向”选项分别定义各个标签的对齐方式和方向。
- ◇ 在字体属性页中定义各个标签的字体及字体颜色。定义完成的图表如下图 14.67 所示。

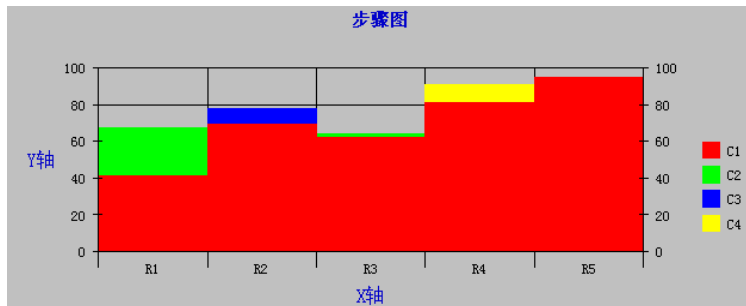


图 14.67 定义后的控件

4. 设置控件名称和函数：在组态王 7.0 中每一个控件必须要有一个唯一的名称。双击该图表控件，弹出动画连接属性对话框，如下图 14.68 所示。

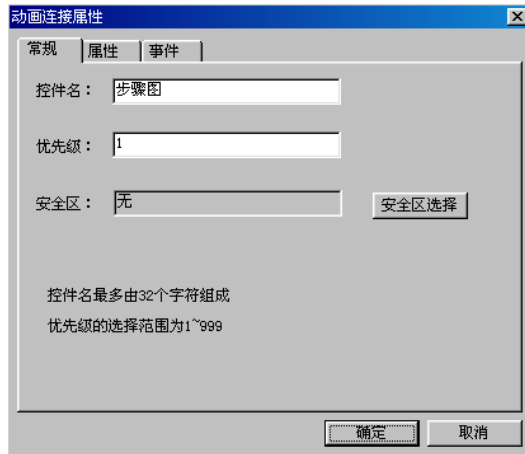


图 14.68 动画连接属性

- ◇ 常规属性页中，在“控件名”编辑框中输入控件的名称，如“步骤图”。
- ◇ 在事件属性页中定义一个函数：当用户双击该图表时，弹出图表控件的版本信息。双击“Db1Click”对应的“关联函数”，弹出控件事件函数对话框，定义函数的名称为“hanshu()”，单击“控件”按钮，在弹出的控件属性和方法对话框中选择“步骤图”的方法“AboutBox”方法。如下图 14.69 所示。

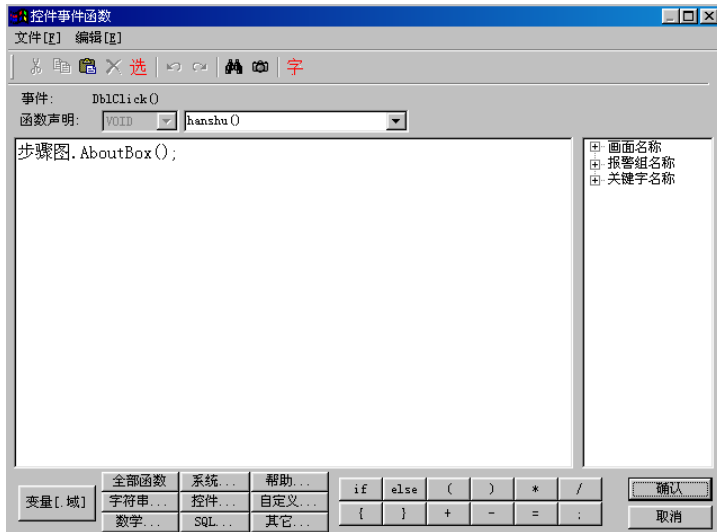


图 14.69 控件事件函数

5. 为控件曲线赋值：设置画面命令语言，初始化控件，和在运行过程中为各条曲线赋值。
假设在组态王 7.0 中的变量：实数 1、实数 2、实数 3 代表三个步骤的数值。
定义画面命令语言 显示时 如下图 14.70 所示。

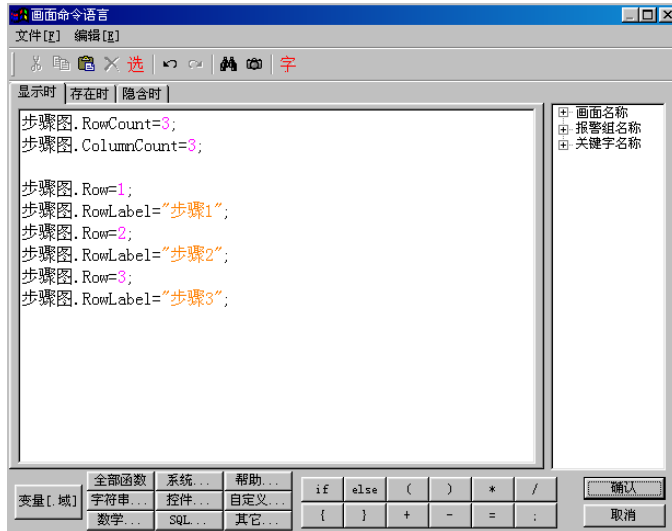


图 14.70 定义显示时函数

初始化图表为三行三列。

定义画面命令语言 存在时： 1000 毫秒

```
步骤图.Column=1;
```

```
步骤图.Row=1;
```

```
//为第一行第一列的第一条曲线赋值
```

```
步骤图.Data=StrFromReal( \\本站点\实型 1, 2, "f" );
```

```
步骤图.Column=1;
```

```
步骤图.Row=2;
```

```
//为第一行第一列的第二条曲线赋值
```

```
步骤图.Data=StrFromReal( \\本站点\实型 2, 2, "f" );
```

```
步骤图.Column=1;

步骤图.Row=3;

//为第一行第一列的第三条曲线赋值

步骤图.Data=StrFromReal( \\本站点\实型 3, 2, "f" );

步骤图.Column=2;

步骤图.Row=1;

步骤图.Data=StrFromReal( \\本站点\实型 1*1.2, 2, "f" );

步骤图.Column=2;

步骤图.Row=2;

步骤图.Data=StrFromReal( \\本站点\实型 2*1.2, 2, "f" );

步骤图.Column=2;

步骤图.Row=3;

步骤图.Data=StrFromReal( \\本站点\实型 3*1.2, 2, "f" );

步骤图.Column=3;

步骤图.Row=1;

步骤图.Data=StrFromReal( \\本站点\实型 1*1.8, 2, "f" );

步骤图.Column=3;

步骤图.Row=2;

步骤图.Data=StrFromReal( \\本站点\实型 2*1.8, 2, "f" );
```

步骤图.Column=3;

步骤图.Row=3;

步骤图.Data=StrFromReal(\\本站点\实型 3*1.8, 2, "f");

保存画面后，切换到组态王 7.0 运行系统，运行结果如下图 14.71 所示。

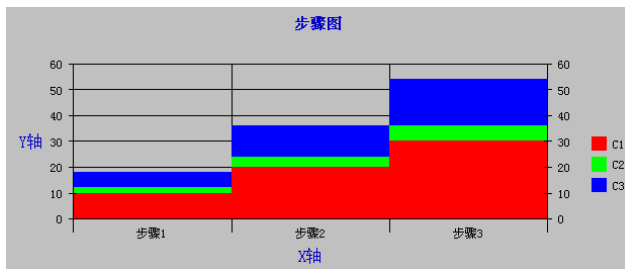


图 14.71 运行控件

双击该控件，弹出控件的“关于”对话框，如图 14.72 所示。



图 14.72 控件信息

使用 Microsoft Chart Control 控件可以制作如棒图、散点图、曲线图、分布图等图表。用户可以按照例中的方法使用。

Active X 控件按照其功能的不同，其属性、方法、事件有多有少，复杂程度也不同。建议用户在使用 Active X 控件前最好先参阅一下控件厂家的说明、帮助等，如属于 Microsoft 的控件，可以参阅 MSDN。

14.3.3 组态王 7.0 提供的 Video 视频控件

Video 视频控件由组态王 7.0 提供的支持视频显示的 Active X 控件。用户可通过该控件与各种型号的视频设备相连接。Video 视频控件的使用步骤为：

14.3.3.1 画面上创建 Video 视频控件

操作步骤如下：

- ◇ 第一步：单击工具箱中的“插入通用控件”按钮或选择菜单命令“编辑\插入通用控件”，则弹出“插入控件”对话框。
- ◇ 第二步：在“插入控件”对话框内选择 Video Control，如下图 14.73 所示。

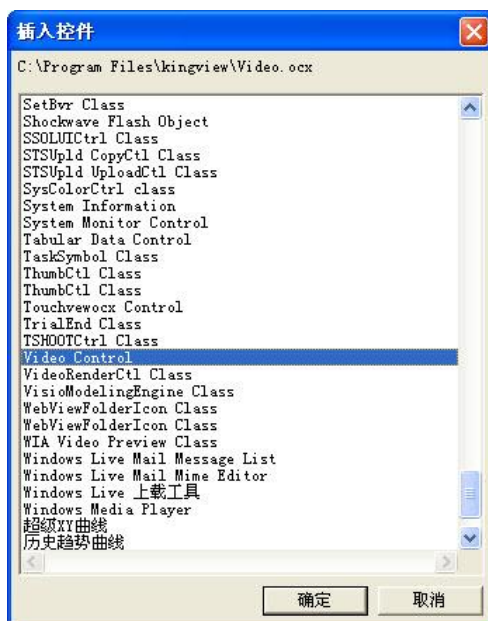


图 14.73 插入控件列表

- ◇ 第三步：用鼠标左键单击“确定”按钮，鼠标变成十字形。然后在画面上画一个矩形框，Video 视频控件就放到画面上了。可以任意移动、缩放 Video 视频控件，如同处理一个单元图素一样。

14.3.3.2 设置 Video 视频控件的属性

用左键双击 Video 控件，弹出“动画连接属性”对话框，或者用右键单击 Video 控件选择菜单命令“动画连接”，则也弹出“动画连接属性”对话框如下图 14.74 所示：

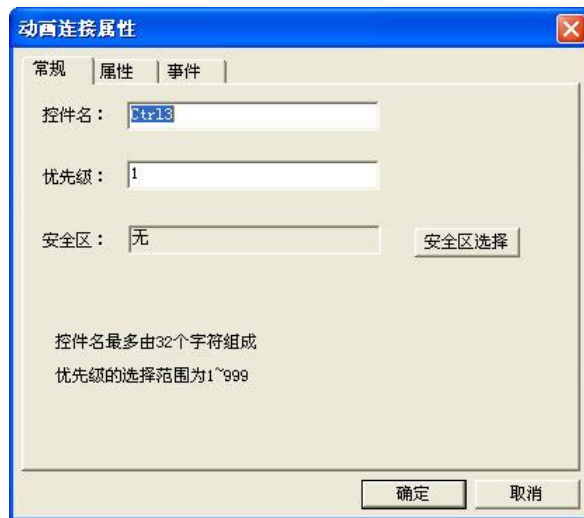


图 14.74 设置控件属性

- **控件名**：一个 Video 视频控件对应一个控件名称，而且是唯一的，不能重复命名，控件名由工程人员设定，例如设为“监控”。
- **优先级**：用于设置访问该控件的操作优先级级别，优先级级别从 0~999。
- **安全区**：单击右侧的“安全区选择”按钮，选择该控件的操作安全区。

14.3.3.3 Video 视频控件的使用

Video 控件的使用主要依靠调用控件提供的方法来实现。控件提供的方法主要有以下几种：

AboutBox ()
CapVideoToAVI (STRING lpszAVIName)
GetDevNum ()
OpenAVI (STRING lpszFileName)
OpenVideo (LONG nResIndex)
ReadPort (LONG nAddr)
SaveVideoFrame (STRING lpszPicName)
SaveVideoFrmeToRect (STRING lpszPicName)
SetFrameRect ()
StopCapVideo ()
VideoCompression ()
VideoDisplay ()
VideoFormat ()
VideoSource ()
WritePort (LONG nAddr, LONG nData)

下面以一个具体例子说明 Video 视频控件的使用及操作步骤。



例：

在大厅内安装一个探头, 视频采集卡插在计算机上, 在屏幕上设置一个 Video 视频控件, 通过打开 Video 视频控件, 则探头所拍摄下来的内容全部在屏幕上显示出来, 并且能通过提供的函数对画面做相应处理。在组态王 7.0 开发系统中建立的视频控件和操作按钮如下图 14.75 所示。



图 14.75 创建视频控件

分别对各个按钮建立命令语言连接, 建立命令语言连接的方式为: 双击按钮, 选择“按下”或“弹起”时, 弹出命令语言界面, 点击“控件”按钮, 弹出“控件属性和方法”对话框, 控件名称选择定义的“监控”, 在属性或方法中会列出所有控件方法, 如下图 14.76 所示。



图 14.76 控件方法列表

调用相应的方法即可完成对视频控件的所有操作。

下面对建立的操作按钮做详细说明：

ON：打开视频采集

BOOL OpenVideo(short nResIndex)

参数：nResIndex 视频设备的设备号，该值的范围为 0—9；

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控.OpenVideo(0)

OFF：关闭视频采集

BOOL CloseVideo()

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. CloseVideo()

设置视频源：弹出设置视频源对话框

BOOL VideoSource()

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. VideoSource()

设置视频压缩：弹出设置视频压缩对话框

BOOL VideoCompression()

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. VideoCompression()

拍照：单帧保存视频图像到一个 BMP 文件

BOOL SaveVideoFrame(LPCTSTR lpszPicName)

参数：lpszPicName，要保存的 BMP 文件名

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. SaveVideoFrame(“c:\my documents\01.bmp”)

摄像：把视频录像到一个 AVI 文件

BOOL CapVideoToAVI(LPCTSTR lpszAVIName)

参数：lpszAVIName，录像后保存的 AVI 文件名

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. CapVideoToAVI(“c:\my documents\01.avi”)

停止摄像：停止摄像

BOOL StopCapVideo()

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. StopCapVideo()

回放：打开一个 AVI 文件

BOOL OpenAVI(LPCTSTR lpszFileName)

参数：lpszFileName，要打开的 AVI 文件名

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. OpenAVI (“c:\my documents\01.avi”)

设置抓拍位置：弹出设置抓拍区域对话框

BOOL SetFrameRect()

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. SetFrameRect()


抓拍：抓拍视频图像到一个 BMP 文件

BOOL SaveVideoFrameToRect(LPCTSTR lpszPicName)

参数：lpszPicName，抓拍保存的 BMP 文件名

返回值：成功返回 TRUE，失败返回 FALSE

例如：监控. SaveVideoFrameToRect (“c:\my documents\01.bmp”)

 注意:

抓拍前可利用 `SetFrameRect()` 弹出的对话框来确定抓拍范围，或者直接用鼠标在画面上选取范围，用鼠表直接选取抓拍范围时，没有任何提示信息。

设置视频显示：弹出设置视频显示对话框

`BOOL VideoDisplay()`

返回值：成功返回 `TRUE`，失败返回 `FALSE`

例如：监控. `VideoDisplay()`

设置视频格式：弹出设置视频格式对话框

`BOOL VideoFormat()`

返回值：成功返回 `TRUE`，失败返回 `FALSE`

例如：监控. `VideoFormat()`

另外，该控件还具有以下方法：

获取视频设备个数：

`Short GetDevNum()`

返回值：视频设备个数

往端口地址写入控制码：

`Long WritePort(short nAddr, short nData)`

参数：nAddr 端口地址

nData 控制码

返回值：返回写入的控制码

备注：如果操作系统是 Windows 98，则直接调用；如果是 Windows NT / Windows 2000 则需要安装组态王 7.0 的 NT 底层驱动。请从光盘下的 Value pack\Video\NT 底层驱动中安装。

从端口地址读数据：

Long ReadPort(short nAddr)

参数：nAddr，端口地址

返回值：返回从端口地址读取的数据

备注：如果操作系统是 Windows 98，则直接调用；如果是 Windows NT / Windows 2000 则需要安装组态王 7.0 的 NT 底层驱动。请从光盘下的 Value pack\Video\NT 底层驱动中安装。

在开发中设置好后进入运行系统，如果是第一次进入运行系统，打开视频后，必须设置视频源。

14.3.4 开放型数据库访问控件

在工程中经常需要访问开放型数据库中的大量数据，如果通过 SQL 函数编程查询，则费时费力，不易使用。针对这种情况，组态王 7.0 提供了一个通过 ADO 访问开放型数据库中数据的 Active X 控件——KVADODBGrid。通过该控件，在组态王 7.0 画面中用户可以很方便的访问数据库、编辑数据库。可以通过数据库查询窗口对数据库中的数据进行查询，也可以用控件的统计函数计算出控件中数据的最大、最小值和平均值等。该控件具有以下

功能:

1. 既可以用 ODBC 连接数据库, 也可以用 ADO 连接数据库。
 2. 显示数据库表中的数据, 可选择显示字段, 编辑查询条件。
 3. 开发状态可设置显示表头。
 4. 在开发和运行状态均可选择是否显示视图 (虚拟表)。
 5. 运行状态可编辑表格中数据。
 6. 运行状态可通过界面或脚本动态设置数据源、数据库、数据表等参数。
 7. 运行状态可通过界面或脚本动态选择字段、编辑查询条件等操作。
 8. 可打印表格中显示的数据, 可进行横向和纵向分页。
 9. 可通过查询窗口帮助用户生成查询语句, 而无需用户输入。
-



注意:

如果用户需要用 ODBC 数据源连接数据库, 在使用该控件前, 需要在系统控制面板中定义一个 ODBC 数据源, 连接到所要查询的数据库。如果通过 ADO 方式直接连接到数据库, 不需要创建 ODBC 数据源。

14.3.4.1 创建 KVADODBGrid 控件

单击工具箱中的“插入通用控件”按钮或选择菜单命令“编辑\插入通用控件”, 则弹出“插入控件”对话框。在“插入控件”对话框内选择 KVADODBGrid Class, 如图 14.77 所示。用鼠标左键单击“确定”按钮, 关闭对话框。鼠标变成十字形。然后在画面上适当位置按下鼠标左键并进行拖动, 画一个矩形框, 放开鼠标左键, KVADODBGrid 控件就放到

画面上了。可以任意移动、缩放控件，如同处理一个单元图素一样。如图 14.78 所示。

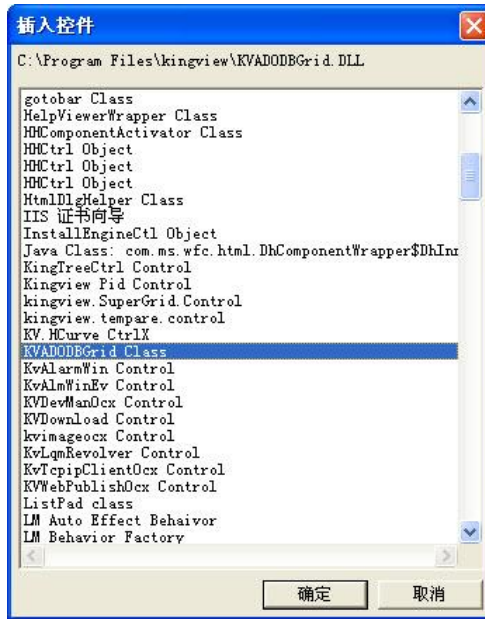


图 14.77 选择 KVADODBGrid 控件

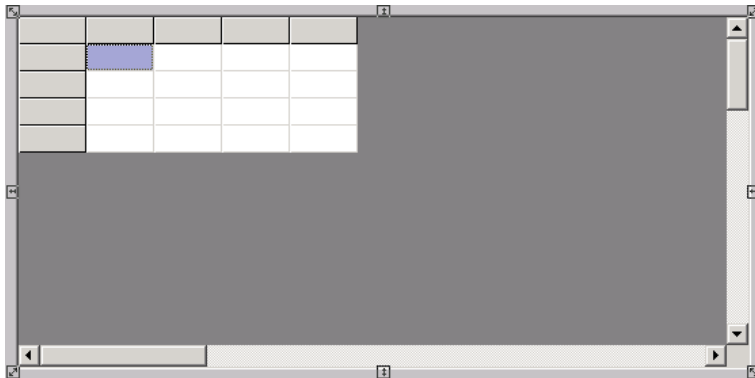


图 14.78 创建后的 KVADODBGrid 控件

14.3.4.2 设置 KVADODBGrid 控件的固有属性

选择控件，单击右键，在弹出的菜单中选择“控件属性”。弹出控件固有属性对话框，如下图 14.79 所示。

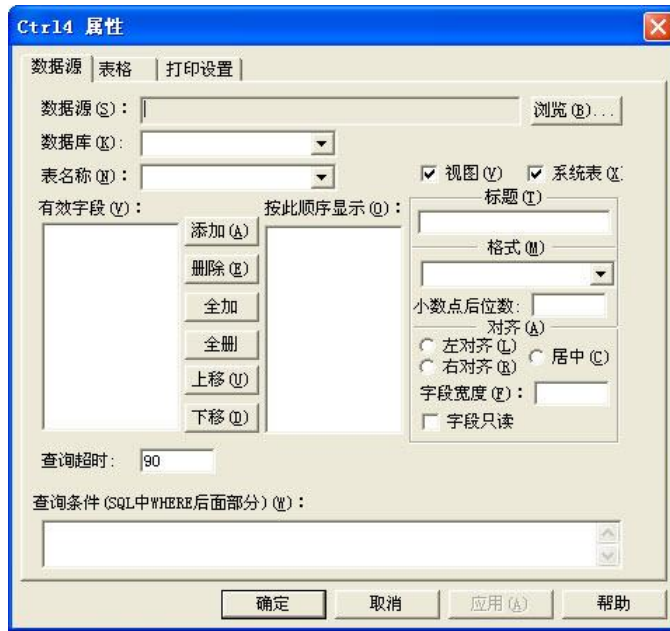


图 14.79 KVADODBGrid 控件属性

1、数据源属性页

该属性页主要定义控件连接的数据源、数据库和数据表，选择要显示的数据表中的字段，对每个字段在控件中显示的标题、格式、对齐方式、小数点位数（如果是数值型的话）进行设置。可以在这里直接指定查询的条件，也可以在运行时修改控件属性以改变查询条件。视图是由多张表中不同字段组成的虚拟表格，视图选项是用来显示数据库中的视图，

当视图选项选中时，视图的名字会出现在表名称的列表中以供选择，当视图选项取消时，视图的名字将会从表名称的列表中消失。

单击属性页右上角的“浏览”按钮，会弹出“数据连接属性”对话框，如图 14.80 所示。在列表中显示了各种数据连接引擎。（注：下面说明中使用的示例数据库为组态王 7.0 演示工程数据库：……Kingview\Example\Kingdemo3\SQL 数据库.mdb）



图 14.80 KVADODBGrid 控件选择连接引擎

如果要使用 ODBC 连接，直接选择“Microsoft OLE DB Provider for ODBC Drivers”，点击“下一步”按钮，定义“连接”属性。可以选择“使用数据源名称”选项来选择已经定义的 ODBC 数据源名称。也可以选择“使用连接字符串”，输入连接字符串。也可以直接选择“编译”按钮对连接字符串进行选择和自动编译。

以上连接定义方法为数据库连接通用方法，具体可以参照系统的相关帮助。

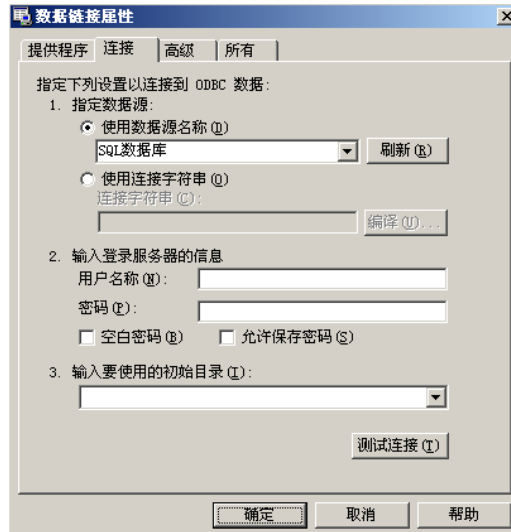


图 14.81 定义 ODBC 连接

该控件还支持 ADO 方式数据访问，可以使用其它数据连接引擎。在图 14.80 上选择相应的引擎名称，如“Microsoft Jet 4.0 OLE DB Provider”。

点击“下一步”按钮，进入“连接”属性页，如下图 14.82 所示。

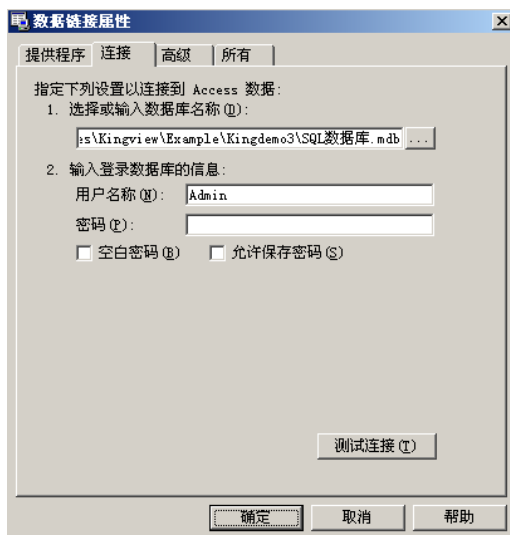


图 14.82 定义 JET 连接

选择要连接的数据库名称，定义完成后，控件即可通过 ADO 方式使用 JET 引擎访问定义的数据库。

选择完数据源后，点击“确定”按钮返回“数据源”属性页。如下图所示。从“表名称”列表框中选择要查询数据的数据表名称。在“有效字段”列表框中系统会自动列出所选表的所有有效字段。通过按钮“添加”、“删除”、“全加”和“全删”，可以选择在控件中显示哪些字段的内容。所选择的要显示的字段列在右侧的列表中。选择“上移”、“下移”按钮来对字段显示的顺序进行排序。

在要显示的字段中选定某个字段，可以对其在控件标题栏中的显示内容和字段对应的列的显示格式进行设置。各项含义如下：

◇ **标题：**是指字段在控件标题栏中具体显示的名称。

- ◇ **格式：**是定义该字段在控件对应列中显示的格式，只限于列表中的四项。
- ◇ **对齐：**定义字段列对应单元格中显示的对齐方式。
- ◇ **小数点后位数：**如果是数字字段，定义显示的最多小数位数。
- ◇ **字段宽度：**定义该字段在单元格中最多显示的字符数。
- ◇ **字段只读：**如果选择该项，则不能通过调用控件的方法来修改数据库中的值。
- ◇ **查询条件：**定义默认的数据库查询条件。只输入 SELECT……WHERE 语句之后的部分。如查询“水温”大于 20 的信息，则在该文本框中输入：水温 > 20



图 14.83 定义显示内容

- ◇ **查询超时：**设置查询超时时间。

2、表格属性页

表格属性页用来定义表格行列、边界等属性。如下图 14.84 所示。



图 14.84 控件的表格属性

各属性含义如下：

- ◇ **控件可见**：设置控件在运行时是否可见。
- ◇ **可调整行高**：设置运行时可以调整控件每行显示高度。
- ◇ **可调整列宽**：设置运行时可以调整控件每列显示宽度。
- ◇ **自动调整列宽**：如果一次显示的字段较多，有可能在一屏显示不了。选择该选项，系统运行时，查询数据结束后，会自动调整列宽以适应在一屏中显示所有

字段。在字段较少，可以一屏显示时，该选择是否选择都不起作用。

- ◇ **数据库内容可修改**：选择该项，在运行中修改控件单元格中的数据时，同时会修改数据库中对应的数据。请慎重使用该选项。
- ◇ **画水平格线**：设置控件在运行时表格是否画水平格线。
- ◇ **画垂直格线**：设置控件在运行时表格是否画垂直格线。
- ◇ **边界类型**：可选“无”、“客户边界”、“静态边界”、“模式边界”四种边界类型。
- ◇ **固定行数**：如固定行数大于 1，则第一行显示选择字段的标题，其它固定行为空且不可编辑。
- ◇ **固定列数**：若固定列数大于 0，那么运行时固定列也显示数据，并不为空。此列最好显示如“序号”等具有排序性质的字段，若数据集中没有排序性质字段，可设置“固定列数”为 0。

3、打印设置属性页

打印设置属性页用例设置控件的打印相关内容，如标题、页眉、页脚的内容和位置，设置页码的格式和显示位置，对于分页打印时的设置，已经打印方式等。如下图 14.85 所示。

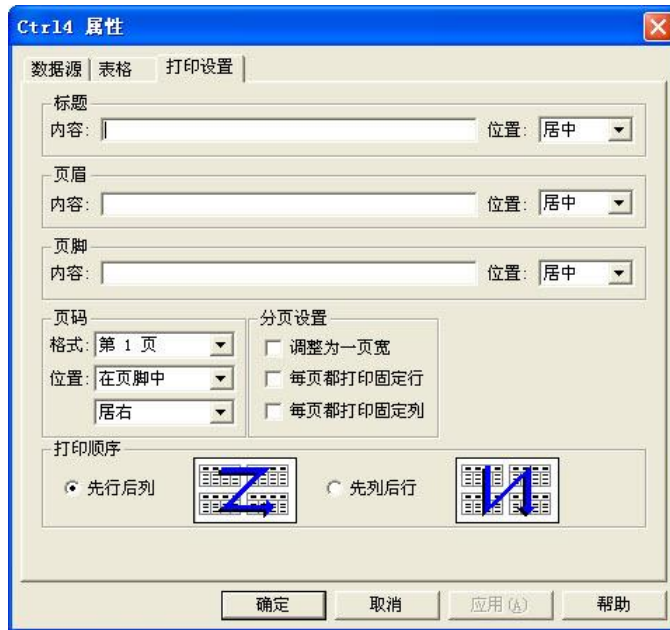


图 14.85 控件的打印设置属性

14.3.4.3 设置 KVADODBGrid 控件的动画连接属性

要使用 KVADODBGrid 控件，必须设置其名称属性等动画连接属性。选择控件，双击鼠标左键，弹出控件的动画连接属性对话框，如图 14.86 所示。控件的动画连接属性共分为三页：“常规”属性页、“属性”列表和“事件”列表页。

在常规属性页中定义控件的名称和操作权限、安全区。

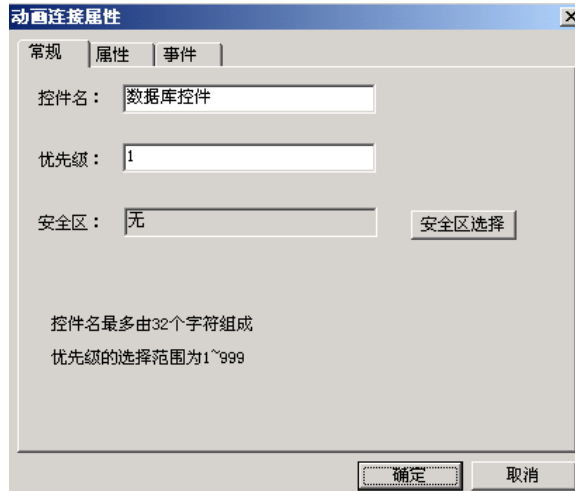


图 14.86 控件的动画连接属性

14.3.4.4 KVADODDBGrid 控件的使用

在使用 KVADODDBGrid 控件时，可以调用控件提供的丰富的属性、方法和事件函数来对控件进行操作。

14.3.4.4.1 KVADODDBGrid 控件属性

下面为 KVADODDBGrid 控件的属性列表和属性含义。

序号	属 性	说 明
1	Rows	获取当前表格的总行数
2	Cols	获取当前表格的总列数
3	FixedRows	获取或设置表格固定行的行数。运行时设置此属性后，需要调用 FetchData () 方法重新得到数

		据集
4	FixedCols	获取或设置表格固定行的列数。运行时设置此属性后，需要调用 FetchData（）方法重新得到数据集
5	Row	获取或设置当前所选单元格（焦点所在单元格）的行（从 0 开始）
6	Col	获取或设置当前所选单元格（焦点所在单元格）的列（从 0 开始）
7	Editable	数据单元格是否可编辑、修改。置“0”不可编辑，置“1”可编辑。
8	CellAccordWithDB	数据库内容是否可以修改。置“1”可以修改，置“0”不可以修改。
9	Table	获取或设置要检索的数据表名称。
10	Where	设置查询条件，如果不需要任何条件，则可以设置为空。
11	ADConnStr	配置连接数据库连接字符串。用户用此属性进行连接数据库的字符串的配置是很不方便和易出错的，推荐不使用，进行配置推荐使用 SetConnectParam 方法进行设置。不过与使用

		SetConnectParam 方法通过 ODBC 进行数据库的连接的方法进行比较，系统运行性能要好些。
12	CurCellText	得到当前所选单元格文本的值。
13	AutoAdjustColWidth	是否自动调整列宽。置“0”表示不自动调整，置“1”自动调整。
14	ResizeCols	是否允许调整列宽。置“0”不可调整列宽，置“1”允许调整列宽。
15	ResizeRows	是否允许调整行高。置“0”不可调整行高，置“1”允许调整行高。
16	Visible	控件是否可见。置“0”控件不可见，置“1”控件可见。
17	HorzLine	画水平格线选择。置“0”不画线，置“1”画线。
18	VertLine	画垂直格线选择。置“0”不画线，置“1”画线。
19	PrintTitle	打印时页标题的内容。
20	PrintHeader	打印时页眉的内容。
21	PrintFooter	打印时页脚的内容。
22	PrintOrder	打印顺序选择，0：先行后列，1：先列后行。
23	ViewEnabled	设置“视图”选项是否有效，为“1”有效，为“0”无效。

下面举例说明控件属性的使用方法。

1. FixedRows 属性:

说明: 得到或设置表格固定行的行数, 运行时设置此属性后, 需要调用

FetchData () 方法重新得到数据集;

读写属性: 读写;

关联变量: 整型变量;

使用方法:

得到 FixedRows 属性到组态王 7.0 整型变量 nFixedRows:

```
nFixedRows = Ctrl.FixedRows;
```

使用组态王 7.0 整型变量 nFixedRows 设置 FixedRows 属性;

```
Ctrl.FixedRows = nFixedRows;
```

使用整型常量设置 FixedRows 属性;

```
Ctrl.FixedRows = 2;
```



例:

```
nFixedRows = Ctrl.FixedRows; // 得到当前控件固定行数,
```

```
nFixedRows 为组态王 7.0 整型变量;
```

```
nFixedRows = 2;
```

```
Ctrl.FixedRows = nFixedRows; //设置固定行数为 2;
```

```
Ctrl.FetchData (); //刷新数据集
```

2. Row 属性:

说明: 得到或设置当前所选单元格 (焦点所在单元格) 的行 (从 0 开始);

读写属性: 读写;

关联变量: 整型变量;

使用方法:

得到 Row 属性到组态王 7.0 整型变量 nRow:

```
nRow = Ctrl.Row;
```

使用组态王 7.0 整型变量 nRow 设置 Row 属性:

```
Ctrl.Row = nRow;
```

使用整型常量设置 Row 属性:

```
Ctrl.Row = 5;
```

3. CellAccordWithDB 属性:

说明: 数据库内容是否可以修改, 为“1”可以修改, 为“0”不可以修改;

设置此属性只在本次运行时有效, 并不修改“开发系统—控件属性—表格—是否修改数据库内容”选项。

读写属性: 读写;

关联变量: 离散型变量;



例:

```
bCellAccordWithDB = Ctrl.CellAccordWithDB;
```

```
//得到当前表格“修改数据库内容”属性值；  
bCellAccordWithDB = ! bCellAccordWithDB;  
  
//此属性值取反；  
Ctrl.CellAccordWithDB = bCellAccordWithDB;  
  
//重新设置此属性；
```

4. Where 属性:

说明：设置查询条件，如果不需要任何条件，则可以设置为空。

读写属性：读写；

关联变量：字符串变量；



例：

```
strWhere = Ctrl.Where; //得到当前控件查询条件，strWhere 为  
组态王 7.0 字符串变量；  
  
strWhere = “R1 > 800”;  
Ctrl.Where = strWhere; //重新设置查询条件  
Ctrl.FetchData (); //按新设置的查询条件显示数据集到控  
件；
```

14.3.4.4.2 KVADODBGrid 控件方法

1. FetchData () 方法:

功能：

执行数据查询，并将查询到的数据集填充到控件中。

使用方法：

```
Ctrl.FetchData ();
```



例如：

```
strWhere = "R1 > 800";
```

```
Ctrl.Where = strWhere; //重新设置查询条件
```

```
Ctrl.FetchData (); //按新设置的查询条件刷新数据集;
```

2. Print () 方法：

功能：

执行表格打印。

使用方法：

```
Ctrl.Print ();
```

3. RefreshData () 方法：

功能：

按照上次查询的条件，重新刷新数据集。

使用方法：

```
Ctrl.RefreshData ();
```

4. RemoveAllData () 方法：

功能：

删除当前控件显示的所有数据。

使用方法：

```
Ctrl. RemoveAllData ();
```

5. SaveToCSV (STRING bstrCSV) 方法：

功能：

将当前控件中显示的所有数据保存成指定的 CSV 格式的文件。

参数：

STRING bstrCSV：保存路径和文件名。

使用方法：

Ctrl.SaveToCsv (“f:\Temp\a1.csv”), 其中 “f:\Temp\a1.csv” 为保存路径和文件名，可以用组态王 7.0 字符串变量，Ctrl 为控件名称。

6. ScrollToBottom () 方法：

功能：

鼠标焦点定位到控件的最底部。

使用方法：

```
Ctrl. ScrollToBottom ();
```

7. GetCertainCellText (LONG lRow, LONG lCol) 方法：

功能：

获取指定单元格中的文本。

参数：

LONG lRow: 指定单元格所在行(从 0 开始)。

LONG lCol: 指定单元格所在列(以 0 开始)。



例如:

```
LONG lRow:
```

```
LONG lCol:
```

```
lRow = 2;
```

```
lCol = 3;
```

```
strCurSelText = Ctrl. GetCertainCellText (lRow, lCol);
```

```
//获取第2行、第3列单元格中的文本到字符串变量strCurSelText。
```

8. AddAllFields (LONG lFieldWidth, BOOL nReadOnly) 方法:

功能:

添加当前数据表的所有字段到控件中。

参数:

LONG lFieldWidth 添加的这些字段的字段宽度, 字段宽度必须大于 0;

BOOL nReadOnly 所添加的字段是否为只读 (1 表示字段只读, 0 表示字段可读写)。

说明:

若需要按设置的列宽显示数据, 应在控件属性中去掉“自动调整列宽”选项。



例如：

```
Ctrl.SetConnectParam (“MyTestDSN” , “ ” , “ ” , “ ”);  
                                     //连接MyTestDSN数据源 (Access数据库);  
  
Ctrl.Table = “KingTable” ; //选择KingTable数据表;  
  
Ctrl.AddAllField(75, 1); //增加此表中所有字段到控件;  
  
Ctrl.FetchData (); //刷新数据集;
```

9. AddField (STRING bstrName, STRING bstrTitle, LONG lFieldWidth, BOOL nReadOnly) 方法：

功能：

根据指定的字段名添加字段到控件中。

参数：

STRING bstrName: 指定添加的字段的字段名，必须为有效字段名称；

STRING bstrTitle: 被添加字段的实际显示名称，若为空，则取缺省的名称即该字段名作为显示名称；

LONG lFieldWidth: 字段的宽度，字段宽度必须大于0；

BOOL nReadOnly: 所添加的字段是否为只读（1 表示字段只读，0 表示字段可读写）。

说明：

如果添加的字段已经存在，则方法调用失败。

若需要按设置的列宽显示数据，应在控件属性中去掉“自动调整列宽”选项



例如：

```
Ctrl.SetConnectParam (“MyTestDSN”, “”, “”, “”);  
//连接MyTestDSN数据源 (Access数据库);  
  
Ctrl.Table = “KingTable”;  
//选择KingTable数据表;  
  
Ctrl.AddField (“R1”, “温度”, 75, 1);  
//增加此表中字段名为R1的字段, 显示字段名称为  
“温度”, 列宽75, 只读;  
  
Ctrl.FetchData (); //刷新数据集;
```

10. AddFieldNo (LONG lCol, STRING bstrTitle, LONG lFieldWidth, BOOL nReadOnly)

方法：

功能：

根据指定的字段所在列号（以 0 开始）添加字段到控件中。

参数：

LONG lCol： 指定添加的字段在数据表中以 0 开始的列号，列号必须为有效列号；

STRING bstrTitle： 被添加字段的实际显示名称，若为空，则取缺省的名

称即该字段名作为显示名称；

LONG lFieldWidth: 字段的宽度，字段宽度必须大于0；

BOOL nReadOnly 所添加的字段是否为只读（1 表示字段只读，0 表示字段可读写）。

说明：

添加的字段已经存在，则方法调用失败。

若需要按设置的列宽显示数据，应在控件属性中去掉“自动调整列宽”选项。



例如：

```
Ctrl.SetConnectParam (“MyTestDSN”, “”, “”, “”);
```

```
//连接MyTestDSN数据源（Access数据库）;
```

```
Ctrl.Table = “KingTable”;
```

```
//选择KingTable数据表;
```

```
Ctrl.AddFieldNo(1, “温度”, 75, 1);
```

```
//增加第1列字段，显示字段名称为“温度”，列宽
```

```
75，只读;
```

```
Ctrl.FetchData (); //刷新数据集;
```

11. DeleteAllFields () 方法:

功能:

删除当前控件中显示的所有字段。

说明:

此方法只删除当前表格中的显示字段，实际数据库中该字段并没有真正删除。

使用方法:

```
Ctrl.DeleteAllFields ();
```

删除当前表格中显示的所有字段。

12. DeleteField (STRING bstrName) 方法:

功能:

删除当前控件中显示的指定名称的字段。

参数:

STRING bstrName: 指定字段名称。

说明:

当所选字段中没有指定的bstrName字段，则调用失败。此方法只删除当前表格中的显示字段，实际数据库中该字段并没有真正删除。

使用方法:

```
Ctrl.DeleteField (strSelColName);
```

删除当前表格中显示的strSelColName 指定字段，StrSelColName为组态王7.0字符串变量。

13. DeleteFieldNo (LONG lCol) 方法:

功能:

删除当前表格中显示的指定列号的字段。

参数:

LONG lCol: 指定字段所在列(以0开始)。

说明:

当lCol为无效列时, 则调用失败。此方法只删除当前表格中的显示字段, 实际数据库中该字段并没有真正删除。

使用方法:

```
Ctrl.DeleteFieldNo (nSelCol );
```

删除当前表格中显示的nSelCol列指定字段, nSelCol为组态王7.0整型变量。

14. MaxValueInField (LONG lCol) 方法:

功能:

求指定字段所在列的最大值。

参数:

LONG lCol: 指定字段所在列(以0开始)。

使用方法:

```
fMaxValue = Ctrl.MaxValueInField(nSelCol);
```

得到当前表格 nSelCol 指定列数据的最大值。fMaxValue 为组态王 7.0 实

型变量，nSelCol 为组态王 7.0 整型变量。

15. MaxValueInFieldName (STRING bstrFieldName) 方法:

功能:

求指定字段所在列的最大值。

参数:

STRING bstrFieldName: 指定字段名称。

使用方法:

```
fMaxValue = Ctrl.MaxValueInFieldName(strSelColName);
```

得到当前表格 strSelColName 指定字段数据的最大值。fMaxValue 为组态王 7.0 实型变量，strSelColName 为组态王 7.0 字符串变量。

16. MinValueInField (LONG lCol) 方法:

功能:

求指定字段所在列的最小值。

参数:

LONG lCol: 指定字段所在列(以0开始)。

使用方法:

```
fMinValue = Ctrl.MinValueInField(nSelCol);
```

得到当前表格 nSelCol 指定列数据的最小值。fMinValue 为组态王 7.0 实型变量，nSelCol 为组态王 7.0 整型变量。

17. MinValueInFieldName (STRING bstrFieldName) 方法:

功能:

求指定字段所在列的最小值。

参数:

STRING bstrFieldName: 指定字段名称。

使用方法:

```
fMinValue = Ctrl.MinValueInFieldName(strSelColName);
```

得到当前表格 strSelColName 指定字段数据的最小值。fMinValue 为组态王 7.0 实型变量, strSelColName 为组态王 7.0 字符串变量。

18. SumInField (LONG lCol) 方法:

功能:

求指定字段所在列的累加值。

参数:

LONG lCol: 指定字段所在列(以0开始)。

使用方法:

```
fSumValue = Ctrl.SumInField(nSelCol);
```

得到当前表格 nSelCol 指定列数据的累加值。fSumValue 为组态王 7.0 实型变量, nSelCol 为组态王 7.0 整型变量。

19. SumInFieldName (STRING bstrFieldName) 方法:

功能:

求指定字段所在列的累加值。

参数:

STRING bstrFieldName: 指定字段名称。

使用方法:

```
fSumValue = Ctrl.SumInFieldName(strSelColName);
```

得到当前表格 strSelColName 指定字段数据的累加值。fSumValue 为组态王 7.0 实型变量, strSelColName 为组态王 7.0 字符串变量。

20. AverageValueInField (LONG lCol) 方法:

功能:

求参数LONG lCol以0为基数的字段所在列的平均值。

参数:

LONG lCol: 指定字段所在列(以0开始)。

使用方法:

```
fAveValue = Ctrl.AverageValueInField(nSelCol);
```

得到当前表格 nSelCol 指定列数据的平均值。fAveValue 为组态王 7.0 实型变量, nSelCol 为组态王 7.0 整型变量。

21. AverageValueInFieldName (STRING bstrFieldName) 方法:

功能:

求STRING bstrFieldName字段所在列的平均值。

参数:

STRING bstrFieldName: 为指定的字段名。

使用方法:

```
fAveValue = Ctrl.AverageValueInFieldName (strSelColName);
```

得到当前表格 strSelColName 指定字段数据的平均值。fAveValue 为组态王 7.0 实型变量, strSelColName 为组态王 7.0 字符串变量。

22. DifferLastFirstInField (LONG lCol) 方法:

功能:

求参数LONG lCol以0为基数的字段所在列的差值(最后一条记录与第一条记录的差值)。

参数:

LONG lCol: 指定字段所在列(以0开始)。

使用方法:

参数为组态王7.0变量:

```
fDifValue = Ctrl. DifferLastFirstInField(nSelCol);
```

得到当前表格 nSelCol 指定列数据的差值。fDifValue 为组态王 7.0 实型变量, nSelCol 为组态王 7.0 整型变量。

23. DifferLastFirstInFieldName (STRING bstrFieldName) 方法:

功能:

求STRING bstrFieldName字段所在列的差值(最后一条记录与第一条记录的差值)。

参数:

STRING bstrFieldName: 指定字段名称。

使用方法:

参数为组态王7.0变量:

```
fDifValue = Ctrl. DifferLastFirstInFieldName (strSelColName);
```

得到当前表格 strSelColName 指定字段数据的差值。 fDifValue 为组态王 7.0 实型变量, strSelColName 为组态王 7.0 字符串变量。

24. Config () 方法:

功能:

在运行状态弹出“数据查询选择”对话框供用户配置。如下图 14.87 所示。

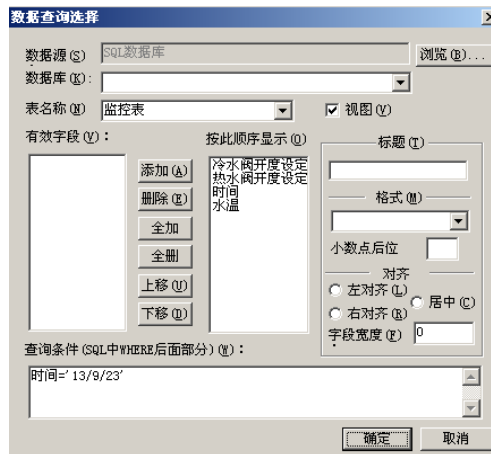


图14.87 config () 方法调用的数据查询选择对话框

25. SetConnectParam (STRING bstrDNS, STRING bstrCatalog, STRING bstrUserName, STRING bstrPWD) 方法:

功能:

设置ODBC连接参数。

参数:

STRING bstrDNS: ODBC连接的数据源名称;

STRING bstrCatalog: 指定的数据库名称, 在SQL Server、Oracle中指定数据库名称, 在Access数据库中为空, 不需指定, 用空格表示, 不能为空字符串;

STRING bstrUserName: 用于登陆连接的用户名, 如为空, 用空格表示, 不能为空字符串;

STRING bstrPWD: 登陆用户名的密码, 如为空, 用空格表示, 不能为空字符串。



例如:

Access数据库: Ctrl.SetConnectParam("MyTestDSN", " ", " ", " ");

SQLServer数据库: Ctrl.SetConnectParam("MySQLODBCName", "Northwind", "sa", " ");

一般, 此方法与设置控件的“Table”属性相配合来更改选择连接到的数据库中的某一具体表。

如以下程序片断将数据源名为“MyTestDSN”中表“TableOne”中的字段加入到控件中显示。

```
Ctrl.SetConnectParam ("MyTestDSN", " ", " ", " ");  
Ctrl.Table = "TableOne";  
Ctrl.AddField("Name", "姓名", 50, 1);
```

26. **SetSqlStringWhere**(string bstrWhere1, string bstrWhere2, string bstrWhere3, string bstrWhere4, string bstrWhere5, string bstrWhere6) 方法:
功能:

功能:

将六个string型变量:bstrWhere1-bstrWhere6合并成一个where语句。

参数:

bstrWhere1-bstrWhere6: 组合where语句的字符串。如为空, 用空格表示, 不能为空字符串;



例如:

Access数据库:

```
Ctrl.SetSqlStringWhere("冷水阀开度设定>20 ", "and ", "水温>10", "  
", " ", " ");
```

27. **SetRowHeightMothed**(LONG lRow, LONG lNewValue) 方法

功能:

设定指定行的行高。

参数:

LONG lRow: 指定行的行数(从0开始)。

LONG lNewValue: 设置值。



例如:

设置当前数据表第5行的行高为30:

```
Ctrl.SetRowHeightMothed(5, 30);
```

28. SetAllRowHeightMothed(LONG lNewValue)方法

功能:

设定所有行的行高。

参数:

LONG lNewValue: 设置值。



例如:

设置当前数据表所有行的行高为30:

```
Ctrl.SetAllRowHeightMothed(30);
```

29. SetAllRowHeightMothed(LONG lNewValue)方法

功能:

设定所有行的行高。

参数:

LONG lNewValue: 设置值。



例如:

设置当前数据表所有行的行高为30:

```
Ctrl. SetAllRowHeightMothed(30);
```

30. SetColLength(LONG lCol, LONG lLen)方法

功能:

设定指定列的列宽度。

参数:

LONG lCol: 指定列的列数(从 0 开始);

LONG lNewValue: 设置的列宽度(像素大小)。



例如:

设置当前数据表的第3列的列宽为30:

```
Ctrl. SetColLength (2, 30);
```

31. PrintPreview()方法

功能:

打印预览。



例如:

```
Ctrl. PrinterPreview();
```

32. SetStartEndRow(LONG Start, LONG End)方法

功能:

设定打印的起始行和结束行。

参数：

LONG Start：指定打印的起始行数(从 0 开始)；

LONG End：指定打印的起始行数(从0开始)。



例如：

打印当前数据表的第3到第6行：

Ctrl. SetStartEndRow (2, 5)。

14.3.4.4.3 KVADODBGrid 控件事件

以下控件事件使用请在控件的“动画连接属性—事件属性页”中关联组态王7.0控件事件函数，在函数输入命令语言。

1. MouseMove 事件：

说明：在控件内移动鼠标产生的事件；

2.MouseDown 事件：

说明：鼠标按下事件；

3. MouseUp 事件：

说明：鼠标弹起事件；

4. Click 事件：

说明：鼠标单击事件；

5. KeyDown 事件：

说明：键按下事件；

6. KeyUp 事件：

说明：键弹起事件；

7. KeyPress 事件：

说明：键压下时发生的事件；

8. StartEditing 事件：

说明：正在编辑单元格事件；

9. EndEditing 事件：

说明：完成编辑单元格事件；

14.3.4.5 KVADODBGrid 控件的查询条件向导

KVADODBGrid控件为用户在运行状态中提供了一个查询条件设置向导。如图14.77所示。用户可通过调用QueryDialog（）方法实现查询条件的自动配置。

函数调用方法：`nReturnValue = Ctrl.QueryDialog（）`；

参数：无

返回值：0 --查询窗口按“取消”返回，1--查询窗口按“确定”返回。

执行此函数后将弹出“查询条件对话框”，如图14.88所示。

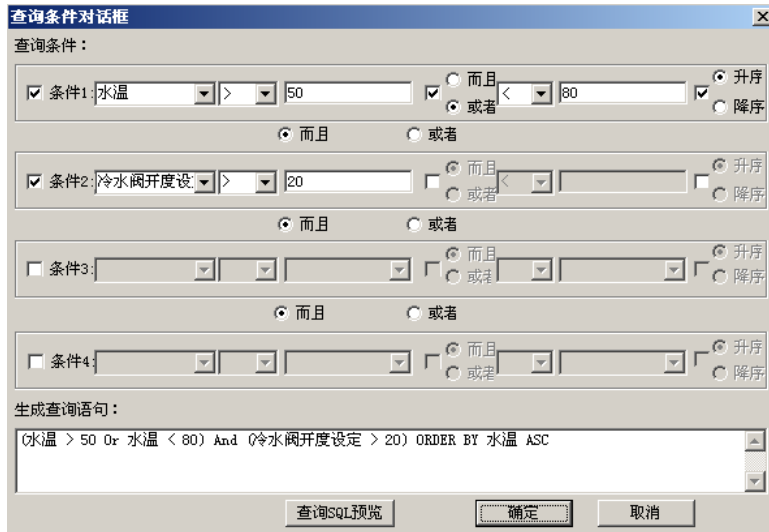


图14.88 控件查询条件对话框

此对话框可以设置四组查询条件的组合，并可以对查询数据进行排序。在对话框的左侧有四个复选框可以选择，可用于确定共有几组查询条件。用户可从下拉框中选择查询条件所需要的字段，之后选择比较操作符，构成条件所需要的数值。如果选择的字段为日期或时间型的类型，可以方便地从日期时间控件中选取日期时间值。在对话框的右侧可以选择是否按照相应的字段进行排序，可以指定按升序（ASC）或降序（DESC）排序。

配置好查询条件后点击“查询SQL预览”按钮，可自动生成查询语句中的查询条件，显示在“生成查询语句”编辑框中。点击“确定”后可将该查询条件保存下来。之后可使用FetchData（）方法按照此查询条件显示数据。当用户修改了控件的where属性后，该查询条件变为新的查询条件。

14.3.4.6 KVADODBGrid 控件连接数据库举例

1. 连接到SQL Server数据库：（通过ADOConnStr属性进行ADO连接）

其中数据库服务器名为 MySQLServer，连接的数据库为“Northwind”，表名为“Order Details”，登陆用户名为“sa”，密码为空，不需要建立 ODBC 数据源。以下程序片断将连接到 SQL Server 中，并 Order Details 表的所有字段添加到控件中显示。

```
Ctrl.DeleteAllFields ();
```

```
Ctrl.ADOConnStr = "Provider = SQLOLEDB.1; Persist Security Info = FALSE;
```

```
User ID = sa; Initial Catalog = Northwind; Data Source=MySQLServer";
```

```
Ctrl.Table = "Order Detail";
```

```
Ctrl.AddAllFields (45, 0);
```

2. 连接到 Access 数据库：（通过 ADOConnStr 属性进行 ADO 连接）

打开的数据库为“D:\KVADODBGrid 例程\kvdb.mdb”，此时不用建立 ODBC 数据源，登陆用户名和密码均为空。以下程序片断将连接到 Access 数据库“D:\KVADODBGrid 例程\kvdb.mdb”中，并将表“Tableone”的所有字段添加到控件中显示。


```
Ctrl.DeleteAllFields ();
```

```
Ctrl.ADOConnStr = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source =
```

```
D:\KVADODBGrid 例程\kvdb.mdb; Persist Security Info = False";
```

```
Ctrl.Table = "Tableone";
```

```
Ctrl.AddAllFields (45, 0);
```

 注意:

如手工填写 ADOConnStr 字符串需具备较深的数据库知识且很容易出错。可查看通过界面交互方式建立连接后的 ADOConnStr 属性值，并将此值应用到实现动态建立数据库连接功能的脚本编写过程中。

具体实现方式:

首先在组态王 7.0 中建立字符串内存变量如 strADO。在开发系统用户界面中交互设置好到某数据库的 ADO 连接后，将此连接的 ADOConnStr 赋值给 strADO，即 strADO=Ctrl0.ADOConnStr，此时就可以在运行状态下通过 strADO 查看此连接的 ADO 字符串值了，可将此字符串值记下来供脚本编写中使用。

3. 连接到 SQL Server 数据库: (用 SetConnectParam 方法调用通过 ODBC 数据源进行连接) 连接的数据库为 “Northwind”，表名为 “Order Details”，登陆用户名为 “sa”，密码为空，需要建立 ODBC 数据源，名称为 “MyTestDSN”，以下程序片断将连接到 SQL Server 中，并将表 “Order Details” 中的所有字段添加到控件中显示。

```
Ctrl.DeleteAllFields ();  
  
Ctrl.SetConnectParam (“MySQLODBCName”, “Northwind”, “sa”, “ ”);  
  
Ctrl.Table = “Order Details”;  
  
Ctrl.AddAllFields (45, 0);
```

4. 连接到 Access 数据库: (用 SetConnectParam 方法调用通过 ODBC 数据源进行连接) 建立连接到 Access 数据库的 ODBC 数据源，将此数据源中表 “TableOne” 中的所

有字段添加到控件中显示。

```
Ctrl.DeleteAllFields ();
```

```
Ctrl.SetConnectParam ("MyTestDSN", " ", " ", " ");
```

```
Ctrl.Table = "Tableone";
```

```
Ctrl.AddAllFields (45, 0);
```

注：1、2 事例采用 ADODConnStr 属性通过 ADO 进行连接的运行性能要好于 3、4 事例采用 SetConnectParam 方法通过 ODBC 进行数据库的连接。

KVADODBGGrid 控件可以用做大批量数据的查询工具使用。目前该控件有下列功能上的限制：


- ✧ 一个控件只能从一个表中提取数据，不能从多个表中联合查询数据。不过用户可以自己建立视图，此视图可以从多个表中联合查询，通过选择视图可间接的达到从多个表中联合查询数据的目的。
- ✧ 不能整体更新数据库中内容。另外，组态王 7.0 还提供了超级 X-Y 曲线控件，是 Active X 控件，具体使用方法请参见第八章 趋势曲线和其它曲线。组态王 7.0 产品中根据用户需求不断增加和更新控件和应用程序解决方案，其它实用解决方案产品及使用说明请参见组态王 7.06.55 产品 CD 中的“解决方案产品”目录，或访问亚控网站。

14.3.5 树形控件

树形控件是一种人们熟悉的用户界面控件，广泛地用来分级显示项目列表的窗口。树

形控件所包含的项目以相互关联的方式显示在控件中,通过点击位于某个层次的项目节点可以展开下一层次中从属于该节点的所有项目。树形控件适合于管理层次较多且相互间隶属关系较为清晰的项目元素。

14.3.5.1 创建 KingTree 控件

在组态王 7.0 图形开发系统中打开任一画面,单击工具箱中的  图标,选择“KingTreeCtrl”,如下图所示。此时把鼠标放在画面编辑区,鼠标将变成小“十”字形,在画面上需要插入控件的地方按下鼠标左键,拖动鼠标,画面上出现一个矩形框,表示创建后控件界面的大小。松开鼠标左键,控件在画面上显示出来。

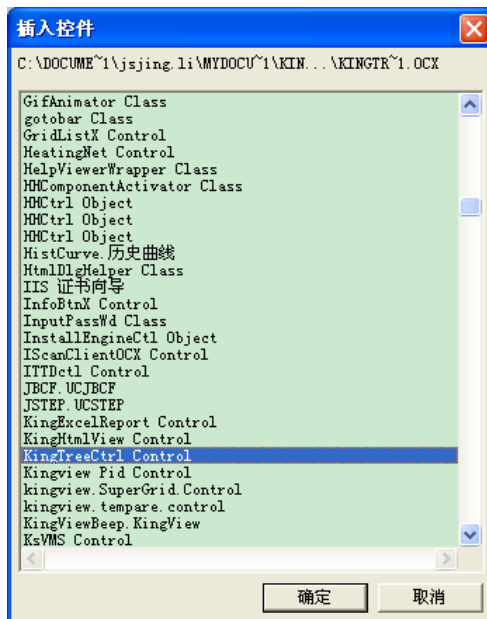


图 14.89 插入控件列表

14.3.5.2 设置 KingTree 控件的固有属性

选择控件，单击右键，在弹出的菜单中选择“控件属性”。弹出控件固有属性对话框，如下图 14.90 所示。

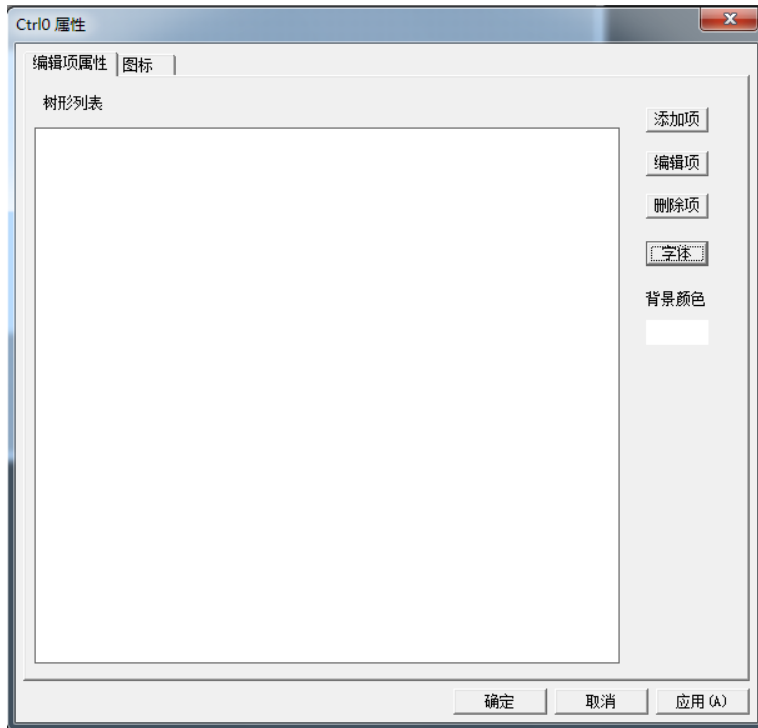


图 14.90 KingTree 控件属性

14.3.5.2.1 编辑项属性页

该属性页主要实现定义树视图控件中各项的功能。

- ✧ **添加项：**单击“编辑项”按钮，弹出如图“编辑项”对话框。

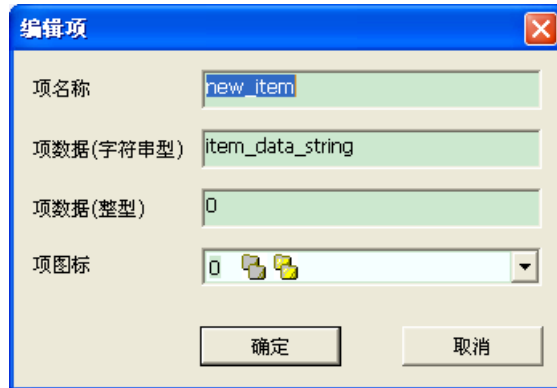


图 14.91 添加项

- 项名称: 设置树形控件中显示项目的名称。对应事件函数参数 `item_name`。
- 项数据 (字符串型): 树形控件中标识该项目的字符串, 供事件函数参数 `item_data_string` 调用。
- 项数据 (整数): 树形控件中标识该项目的整型数据, 供事件函数参数 `item_data_long` 调用。
- 项图标: 设置树形控件中标识项目的图标显示。对应事件函数参数 `item_item_type`。

树形控件支持建立多层次项目。选中需要建立子项目的项目名称, 单击“添加项”或在右键快捷菜单中选择“新建”, 即可建立该项目的子项目。


✧ **编辑项:** 选中需要编辑的项, 单击“编辑项”, 弹出如图4所示对话框。



图 14.92 编辑项

在编辑项对话框中重置相应项，具体请参考“添加项”设置。

- ◇ **删除项：**删除选中的项。此项甚用。
- ◇ **字体：**为树形控件中的项文本设置字体。
- ◇ **颜色：**为树形控件设置背景色。

 注意：

1. 树形控件支持建立多级子项目。
2. 删除项时，同时删除其所属的子项目。

快捷菜单：

右击编辑项属性页空白处，弹出下图 a 所示快捷菜单，右击项目名称弹出快捷菜单下图 b 所示



a



b

- ◇ **清空:** 删除选中项目下所属的全部子项目。
- ◇ **导出:** 图 a 是将控件的所有项目信息导出为文件格式，图 b 是将选中项目及其所属子项目的全部信息导出，导出文件为 .dr1 格式。
- ◇ **导入:** 将项目信息添加到树形控件中，导入文件为 .dr1 格式。

 **注意:**

利用导出功能导出的 .dr1 文件包括项目编辑项信息，而不包含事件函数脚本信息。

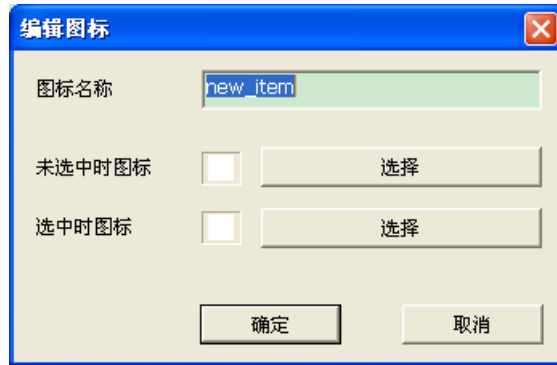


图 14.94 编辑图标对话框

- 图标名称：新建图标的名称。
 - 未选中时图标：未选中项时，显示的图标。
 - 选中时图标：选中项时，显示的图标。
- ◇ **编辑**：修改已建立的图标。
- ◇ **删除**：删除选中的图标。

14.3.5.3 设置 KingTree 控件的动画连接属性

用左键双击 KingTree 控件，弹出“动画连接属性”对话框，或者用右键单击 KingTree 控件选择菜单命令“动画连接”，则也弹出“动画连接属性”对话框如下图 14.90 所示：

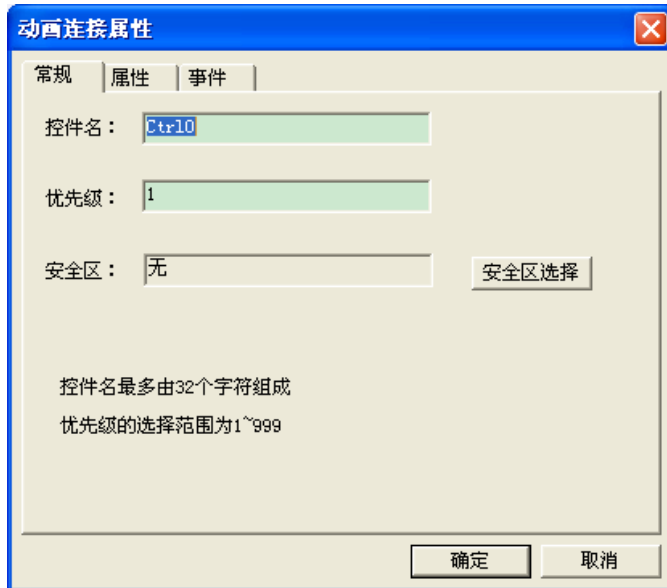


图 14.95 设置控件属性

14.3.5.3.1 常规属性页

动画连接属性页——常规如下图所示：

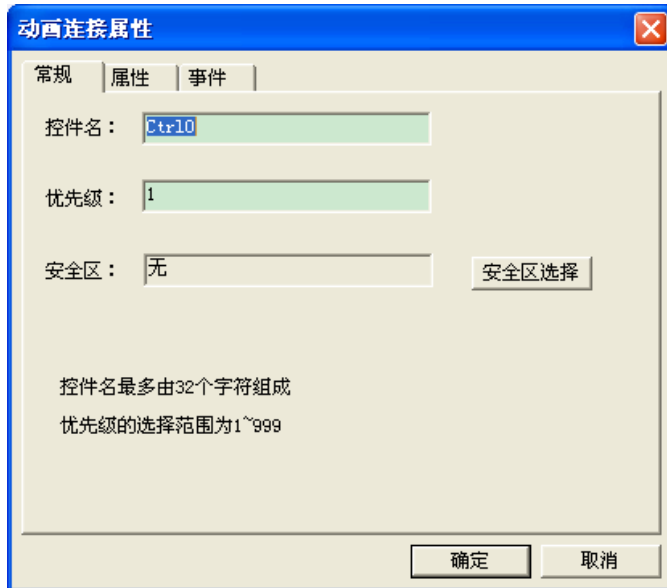


图 14.96 常规属性设置

- ◇ **控件名：**一个 KingTree 控件对应一个控件名称，而且是唯一的，不能重复命名，控件名由工程人员设定，例如设为“Ctrl10”。
- ◇ **优先级：**用于设置访问该控件的操作优先级级别，优先级级别从 0-999。
- ◇ **安全区：**单击右侧的“安全区选择”按钮，选择该控件的操作安全区。

14.3.5.3.2 事件属性页

动画连接属性页——事件如下图所示：



图 14.97 事件设置页

- ◇ **控件事件函数:** SelChanged()
- ◇ **函数原形:** SelChanged(String item_name, String item_data_string, LONG item_data_long, LONG item_item_type)
- ◇ **函数功能:** 应用树形控件中各个项目的设置信息，实现项目事件的响应。
此项功能通过输入脚本函数实现。

14.3.5.4 KingTree 控件的使用

第一步：新建树形控件

第二步：设置树形控件

1. 在图标项下添加树形项图标：



图 14. 98 添加树形图标

2. 添加和设置编辑项属性，如下图所示

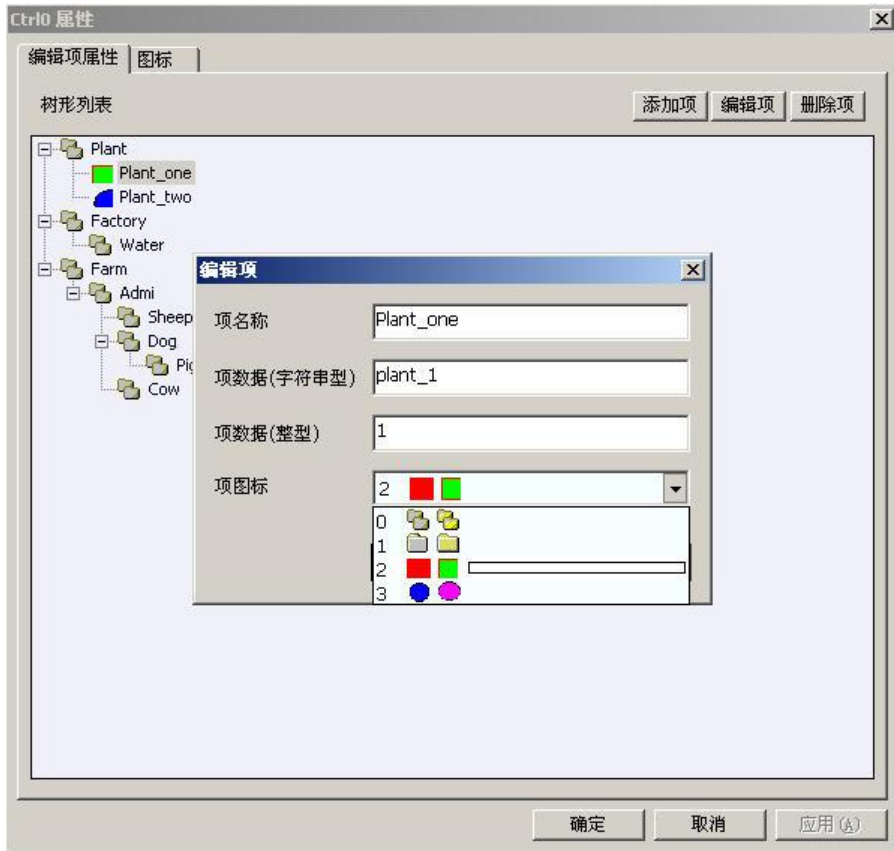


图 14.99 设置编辑项

第三步：编辑脚本。

例如，在控件事件函数窗口中输入如下脚本：

```
if(item_name == "Plant_one")  
ShowPicture("Hist");
```

第四步：保存设置，运行组态王 7.0。

切换到运行系统，单击“Plant_one”项，则弹出“Hist”窗口。

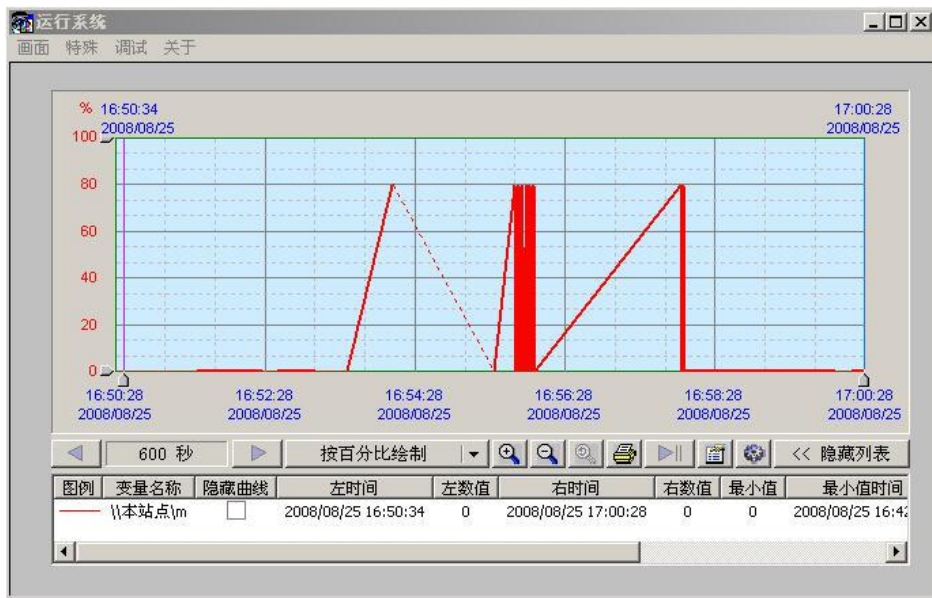


图 14.100 运行效果图

14.3.6 温控曲线控件

在某些制造行业生产产品的过程中，往往需要控制某些生产参数随着时间的推移不断进行调整、变化。如面包生产过程中，在不同的时间段，需要对加温温度进行控制，每个阶段要求的时间长度和温度值不同。这样，整个生产过程的参数按照时间关系就形成了

一条前后连接的曲线。该曲线被称为温控曲线或计划曲线。

为了方便用户制作温控工程，组态王 7.0 提供了温控表格和温控曲线控件。温控曲线控件的功能主要是显示设定曲线和实时曲线，设定曲线的数据来自于数据库文件；而表格控件则负责设置设定曲线。用户通过表格控件操作数据库文件中的数据，添加、删除记录，数据库文件中的每一条记录对应显示曲线上的每一个控制段，如下图所示：

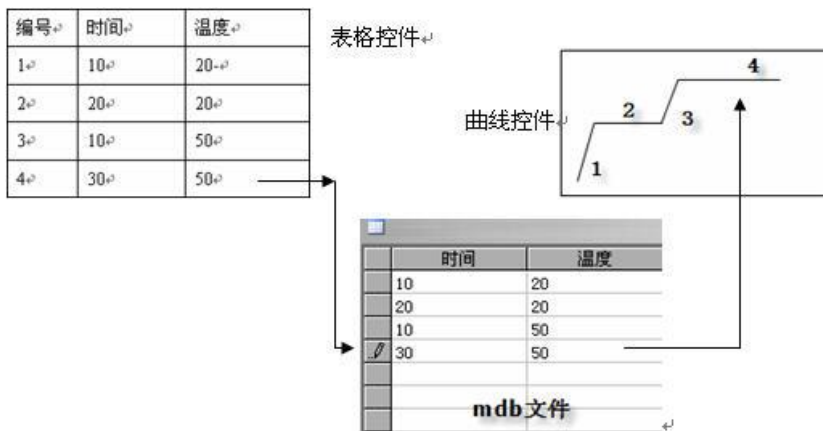


图 14.101 数据库文件到曲线的转换

以表格控件的第 4 条记录为例，参照上一条记录，第 4 条记录表示在 30 分钟内，温度保持为 50 摄氏度不变；第 3 条记录表示在 10 分钟内，温度从 20 摄氏度上升到 50 摄氏度。表格控件把这些信息存储到数据库文件中，曲线控件从数据库文件中再读出这些数据，显示到画面上，第 4 条记录就对应到曲线上的第 4 段水平线了。

控制参数编辑表主要有以下功能：

◇ 用户可以定义要使用的参数编辑表的表头，表格格式设置类似，可灵活设置；

- ◇ 用户定义各参数与控制设备（如 PLC）的信号通道（寄存器）的对应关系；
- ◇ 用户在组态时即可输入、编辑、保存控制参数数据；
- ◇ 控制参数数据的保存采用通用数据库，具有通用性；
- ◇ 在系统运行时，自动调入用户事先设置的参数。如果用户需要加入和删除控制段，在表格上直接选中要插入控制段，或要删除的控制段，通过简单的一个菜单项的选择，可轻松的插入或删除控制段。在用户没有确认这些操作时，系统会对这些项用不同的颜色标识，方便用户使用；
- ◇ 该表格编辑的控制参数数据具有通用性，在一个工程中使用的控制参数，可以使用到另外的工程中；
- ◇ 可以对表格进行打印预览、打印等操作。

温控曲线具有以下特点：

- ◇ 一个温控曲线图表上可以同时绘制多达 20 条的实时曲线；
- ◇ 曲线时间坐标轴的长度可以按照用户定义的时间参数来自动调整；
- ◇ 曲线图表的格式设置内容广泛，形式多样，适用各种场合的要求；
- ◇ 支持游标的使用，游标移动时可自动显示掠过的曲线点的数据；
- ◇ 支持图例的名称定义，图例的显示和打印；
- ◇ 曲线支持导航功能：放大、缩小、移动；曲线自动将绘制的实时曲线和参数曲线保存为历史数据，用户可在控制过程结束后查阅；
- ◇ 用户可自由控制实时曲线的绘制启动和停止；
- ◇ 最重要的功能：如果系统突然出现当机，当系统再启动后，曲线会调出当机前的

实时曲线，然后按照实际时间继续绘制曲线，有效的保护了数据的完整性。

在组态王 7.0 系统中，提供了参数数据批量下载的功能，用户无需再定义大量的变量，只需在脚本中简单调用该功能，系统将自动将参数数据轻松下载到控制设备中，该功能的实现需要使用 512 点以上的加密锁。另外，温控曲线的参数曲线与参数设置表间的数据结合也只需简单的一个配置即可完成，具有很强的易用性。

14.3.6.1 温控表格控件概述

表格控件以 OCX 控件形式存在，可被组态王 7.0 开发系统与运行系统调用。表格控件主要实现 Excel 方式编辑控制参数，并能将编辑结果传递到温控曲线实时监控。在开发时能对表头进行设计，具有合并、拆分单元格功能，并可进行字段设计，对应实际控制参数（在表格中以列形式存在）。运行时，能以行删除、插入方式实现未执行控制段（在表格中以行形式存在）的删除与添加操作，能通过单元格双击弹出对话框方式修改控制段参数内容，并自动保存到控制参数设置表。其显示内容可以表格形式打印，在运行时以脚本方式或属性关联方式调用。

14.3.6.2 如何在组态王 7.0 中使用温控表格控件

14.3.6.2.1 创建表格控件

在组态王 7.0 工具箱上单击“插入通用控件”或选择菜单“编辑\插入通用控件”命令。弹出“插入控件”对话框，如图 14.102 所示。

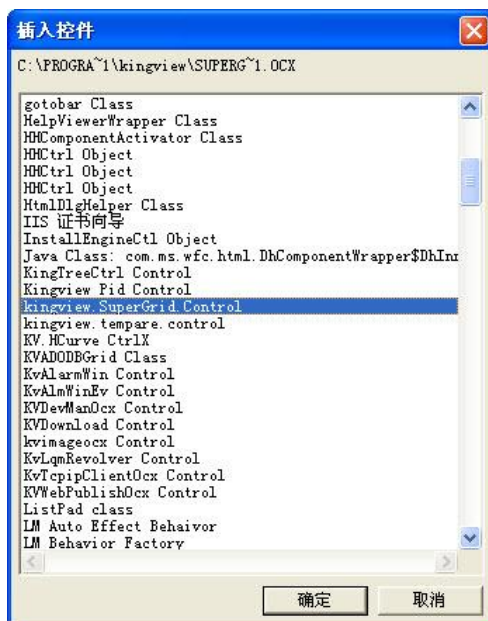


图 14.102 插入通用控件对话框

在对话框的列表中找到“kingview.SuperGrid.Control”项，选中它，然后单击“确定”按钮，或直接用鼠标双击该项。“插入控件”对话框自动关闭，按照本节开始所描述的方法在画面上创建控件。创建后的控件如图 14.103 所示。

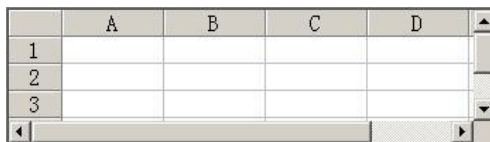


图 14.103 创建后的表格控件

14.3.6.2.2 设置表格控件的动画连接属性

用鼠标双击控件，系统弹出控件“动画连接属性”对话框。如图 14.104 所示。动画

连接属性页由三页组成：常规、属性和事件。

1、常规属性页

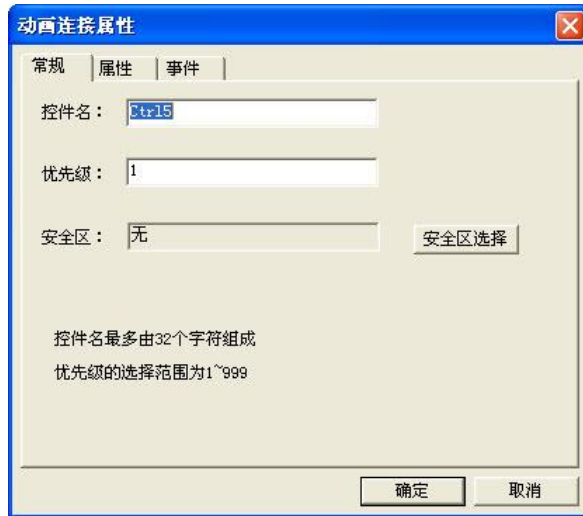


图 14.104 控件动画连接属性对话框——常规页

在这个对话框的常规页面中可以分别设置控件的名称、优先级和安全区。

2、属性页

单击动画连接属性的“属性”标签，显示“属性”页，如图 14.105 所示。




图 14.105 控件动画连接属性对话框——属性页

在“属性”页的列表中，列出了控件的所有属性和类型。例如，nControlID 属性，它的重要作用是通知表格控件执行到哪一个控制段，而通过调用温控曲线控件的方法 GetCurrentControlField() 可以得到当前执行段。而后和 nControlID 关联起来；RunTimeCanEdit 属性，使得用户在运行状态下可以通过脚本对该属性赋值，来控制是否可以编辑表格。

有关控件动画连接属性的具体含义、设置、使用方法请参见《组态王 7.06.55 使用手册》“控件”一章。

14.3.6.2.3 表格控件的初始化

控制参数的定义和初始输入，参数与设备寄存器的对应配置，表格的格式设置等都需要在工程组态时完成。

 注意：

因为表格控件的特殊性，在定义完成控件的动画连接属性后，在选中控件的情况下，同时按组合键“<Ctrl>+<Alt>+0”进入表格控件的编辑状态；再次按组合键“<Ctrl>+<Alt>+0”，退出表格控件的编辑状态。

表格处于编辑状态后，双击表格窗口的灰色部分（表格单元格区域外没有单元格的的部分），弹出“超级表格设计”对话框，如图 14.106 所示。该对话框主要设置表格控件表头部分的行列数目。默认的初始行列为 5 行 5 列。



图14.106 超级表格设计对话框

对话框中的行数和列数设置是针对表格的表头部分，标题部分在系统运行时系统将自动添加。

行数：代表两个部分。假如设置行数为 5，则前 4 行设置表头部分，最后一行设置表体部分格式（单元格边框、字体、颜色等）。系统进入运行状态后，表头部分将处于锁定保护状态，不能对其进行任何操作。

 注意：

所以在设计表头时，最后一行不能进行单元格的合并，只能在单元格中输入字符，和设置其单元格格式。这里提供一个简单的计算表头行列数的公式：行数=设计表头区需要行数+1；列数=所有参数个数

表格控件提供了“表格工具栏”，如图 14.107 所示。有关“表格工具栏”中各按钮的具体含义、设置和使用方法，请参见《组态王 7.06.55 使用手册》中的“报表系统”一章。



图14.107 表格工具栏

14.3.6.2.4 报表表头的设计（报表表头的保存和加载）

设计好一个表格表头后，单击“表格工具栏”的“保存模版表头”按钮，系统弹出“另存为”对话框，如图 14.108 所示。输入文件名，单击“保存”，将当前文件保存为模版文件，扩展名为.ght。



图14.108 保存模版表头


在画面中选中要加载表头模版的表格控件，单击“表格工具栏”中的“打开表头模版”按钮“A screenshot of a Windows-style '打开' (Open) dialog box. The title bar reads '打开'. The '查找范围 (I):' field shows a folder icon and the text '新'. The main area contains a list of files and folders: 'Webs' (folder), '本站点' (folder), '温控曲线' (folder), 'biaotou. ght' (file), and 'rde. ght' (file). At the bottom, the '文件名 (N):' field contains '*. ght'. The '文件类型 (T):' dropdown menu is set to '组态王表头模板文件 (*. ght)'. To the right of the filename field is a '打开 (O)' button, and to the right of the file type dropdown is a '取消' button.

图14.109 加载表头模版

加载表头模版时，除加载表头格式外，其中保存的参数数据库关联信息、字段名、时间字段等信息同时被加载进去，即表格的数据库关联被加载的表头信息更新。

14.3.6.2.5 与数据库的关联

设计好表头后，就可以进行组态阶段的控制参数的配置和录入。

编辑状态下，在表格单元格上单击鼠标右键，选择“字段名设置”项，弹出“字段名设置”对话框，如图 14.110 所示。



图14.110 字段名设置

“编号”表示表格中的列标编号，其个数与“超级表格设计”中的列数相同；双击“字段名”中的名称，设置控制参数名称。设置完成后，单击“确定”按钮，关闭对话框。



图14.111 编辑后的字段名设置对话框

组态王 7.0 中对控制参数的保存采用数据库的形式，所以这里设置的字段名为存储参数数据表中的字段名称。在这里设置的数据在系统运行时加载数据后显示在表格控件中，用户可以对表格中的数据进行编辑。

- ◇ 编辑状态下，在表格单元格上单击鼠标右键，选择“虚参数设置”项，弹出“虚参数表设置”对话框，如图 14.112 所示。




图14.112 虚参数表设置

单击对话框中的“添加”按钮，则自动增加一个虚参数字段，双击“虚参名”下的名称，修改虚参数名；如果不需要某个虚参数，单击“删除”按钮，对话框将自动排列当前的列表。定义完成后，单击“确定”按钮，关闭对话框。



图14.113 编辑后的虚参数表

- **控制参数数据库文件路径名设置：** 设定参数、驱动配置数据库文件的存储路径和文件名。单击“浏览”按钮选择存储路径和文件名或直接在显示框中输入，系统默认保存在当前工程目录下，名字为“control.mdb”。数据库中包含三张表，分别为参数配置表、虚参数表和驱动配置信息表。
- **ID：** 顺序号，每添加一个字段到表中，ID号都会自动增加。
- **参数名：** 需要下载的参数，双击对应ID的单元格，弹出可设置的参数下拉列表框，包含是参数和虚参数，在列表中选择一个参数名。
- **属性：** 根据所选择的参数，系统自动显示其属性（实参数或虚参数）。
- **设备名：** 参数所对应的设备名。该设备名必须是组态王7.0中已定义过的设备名称，名称要一致，否则系统无法正常下载参数。
- **寄存器名：** 该参数下载对应设备的寄存器名称。
- **起始地址、终止地址：** 该参数数据集下载到设备寄存器时对应的寄存器的起始地址和终止地址范围。
- **数据类型：** 参数（寄存器）的数据类型。
- **最小值、最大值：** 输入对应参数的最小值和最大值。这里定义的范围必须是所选数据类型的范围内，在输入初始控制参数和系统运行过程中修改参数必须在这个范围之内，否则会给出相应的提示对话框。如果调用脚本修改参数时，超出设定范围，则会按照设定范围的边界显示。

 **例：** 如参数温度，数据类型为SHORT，设置其最小最大值为-50、150，在编辑控制参数时输入300，则会给出相应的提示消息，如果用脚本在运行的过程中修改，如

赋值为300，则系统会将该参数置为其最大值。

关于设备定义和寄存器名称、范围、数据类型请参考组态王 7.0 关于设备支持的相关帮助内容。

- **添加：**新增加一个要下载的参数到配置表中。
- **删除：**从表中删除已建好的字段。
- **校验：**添加字段的过程中校验用户的输入，如果有误，系统会给出相应的提示信息。
- **确认：**配置完成后，单击“确认”按钮，保存数据信息，并关闭对话框。
- **取消：**取消当前的操作，关闭对话框。

☆ 每个控制过程都由不同时间长度的曲线段组合而成，而每个曲线段的时间加起来，就是整个控制周期。组态王 7.0 温控版中可以自动根据提供的时间字段的值之和作为温控曲线时间轴的长度。为实现该目的，在组态时需要指定参数中的时间字段。

在任意单元格上单击鼠标右键，选择“设置时间字段”项，弹出“时间字段设置”对话框，如图 14.115 所示。在列表中选择要定义为时间的自动名称，如“段时”，单击“确定”，系统给出选择提示对话框。时间字段支持的数据类型有 LONG，SHORT，USHORT。



图 14.115 时间字段设置

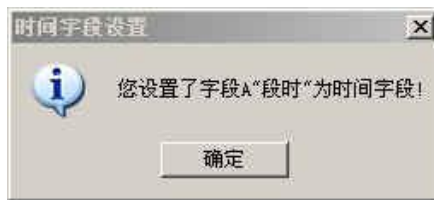


图 14.116 时间字段设置成功

- ◇ 控制参数设置表用于保存需下载到现场设备的控制参数的数据表，可在开发时由表格控件以配方方式生成、编辑，也可在运行时进行修改编辑。在表格的任意单元格上点击鼠标右键，在弹出快捷菜单中选择“编辑控制参数”，弹出设置时间字段对话框，如图 14.117 所示，控制参数设置表的第一列为系统自动编号列，由系统自动生成，也可看作是段号。其它列对应一个字段，字段名来源于表格控件的字段设计和虚参数设计；控制参数设置表的行对应控制段。



图14.117 编辑控制参数对话框

可以通过“添加”、“插入”和“删除”按钮对数据个数进行设置；双击相应单元格设置数据项。

✧ **变更关联数据库** 当表格控件不再需要与参数数据库关联时，可以取消数据库的关联。即，在弹出快捷菜单中选择“取消数据库关联”项。取消数据库关联后，表格控件不再拥有任何配置信息，此时可以新建配置信息。

如果拥有多个参数数据库，针对不同的控制对象需要加载不同的数据库时，可以选择“关联其它数据库”。在弹出数据库文件选择对话框中进行选择即可。

也可以将当前配置好的数据库另命名保存，重复使用。在弹出快捷菜单中选择“另存其它数据库”项，弹出文件保存对话框，选择保存路径和文件名即可。

选择“另存其它数据库”后，当前表格关联的数据库自动关联到新的数据库上。

✧ **表格安全管理** 选择弹出快捷菜单中的“属性设置”，弹出如图 14.118 所示对

话框。



图14.118 属性设置

- **“最大编辑行数”**：表示表格在组态或运行时添加的控制参数数据记录行数不能超过的最大数值，行数设置范围为1-195。最大行数的设置可以帮助用户进行驱动配置时正确计算参数的地址范围，防止用户在系统运行过程的操作出现错误，导致参数下载失败。
- **“锁定方式”**：分为“区域锁定”和“行锁定”。“区域锁定”是表格控件在运行过程中将所有已执行的控制段都锁定起来，不允许用户对这些锁定的数据进行任何编辑，包括插入或删除行，及对单个数据的修改。“行锁定”是对当前执行的控制段进行锁定，其前面的已执行过的控制段可以进行单个数据的修改，但不能执行其它操作，如插入行，删除行等操作。这些可以保护运行过程中控制段数据的紊乱。
- 选择密码保护功能，在运行中只有输入正确的密码，才能对表格控件的控制段数据进行控制段的插入、添加、删除、数据的修改等操作。在运行时，如果需要修改控制段参数，在表格上点击鼠标右键，在弹出的快捷菜单中

选择“锁定操作”，系统弹出“解除锁定”对话框，输入正确的密码后，就可以进行操作了。操作完成后，再次选择“锁定操作”菜单，系统重新对表格进行保护。

保护表格数据不被任意编辑，除使用“密码保护”功能外，也可以在定义控件动画连接属性时定义控件的优先级和安全区，这样利用组态王7.0中的“用户配置”来达到管理的目的，用户只有使用组态王7.0的用户权限登录成功后才可以操作控件。

14.3.6.2.6 系统运行中编辑表格

在组态王7.0运行系统启动后，表格控件处于初始状态。首先要激活表格控件，加载参数数据，才能进行编辑操作。在按钮或画面的脚本命令语言中调用控件的“Run”方法，在运行系统中打开画面后，执行该脚本程序，使表格进入运行状态，自动加载参数数据。

表格被激活运行后，调入参数数据库，就可以进行编辑了。对表格的编辑可以：添加新行、插入新行、删除行或取消删除、修改参数数据、锁定操作。这些操作，通过在表格控件上点击右键的快捷菜单完成。快捷菜单如下图所示。



14.3.6.2.7 如何使用表格控件

Run()

此函数用于在系统运行中激活表格控件,使其处于运行状态。可以在按钮命令语言中输入:控件名.Run()。运行状态下,单击按钮,则表格控件进入运行状态。

SaveDefault()

此函数用于在运行状态时将表格修改后的数据保存在默认的数据库中。(开发时设定的数据库)

Save(DSTRING strDNPath)

此函数用于在运行状态时,将表格修改后的数据保存在指定路径下的数据库。字符串型strDNPath,是保存路径的完整名称和文件名。



例 1:

```
String strDBPath="E:\数据库\新数据库.mdb"
```

```
Canshubiao.Save(strDNPath);其中strDNPath为表格控件名称。
```

long GetRowCount()

此函数用于获取控制参数表记录行数,返回值为整型,是参数数据的行数(控制段数)。

long GetColCount()

此函数用于获取控制参数表字段列数，返回值为整型，是参数数据的列数（参数个数）。

long GetVirRowCount()

此函数用于获取虚参数表记录行数，返回值为整型。

long GetVirColCount()

此函数用于获取虚参数表列数，返回值为整型。

float GetCellValue(LPCTSTR strFieldName, short nRow)

此函数用于获取控制参数表指定字段“strFieldName”和指定行（控制段）“nRow”相交处的单元格数值。

float GetRCCellValue(short nRow, short nCol)

此函数用于获取控制参数表指定“nRow”行“nCol”列单元格数值，返回值为实型。

float GetVirCellValue(LPCTSTR strVirtualName, short nRow)

此函数用于获取虚参数指定字段“strVirtualName”和指定行（控制段）“nRow”相交处的单元格数值。返回值为实型。

BOOL SetRCCellValue(short nRow, short nCol, float fValue)

此函数用于设置控制参数表指定的“nRow”行“nCol”列单元格的数据。fValue为实型设定值。函数的返回值为BOOL型，设置成功，函数返回1，失败则返回0。

BOOL SetVirCellValue(LPCTSTR strVirtualName, short nRow, float fValue)

此函数用于设置虚参数表指定字段“strVirtualName”和单元格“nRow”行单元格的数据。fValue为实型设定值。函数的返回值为BOOL型，设置成功，函数返回1，失败则返回0。

BOOL SetVirRCCellValue(LONG nRow, LONG nCol, float fValue)

此函数用于设置虚参数表指定的“nRow”行“nCol”列单元格的数据。fValue为实型设定值。函数的返回值为BOOL型，设置成功，函数返回1，失败则返回0。

SetupPage()

此函数用于运行时对要打印的控件表格进行页面设置。

PrintAsReport()

此函数用于运行时将控件表格内容按照表格形式直接输出到指定的打印机，而不再显示打印属性对话框。

PrintManual(BOOL bAuto)

此函数用于运行时按照不同的参数配置，将表格内容自动或手动输出到打印机。BOOL型参数“bAuto”，为1时，同PrintAsReport()功能一样，直接输出到打印机；为0时，首先弹出打印属性对话框，待确认后再输出到打印机。

PrintPreview()

此函数实现控件表格内容的打印预览。

RefreshGrid()

表格控件启动时读一次数据库，此后进行的操作都是写数据库。假如数据库的数据通过其它的方式进行了修改，并需要将数据刷新到表格控件中，就要调用这个函数。

LoadDriverConfig(sDriverFilePath, nRow)

根据“sDriverFilePath”中指定的参数数据库路径和文件名称及“nRow”指定的要下载的参数段号，此函数用于将指定控制段的一个或全部数据一次性写入控制设备中。当nRow=9999时，下载驱动配置表中指定的所有参数的所有数据。此函数为组态王7.0公用函数。

返回值为整型：

- 1 表示下载失败（此处的下载失败只指组态王7.0读取数据错误或驱动配置校验错误，而不是指数据写设备成功）

- 0 表示下载成功
- 1 打开数据库文件失败
- 2 驱动配置表中无记录
- 3 获得控制字段名失败
- 4 根据字段得到驱动信息失败
- 5 没有发现对应设备的采集线程,有可能表中的设备名称和工程不一致
- 6 没有发现对应设备的设备地址结构,有可能表中的设备名称和工程不一致
- 7 没有得到驱动接口指针
- 8 下载参数没有通过驱动程序的校验
- 9 驱动配置信息无参数
- 10 为未定义
- 11 下载点数超过限制
- 12 表没有被记录
- 13 非温控版不支持数据下载
- 14 演示模式不支持数据下载



例 2:

1、下载驱动配置中指定的段号为1的参数

```
long lRet;  
  
String strDBPath="E:\数据库\control.mdb";  
  
long nRow=1;  
  
lRet=LoadDriverConfig (strDBPath, nRow) ;
```

2、下载驱动配置中指定的全部参数

```
long lRet;  
  
String strDBPath="E:\数据库\control.mdb";  
  
lRet=LoadDriverConfig (strDBPath, 9999) ;
```

此函数的使用需要将表格控件和温控曲线控件结合起来，首先调用表格控件的脚本SaveDefault()将表格控件的数据刷新，因为可能修改了数据但没有保存，然后调用组态王7.0的系统脚本执行下载参数的操作，下载完成后，调用温控曲线的脚本SetControlField(String szFilename, String szFieldName)将修改后的设定曲线刷新显示。



注意：

第一次下载参数，此函数会根据参数nRow的值下载全部或者指定行的所有数据，此后，在此调用此函数执行下载功能时，只会对新修改的数据执行下载功能（新插入行及其后面的数据都属于新的数据），未修改数据不会下载。

14.3.6.3 温控曲线

组态王7.0温控版提供的温控曲线在实时曲线的绘制、设定曲线的关联、实时曲线的支持数量等与以前大不相同。用户可以控制实时曲线绘制的启动和停止，还可以自动存储趋势曲线。

温控版温控曲线在组态实时曲线时，直接通过实时采集变量与控件属性的关联，然后设置控件实时曲线的更新周期，使系统启动运行开始控制后，自动定时绘制实时曲线，避免了大量编写脚本程序。每个曲线图表设置设定曲线时，只需选择在组态表格控件时的数据库和参数字段。

该温控曲线每个曲线图表支持20条实时趋势曲线的绘制，可以在运行时隐藏其它实时曲线，而专门查看某条或某几条曲线的趋势。

曲线支持自动保存图表上的设定和实时趋势曲线，当系统关机重启后，系统可以调出关机前的曲线，按照时间进展情况自动开始绘制趋势曲线。当一个控制过程结束后，系统将实时和设定曲线自动保存，用户可以通过曲线工具条中的历史趋势查看按钮打开某个趋势数据库，查看过去某个控制过程的运行趋势。

14.3.6.3.1 创建温控曲线

在组态王7.0工具箱上单击“插入通用控件”或选择菜单“编辑\插入通用控件”命令。在弹出的“插入控件”对话框中，选择“kingview.tempare.control”，单击“确定”。在画面中创建温控曲线如图14.119所示。

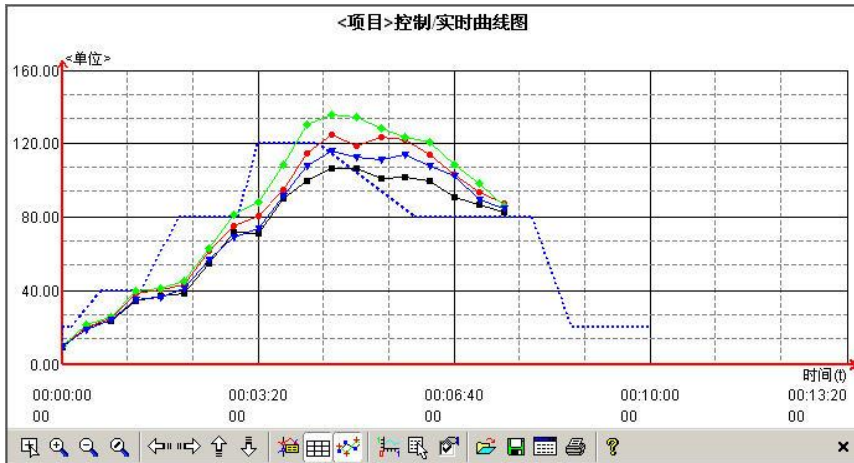


图14.119 创建后的温控曲线

14.3.6.3.2 温控曲线动画连接属性

14.3.6.3.2.1 动画连接属性——常规页

选中画面上创建的曲线，双击左键，弹出“动画连接属性”对话框，如图14.120所示。

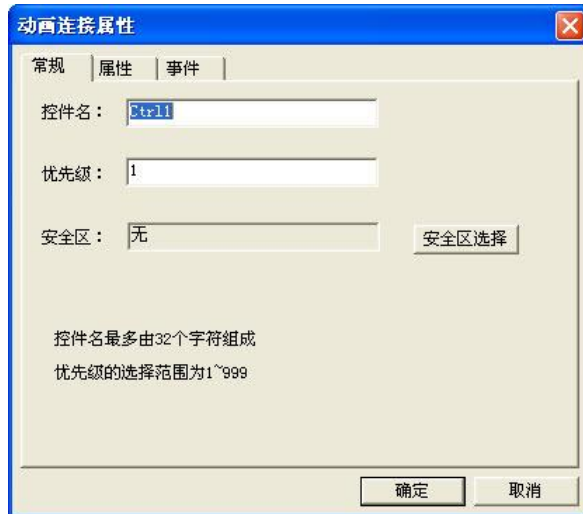


图14.120 动画连接属性对话框—常规页

动画连接属性的具体含义、操作请参加《组态王7.06.5使用手册》“控件”一章中关于Active X控件的描述。

14.3.6.3.2.2 动画连接属性页—属性页

组态王7.0温控版温控曲线中，实时曲线变量与控件的关联是通过温控曲线的控件动画连接属性实现的。打开温控曲线动画连接属性对话框，单击“属性”页，如图14.121所示。属性列表中Plot1~Plot20为实时曲线属性，用户可以直接在这里关联实时曲线变量。



图14.121 动画连接属性对话框—属性

这里关联的属性固定为实型，所以使用这种方法只能关联实型变量。如果用户使用到的必须是其它类型的变量，可以使用命令语言程序将变量的值直接赋给控件的该属性。

控件其它属性

属性名称	关联变量 类型	说明
MainWnd	LONG	控件窗口指针
ShowTitle	BOOL	是否显示标题，1表示显示，0表示不显示
Title	STRING	控件标题文本
LegendType	LONG	图例显示方式： 0.None ; 1.ThinBorder; 2.ClientEdge; 3.StaticEdge

ShowToolBar	BOOL	是否显示工具条：1，显示；0，不显示
ShowGrid	BOOL	是否显示网络：1，显示；0，不显示
EnableContextMenu	BOOL	是否弹出右键菜单：1，弹出；0，不弹出
ShowMark	BOOL	是否显示标记：1，显示；0，不显示
ShowSlide	BOOL	是否显示游标：1，显示；0，不显示
ControlPeriod	FLOAT	控制周期
ShowSetCurve	BOOL	是否显示设定曲线：1，显示；0，不显示
SetCurveLineStyle	LONG	设定曲线线型
TimeUnit	LONG	时间单位：秒，分，时
TimeType	LONG	时间类型：相对时间，绝对时间
TimeFormat	LONG	时间格式：0，ss；1，mm:ss；2，hh:mm:ss； 3，mm；4，hh:mm
UpdateTime	FLOAT	更新周期，秒
MaxYAxisValue	FLOAT	Y轴最大值
MinYAxisValue	FLOAT	Y轴最小值
ShowXAxisTitle	BOOL	是否显示X轴标题：1是，0否
ShowYAxisTitle	BOOL	是否显示Y轴标题1是，0否
XaxisUnit	STRING	X轴单位
YaxisUnit	STRING	Y轴单位

XaxisMainGrids	LONG	X轴主分度数
XaxisSecondGrids	LONG	X轴次分度数
XaxisLineStyle	LONG	X轴坐标线型
YaxisLineStyle	LONG	Y轴坐标线型
MainGridLineStyle	LONG	主网格线型
SecondGridLineStyle	LONG	次网格线型
YaxisMainGrids	LONG	Y轴次分度数
XaxisTitle	STRING	X轴标题
YaxisTitle	STRING	Y轴标题
ControlFieldName	STRING	控制段名称
ShowRightSlide	BOOL	是否显示右游标
ShowDate	BOOL	是否显示日期：1是，0否
DateFormat	LONG	日期格式：0, DD; 1, MM-DD; 2, YY-MM-DD; 3, YYYY-MM-DD
SetCurveLineLadder	BOOL	设定曲线是否显示为梯阶线：1是；0否
MaxControlField	SHORT	最大控制段数

14.3.6.3.2.2 动画连接属性页—事件页

控件事件函数

温控曲线包含4个事件函数：

1、RunStatusChange()；

函数功能：当修改温控曲线的运行状态（启动、停止或重新启动）时产生该事件。

2、ControlFieldChange()；

函数功能：当修改温控曲线关联的设定曲线控制字段时产生该事件。

3、LVernierTimeChange(long lTime) ；

保留，目前未使用。

4、MaxControlFieldChange(short nMaxControlField, short nInRange)；

函数功能：当修改温控曲线关联的最大设定曲线控制段时产生该事件。

14.3.6.3.3 设置温控曲线的固有属性

选中画面上创建后的控件，单击鼠标右键，在弹出的快捷菜单中选择“控件属性”，系统弹出曲线的固有属性对话框，如图14.122所示。

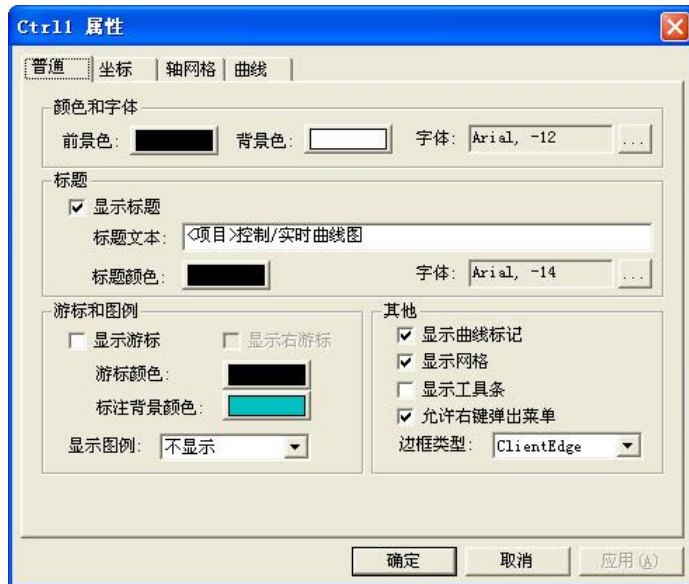


图14.122 温控曲线的固有属性页—普通页

1、普通属性页各项配置：

- ◇ **颜色和字体：**设置曲线的背景颜色、曲线图表上字符标识和分割线颜色（前景颜色）。直接单击颜色和字体按钮，设置颜色和字体。
- ◇ **标题：**为曲线图标添加一个标题。可以对标题的颜色和字体进行相应的设置。
- ◇ **游标和图例：**选择是否显示游标和设置游标颜色及标注背景颜色；设置图例在画面中的位置。
- ◇ **其他：**画面中显示的具体项目的设置。

2、坐标属性页各项配置：

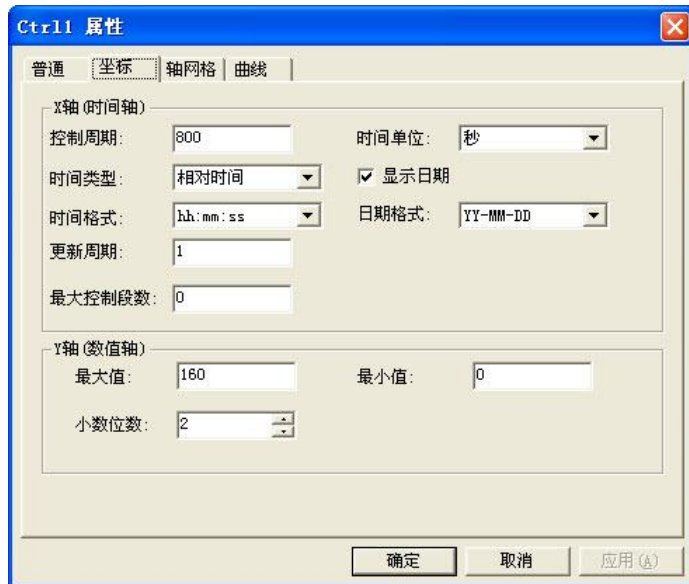


图14.123坐标属性页

- ◇ **控制周期:** 温控曲线执行的一个控制周期的时间长度，为曲线横坐标最大值。如果在控制参数表中设置了“时间”字段，曲线启动后，将根据“时间字段”来调整控制周期。
- ◇ **时间单位:** 选择控制周期的时间单位。
- ◇ **时间类型:** 包括相对时间和绝对时间。相对时间：以实际控制周期为横坐标，当曲线暂停后，再次启动曲线时，曲线的绘制点从原来的暂停点开始。绝对时间：以控制周期的绝对时间为横坐标，如果运行的绝对时间没有超出控制周期时，曲线绘制的数据点不能从原来暂停点开始，而是按照绝对时间的坐标到新的坐标点开始绘制，直至控制周期结束。

- ◇ **显示日期:** 表示在温控曲线的X轴是否显示日期。
- ◇ **时间格式:** 提供多种时间格式供选择, 时间类型不同时, 格式表示的意义有所差异, 如19: 45: 30, 当你选择的是绝对时间, 则次数值是个时间, 即下午7: 45; 如果选择相对时间, 就表示17小时45分钟30秒, 是个时间长度。
- ◇ **日期格式:** 提供多种日期格式供选择, 时间类型不同时, 格式表示的意义有所差异, 和上面的时期格式相类似。
- ◇ **更新周期:** 曲线更新数据的时间频率, 单位为秒。
- ◇ **最大控制段数:** 输入需要执行的最大控制段数, 此处输入的控制段数决定了温控曲线中显示的曲线控制段, 当输入的值超出表格控件中实际的控制段时, 温控曲线将会按照实际的表格控件的控制段数显示。
- ◇ **最大值: Y轴最大值。**
- ◇ **最小值: Y轴最小值。**
- ◇ **小数位数:** Y轴数值可以显示的小数位数。

3、轴网格属性页各项配置:

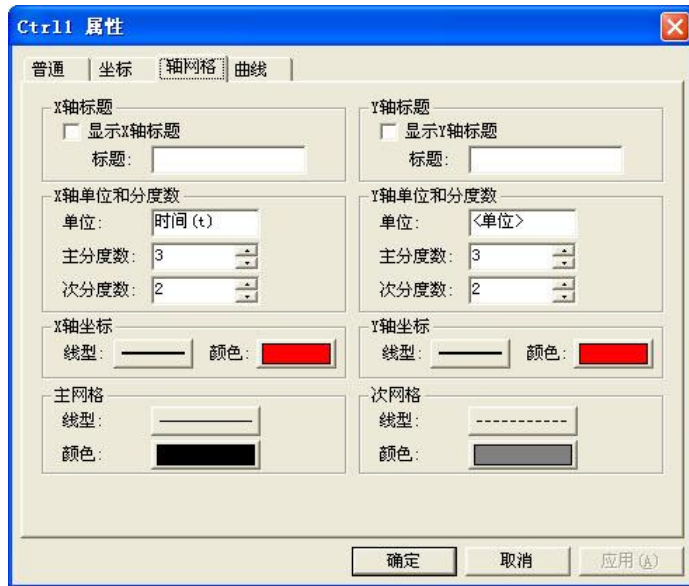


图14.124 温控曲线的固有属性页—轴网格

- ◇ **显示X、Y轴标题选项：** 是否显示轴标题及标题内容。
- ◇ **X、Y轴单位和分度数：** 表示轴单位标识， X方向和Y方向的网格的主、次分度数。
- ◇ **线型：** 分别可以设置X轴坐标、Y轴坐标， 主网格、次网格的线型。
- ◇ **颜色：** 分别可以设置X轴坐标、Y轴坐标， 主网格、次网格的颜色。

4、曲线属性页各项配置：

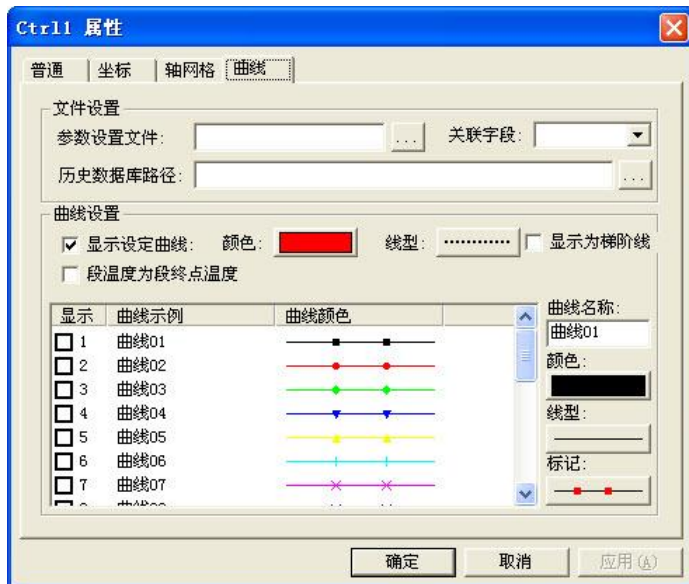


图14.125 温控曲线的固有属性页—曲线

- ◇ **参数设置文件：**设置参数数据库文件所在的路径和文件名。
- ◇ **关联字段：**选择设定曲线关联的参数字段。
- ◇ **历史数据库路径：**设定曲线图表保存历史数据的路径，要指定绝对路径，如：E:\历史数据\……，系统会自动在该路径下生成一个历史数据文件，每一个控制周期生成一个文件，文件的名称是控制周期起始时期和时间组成。
- ◇ **显示设定曲线：**选择是否在控件运行过程中显示设定曲线。
- ◇ **显示为阶梯线：**指设定曲线是否以阶梯方式显示。
- ◇ **曲线名称：**设定左侧列表框中选定的实时曲线的名称。
- ◇ **标记：**设定左侧列表框中选定的实时曲线的标记。只有关联了变量的曲线

在列表框中的设置才是有意义的。

14.3.6.3.4 温控曲线的使用

温控曲线设定完成后，就可以启动运行了。对应运行系统中温控曲线的操作可以使用曲线工具条，也可使用控件提供的方法、属性等通过命令语言对控件进行操作。直接使用工具条会比较方便一些。

14.3.6.3.4.1 温控曲线工具条和右键菜单使用方法

如果在温控曲线“控件属性”配置中选择了“显示工具条”，则在系统运行时可以使用工具条对温控曲线进行操作，温控曲线如图14.126所示。

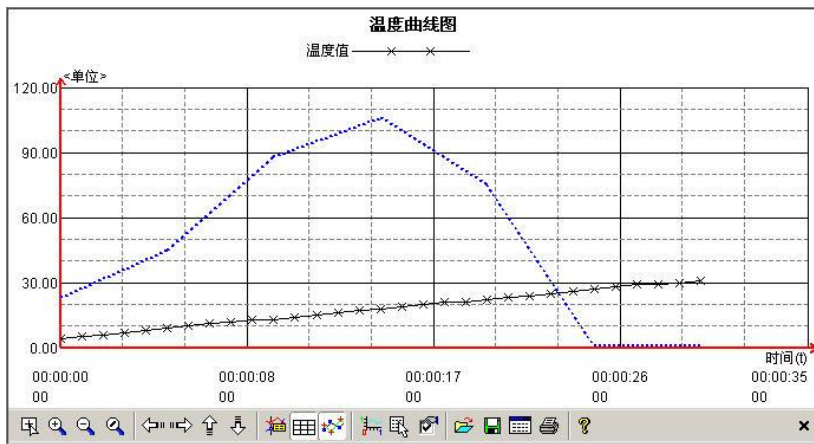


图14.126 温控曲线

操作人员可以利用曲线工具条功能对曲线进行属性修改、无级缩放、移动、保存、打印等操作。



查看历史数据：查看已保存的实时曲线的历史数据库曲线。点击该按钮，系统弹出路径和文件选择对话框，选择需要查看的文件即可。只有当控制周期执行完成后或者曲线在暂停时才可以查询历史数据。运行过程中，当选择绝对时间的控制方式，则只能查看绝对时间控制过程中的历史数据；当选择相对时间的控制方式，则只能查看相对时间控制过程中的历史数据。其他按钮功能请参考“超级X-Y曲线的使用”。

运行系统中，在温控曲线上单击鼠标右键，系统弹出右键快捷菜单，如图14.127所示。

其功能与上述工具条中对应功能相同。

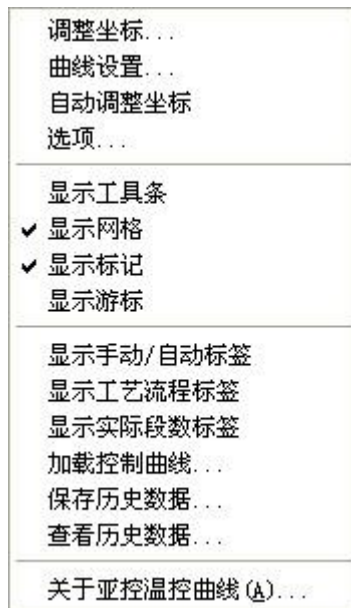


图14.127工具条快捷菜单

14.3.6.3.4.2 温控曲线控件方法

Start()

用于在运行系统中启动温控曲线，使其开始绘制实时趋势曲线。

Stop()

用于在运行系统中暂停温控曲线，使其暂时停止绘制温控曲线。如果再使用stop()函数后，系统会自动计算出上次停止到本次启动之间的时间差，从当前对应的坐标点开始绘制曲线。

ReStart()

重新启动温控曲线，重新开始一个控制周期，不记录上次运行数据。

long GetRunStatus()

返回曲线运行状态。

返回值： 整型

0：未启动温控曲线；

1：温控曲线运行中；

2：温控曲线已停止

Long GetCurrentControlField()

得到温控曲线当前控件执行的段号。该返回值可以直接关联到表格控件的nControlID属性上，用来在表格控件中显示当前的控制段，也可以确定哪些段在控制状态而不被修改。

返回值： 整型

float GetSetValue(long lTime, short nOption)

获得设定曲线在某个时间点上的值。

参数: lTime 整型 相对于坐标原点处的时间值, 单位为秒。

nOption 整型 选项: 0: 根据设曲线设置段的起点和终点线性算出lTime处的值 1:

返回曲线设置段的终点值。

返回值: 实型

BOOL SetControlField(STRING szFilename, STRING szFieldName)

指定数据库文件中指定的字段为温控曲线的设定曲线字段。

参数: szFilename为字符串型, 参数数据库文件的路径;

szFieldName为字符串型, 参数数据库文件中的字段名称。

返回值: 离散型。

- 1, 表示加载成功;
- 2, 0, 表示加载失败

BOOL SetHistoryDbFile(STRING szFilename)

设置曲线历史库路径, 要求此路径在系统中已经存在。

参数: szFilename为字符串型, 历史数据数据库文件的路径

返回值: 离散型。

- 1, 表示设置成功;
- 2, 0, 表示设置失败



例:

在数据词典中定义离散变量aa，画面中建立按钮，在弹起按钮中输入：

```
aa= BOOL SetHistoryDbFile("D:\peixun\新\温控曲线");
```

Print()

用于在运行的过程中打印曲线。

14.3.6.3.5 温控曲线与表格控件的同步关联

在表格控件属性中，包含一个重要属性nControlID，它的重要作用是通知表格控件执行到哪一个控制段，而通过调用温控曲线控件的方法GetCurrentControlField()可以得到当前执行段。而后和nControlID关联起来。

例如：在温控曲线ControlFieldChange()的事件中定义如下事件函数：

```
\\本站点\CtrlID=KvTempCtrl.GetCurrentControlField();
```

然后在表格控件的属性中关联内存变量：CtrlID。在系统运行后出现效果如下图所示。已执行和正在执行的控制段参数在控制参数表格中将不允许再被编辑，表格中相应的行的底色变为灰色。只有当控制结束后才会恢复正常。

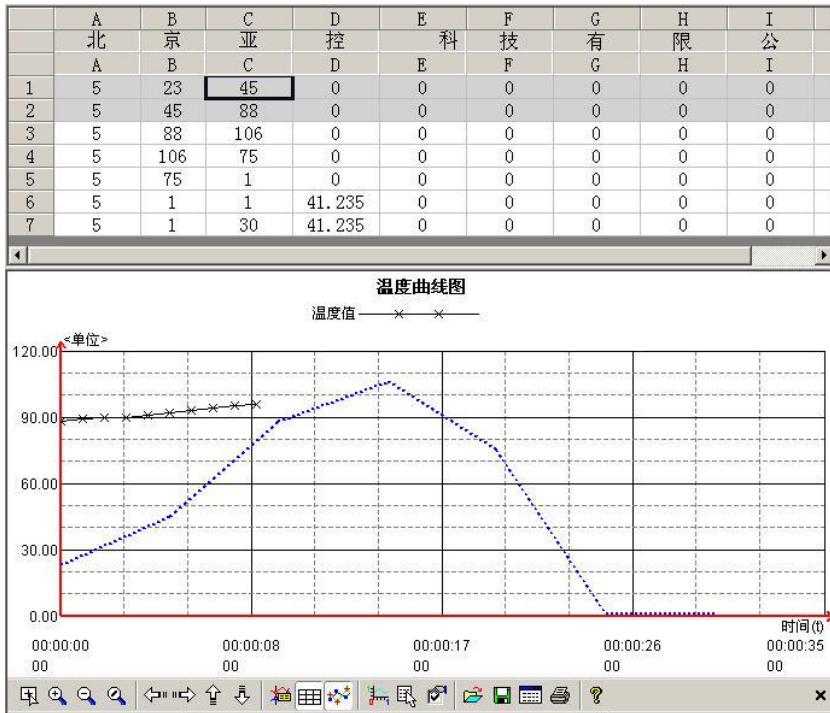


图14.128 温控曲线与表格控件的同步关联

14.3.7 旋转装置控件

14.3.7.1 控件简介

使用该控件，可以在组态王 7.0 运行系统的画面中动态显示旋转装置的旋转状态；可以方便的控制旋转装置的起停和旋转速度。

14.3.7.2 旋转效果配置文件说明

在程序文件夹中，有一个名为 KvLqmRevolverCtrl.ini 的文件，该文件为旋转效果配

置文件。其中：KvLqmRevolverCtrl：标题名称。不允许用户修改。

- ◇ **时间间隔：**旋转速度控制参数，可设置范围为 10—3000，数值越小旋转速度越快，默认为 20。为了达到比较好的视觉效果，建议将该参数设在 10-100 之间。
- ◇ **正常旋转位图文件数：**显示的正常旋转位图的文件个数，可设置范围为 2—6，默认为 6。
- ◇ **正常旋转位图文件 0—正常旋转位图文件 5：**正常旋转状态下时显示的不同位图文件，用户可修改该位图文件。
- ◇ **故障旋转位图文件数：**显示的故障旋转位图的文件个数。建议与正常旋转位图文件数设置相同。
- ◇ **故障旋转位图文件 0—故障旋转位图文件 5：**故障旋转状态下显示的不同位图文件。用户可修改该位图文件。6 个正常旋转位图依次相差一个相同角度，用户可以根据需要调整正常旋转位图文件数和正常旋转位图文件的顺序，配以适当的时间间隔，以达到不同的旋转效果。故障旋转位图亦如此。

 注意：

在 KvLqmRevolverCtrl.ini 的文件中，“=”号左边的部分不容许用户修改。旋转装置显示控件消耗内存的数量决定于位图文件数量和位图尺寸，为了降低内存消耗，建议位图尺寸在 128*128 象素点以下。

14.3.7.3 控件使用方法

在组态王 7.0 画面开发系统的工具箱中点击“插入通用控件”按钮或选择“编辑”菜单下的“插入通用控件”，在通用控件列表中选择 KvLqmRevolver Control，在画面上放置该控件。如果 KvLqmRevolverCtrl.ini 中的设置有错误，放置控件时，画面上不会显示该控件，组态王 7.0 信息窗口会有错误信息的提示。

在组态王 7.0 开发中仅显示该控件的正常静止状态。

可以通过配置控件的属性页来指定配置文件（使用的位图和 KvLqmRevolverCtrl.ini）所在的位置，以改变控件的属性。

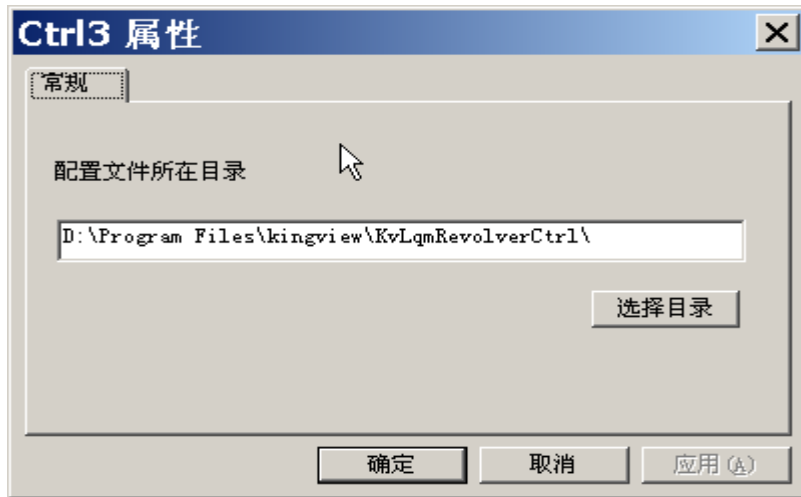


图14.129 配置文件属性

该控件有两个属性可以用来关联组态王 7.0 的变量：

- ◇ is_bad：表示是否是故障状态。当 is_bad=0 时该控件显示正常位图；当 is_bad=1 时 KvLqmRevolverCtrl 显示故障位图。
- ◇ is_rotate：表示是否旋转。当 is_rotate=1 时该控件以 KvLqmRevolverCtrl.ini 文件中指定的时间间隔和位图文件数依次显示位图，产生旋转效果。

下面举例说明该控件的使用方法。KvLqmRevolverCtrl.ini 使用默认配置。

例：

组态王 7.0 数据词典中定义 I0 整形变量“bad”和“rotate”，“bad”表示是否处于故障状态，“rotate”控制装置是否旋转。在组态王 7.0 画面中放置

KvLqmRevolver Control 控件，鼠标双击该控件，在弹出的“动画连接属性”界面的“属性”页中关联这两个变量，如下图所示。



图14.130 旋转控件动画连接属性

组态王 7.0 运行系统中：

当变量 rotate 的值为 1 时，装置旋转：如果变量 bad 的值为 0，显示 KvLqmRevolverCtrl.ini 文件中的正常旋转状态位图，如果 bad 的值为 1，显示 KvLqmRevolverCtrl.ini 文件中的故障旋转状态位图。当变量 rotate 的值为 0 时，停止旋转。

14.3.8 PID 控件

14.3.8.1 控件简介

KingviewPid 控件是组态王 7.0 提供的用于对过程量进行闭环控制的专用控件。通过该控件，用户可以方便的制作 PID 控制。

- ✧ pid 控制算法：标准型, 分为增量型输出和反向作用。
- ✧ 显示过程变量的精确值，显示范围[-999999.99~999999.99]。
- ✧ 以百分比显示设定值（SP）、实际值（PV）和手动设定值（M）。

- ◇ 开发状态下可设置控件的总体属性、设定/反馈范围和参数设定。
- ◇ 运行状态下可设置PID参数和手动自动切换。

14.3.8.2 如何创建PID控件

在画面中插入控件：组态王 7.0 画面菜单中编辑\插入通用控件，或在工具箱中单击“插入通用控件”按钮，在弹出的对话框中选择“Kingview Pid Control”，单击确定。

按下鼠标左键，并拖动，在画面上绘制出表格区域。

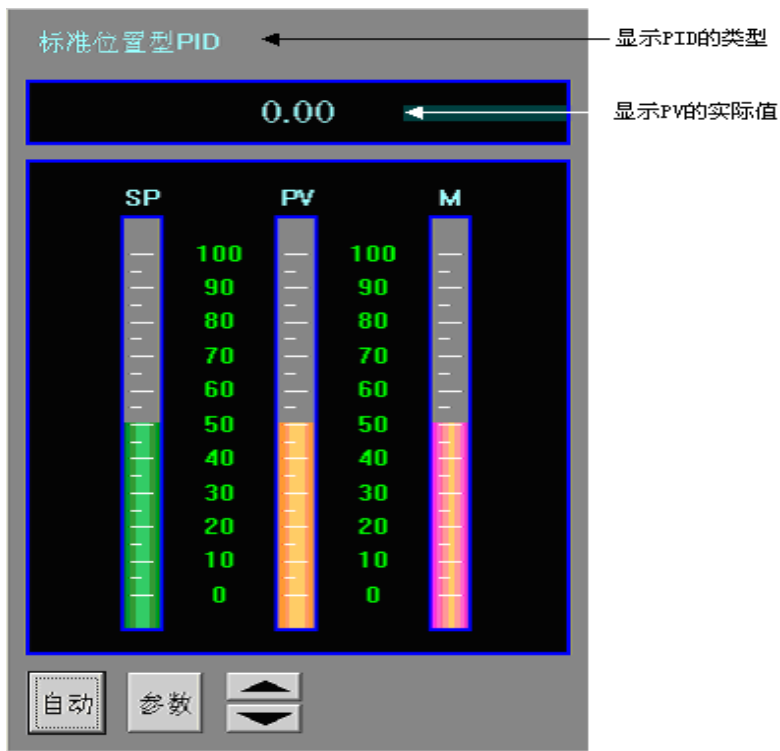


图14.131 PID控件界面

14.3.8.3 如何定义 PID 属性

1、定义动画连接属性

双击控件或选择右键菜单中动画连接，在弹出的属性页中设置控件名称等信息。

◇ 常规属性：

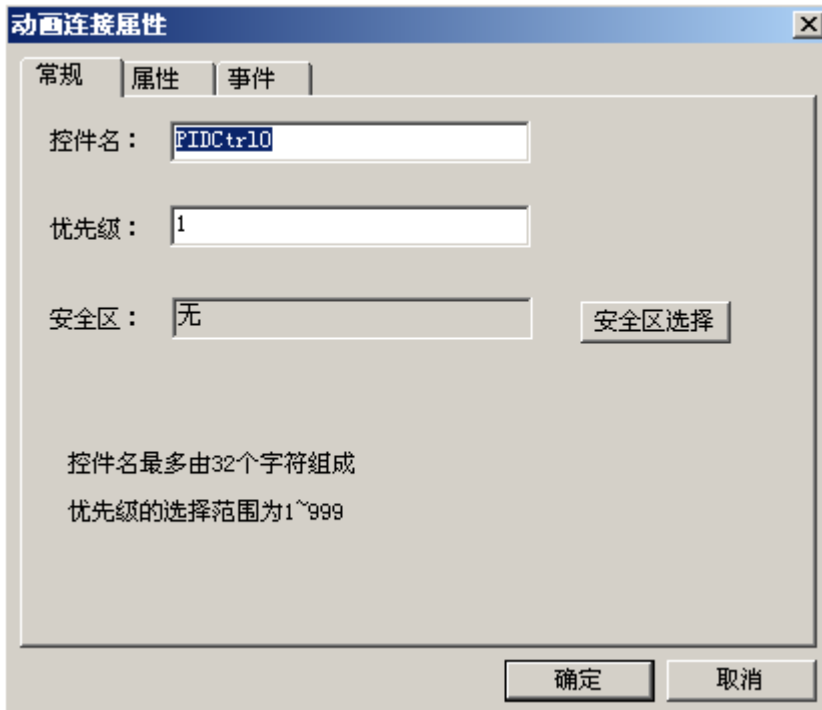


图14.132 PID控件动画连接常规界面

- **控件名称：**应符合组态王 7.0 中关于名称定义的规定，例如：PIDCtr10。
- **优先级：**是控件的操作优先级，范围在 1~999。
- **安全区：**安全区只允许选择。

◇ 关联属性:

属性	类型	关联变量
SP	FLOAT	<->\\本站点\sp
PV	FLOAT	<->\\本站点\pv
YOUT	FLOAT	<->\\本站点\du
Type	LONG	
CtrlPeriod	LONG	
FeedbackFilter	BOOL	
FilterTime	LONG	
CtrlLimitHigh	FLOAT	
CtrlLimitLow	FLOAT	
InputHigh	FLOAT	
InputLow	FLOAT	
OutputHigh	FLOAT	
OutputLow	FLOAT	
Kp	FLOAT	
Ti	LONG	
Td	LONG	
Tf	LONG	
ReverseEffect	BOOL	
IncrementOutput	BOOL	

图14.133 PID动画连接关联属性界面

- SP: FLOAT, 控制器的设定值。
- PV: FLOAT, 控制器的反馈值。
- YOUT: FLOAT, 控制器的输出值。
- Type: LONG, PID 的类型。
- CtrlPeriod: LONG, 控制周期。
- FeedbackFilter: BOOL, 反馈加入滤波。
- FilterTime: LONG, 滤波时间常数。

- CtrlLimitHigh: FLOAT, 控制量高限。
- CtrlLimitLow: FLOAT, 控制量低限。
- InputHigh: FLOAT, 设定值 SP 的高限。
- InputLow: FLOAT, 设定值 SP 的低限。
- OutputHigh: FLOAT, 反馈值 PV 的高限。
- OutputLow: FLOAT, 反馈值 PV 的低限。
- Kp: FLOAT, 比例系数。
- Ti: LONG, 积分时间常数。
- Td: LONG, 微分时间常数。
- Tf: LONG, 滤波时间常数。
- ReverseEffect: BOOL, 反向作用。
- IncrementOutput: BOOL, 是否增量型输出。
- DeadBandLow: Long, 无效。
- Status: BOOL, 手自动状态。
- M: FLOAT, 手动设定值。
- PercentRange: float, 手动时调节的调节幅度, 默认是 1, (可以在运行时, 点击参数按钮在手动调节比率里面调节此参数)。

注意:

在使用变量关联时，只有控件所处的画面处于激活状态，控制功能才会执行。

2、定义控件属性

选择控件右键菜单中“控件属性”。弹出控件固有属性页，可分别设置如下属性：

◇ 总体属性：

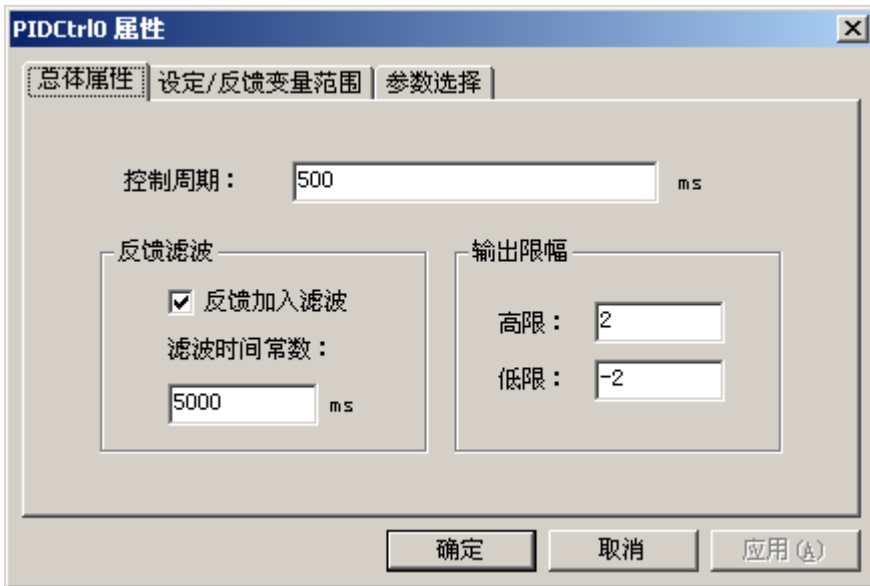


图14.134 PID控件总体属性界面

- **控制周期：** PID 的控制周期，为大于 100 的整数。且控制周期必须大于系统的采样周期。
- **反馈滤波：** Pv 值在加入到 PID 调节器之前可以加入一个低通滤波器。
- **输出限幅：** 控制器的输出限幅 Yout 的值。

◇ 设定/反馈变量范围:

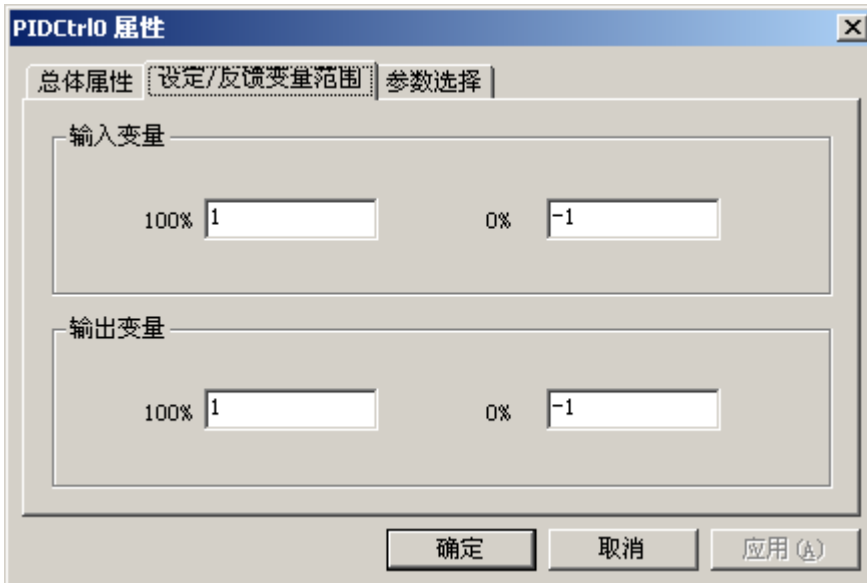


图14.135 PID控件设定/反馈变量范围属性界面

- **输入变量:** 设定值 sp 或者反馈值 pv 对应的最大值 (100%) 和最小值 (0%) 的实际值。设定值 sp 与反馈值 pv 一般最大值、最小值相同。
- **输出变量:** 输出值 Yout 对应的最大值 (100%) 和最小值 (0%) 的实际值

◇ 参数选择:

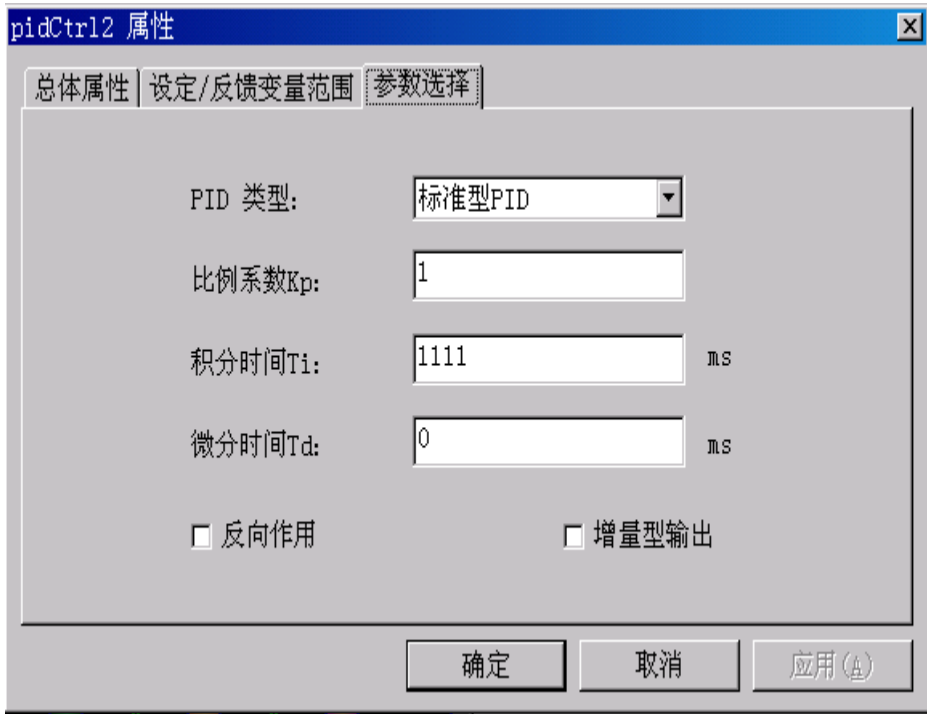


图14.136 PID控件参数选择属性界面

- PID 类型：选择使用标准型。
- 增量型输出：控制器输出为增量型。
- 反向作用：输出值取反。
- 比例系数 K_p ：设定比例系数。一般取值范围：1-10。
- 微分时间 T_d ：设微分时间常数，即对相同输出调节量，微分项超前于比例项响应的的时间，一般取值为 0。

- 积分时间 T_i : 设定积分时间常数, 就是积分项的输出量每增加与比例项输出量相等的值所需要的时间。一般取值范围: 1000-5000ms。

注意:

增量型对象暂时不易实现, 控制周期不能低于采样周期。

14.3.8.4 如何使用 PID 控件

1、命令语言中的使用

- ◇ 在画面命令语言中使用

此时, 只有控件所处的画面处于激活状态, 控制功能才会执行, 如果工程中存在多个画面, 并且 PID 控件画面并不总是处于激活状态, 则应该采用命令语言的方式使用 PID 控件。即, 在控件所处画面的画面命令语言中, 使用赋值的方式, 显示地交换 PID 控制值。选择画面命令语言中的控件, 如下所示:

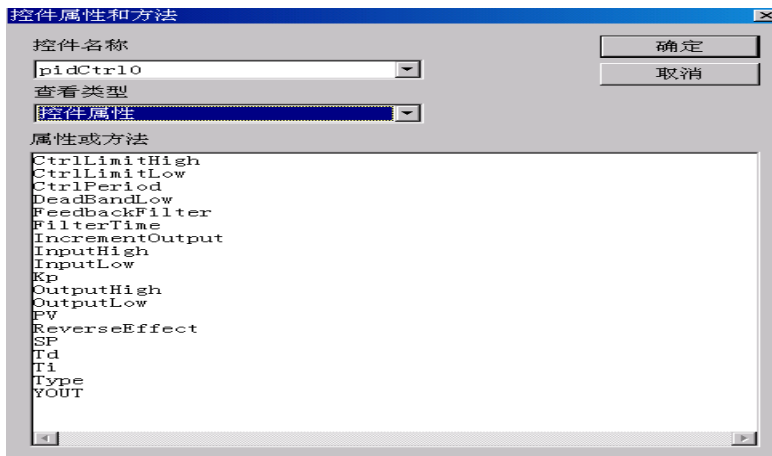


图14.137 控件和属性列表

在属性或方法中选择相应的选项，在存在时出现命令语言，如我们选择 SP 双击，则如下显示：

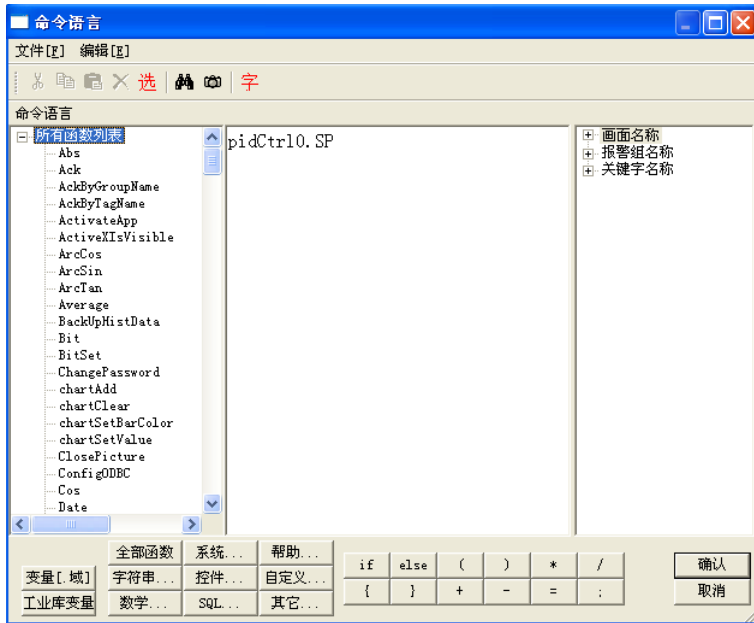


图14.138 编写画面命令语言

- 显示时：当画面由隐含变为显示时，则“显示时”编辑框中的命令语言就被执行一次。
- 存在时：只要该画面存在，则“存在时”编辑框中的命令语言就反复按照设定的时间周期执行。
- 隐含时：当画面由显示变为隐含或关闭时，则“隐含时”编辑框中的命令语言

就被执行一次。

输入命令语言，如下：

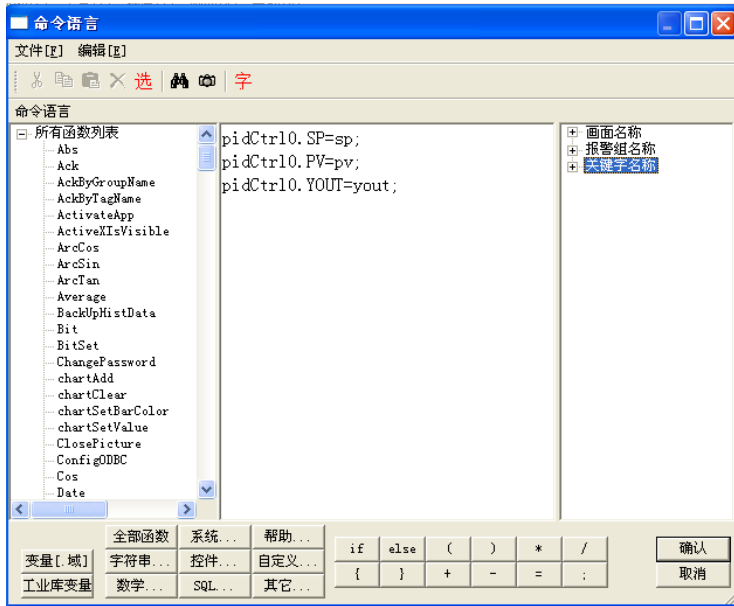


图14.139 编写存在时脚本

其中，SP 为设定变量，PV 为反馈变量，YOUT 为控制器输出变量。

✧ 在使用工程浏览器的应用程序命令语言

应用程序语言可以在程序启动时执行、关闭时执行或在程序运行期间定期执行。

应用程序命令语言的运行时编写同上面。程序如下：

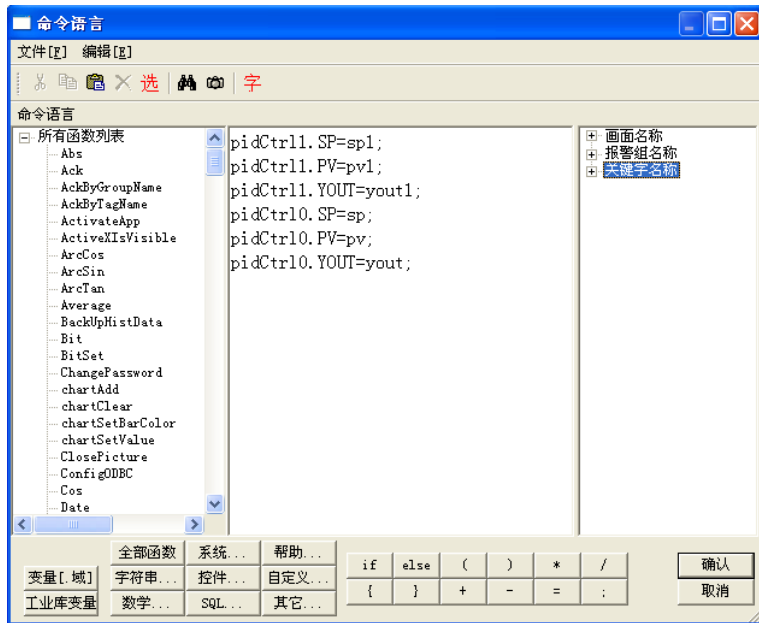


图 14.140 应用程序命令语言

2、运行时操作

- 手动/自动，自动时，控制器调节作用投入。手动时，控制器输出为手动设定值经过量程转换后的实际值。手动设定为 M，是 YOUT 的值。
- 手动值设定（上/下），每次点击手动设定值增加/减少 1%

3、运行时的参数设置

如图下图所示。

- 标准型 PID 参数：比例系数、积分常数、微分常数，PID 的常规参数
- 反向作用：输出值取反

- 手动情况下设定手动调节比率：

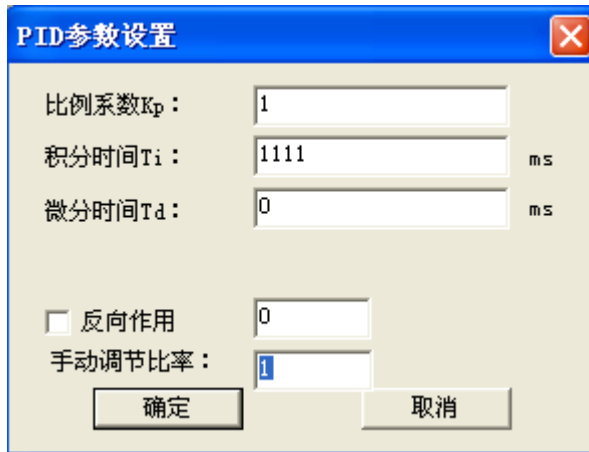


图14.141 PID控件设定/反馈变量范围属性界面

14.3.9 位图控件

14.3.9.1 控件简介

位图控件用来在组态王 7.0 画面上显示位图（jpg, bmp, wmf 等格式）。组态王 7.0 运行过程中通过切换图片文件路径的连接可以显示不同的位图。

14.3.9.2 使用方法

在组态王 7.0 画面开发系统的工具箱中点击“插入通用控件”按钮或选择“编辑”菜单下的“插入通用控件”，在通用控件列表中选择 kvimageocx Control，在画面上放置该控件。

可以通过配置控件的常规属性页来指定显示的位图文件。选择控件右键菜单中的“控

件属性”，如下图所示，选择显示的位图文件。

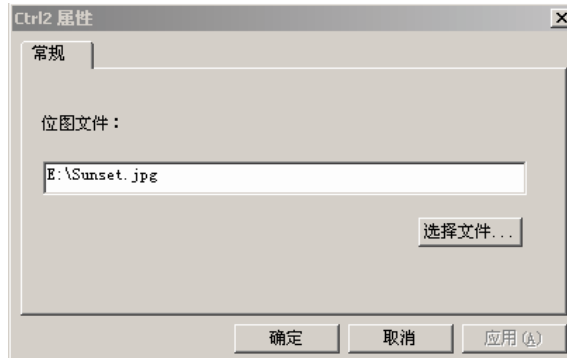


图14.142 控件属性

组态王 7.0 运行，控件显示选择的位图。如果需要在组态王 7.0 运行系统中更换显示的位图文件，执行如下操作。在组态王 7.0 开发系统下，双击画面上的 kvimageocx 控件，系统弹出“动画连接属性”对话框。在“属性”页中提供了“filename”属性（this_ocx 属性为程序内部使用，用户不要使用），该属性用于设置要显示的位图的路径和文件名。将该属性关联组态王 7.0 字符串变量，如下图所示，关联组态王 7.0 变量 FileName。

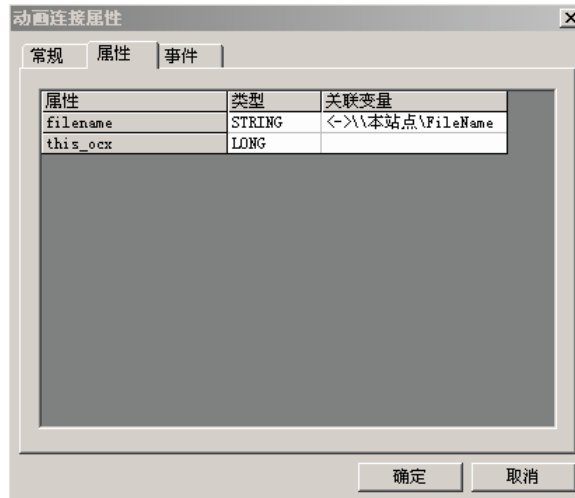


图14.143 控件属性

在组态王 7.0 运行系统中修改该“filename”属性关联的组态王 7.0 字符串变量的值，可以显示不同的位图。

例如：

组态王 7.0 数据词典中定义内存字符串变量：FileName。在画面上放置位图显示控件。如图 14.143 所示，该控件“filename”属性关联组态王 7.0 变量“FileName”。组态王画面上放置按钮 1，按钮命令语言中输入：

```
\\本站点\ FileName = " E:\pic1.jpg "；
```

组态王 7.0 画面上放置按钮 2，按钮命令语言中输入：

```
\\本站点\ FileName = " F:\pic2.bmp "；
```

运行组态王 7.0，点击按钮 1，控件显示位图 pic1.jpg；点击按钮 2， 控件显示位图 pic2.bmp。

第十五章 配方管理

- ☞ 介绍配方的相关知识
- ☞ 教您如何制作及使用组态王 7.0 配方

15.1 配方管理概述

15.1.1 什么是配方

什么是配方？在制造领域配方是用来描述生产一件产品所用的不同配料之间的比例关系。配方是生产过程中一些变量对应的参数设定值的集合。例如，一个面包厂生产面包时有一个基本的配料配方，此配方列出所有要用来生产面包的配料成份表（如水，面粉，糖，鸡蛋，香油等）。另外，也列出所有可选配料成份表（如果酱，维生素，巧克力等），而这些可选配料成份可以被添加到基本配方中用以生产各种各样的面包。下表为某一面包厂生产面包时的配方：

	配方 1	配方 2	配方 3
配料名	果酱面包	巧克力面包	维生素面包
水	200 克	200 克	200 克

面粉	4500 克	4500 克	4500 克
盐	325 克	325 克	325 克
糖	500 克	500 克	500 克
鸡蛋	10 个	10 个	10 个
香油	300 克	300 克	300 克
水果	5 个	0	0
巧克力	0	500 克	0

注：表中的数字是示意性的。

又如，在钢铁厂，一个配方可能就是机器设置参数的一个集合，而对于批处理器，一个配方可能被用来描述批处理过程中的不同步骤。组态王 7.0 支持对配方的管理，用户利用此功能可以在控制生产过程中得心应手，提高效率。比如当生产过程状态需要大量的控制变量参数时，如果一个接一个地设置这些变量参数就会耽误时间，而使用配方，则可以一次设置大量的控制变量参数，满足生产过程的需要。

15.1.2 组态王 7.0 中的配方管理

组态王 7.0 提供的配方管理由两部分组成：配方管理器和配方函数集。

◇ 配方管理器

配方管理器打开后，弹出对话框，用于创建和维护配方模板文件。所有配方都在配方模板文件中定义和存储，每一个配方模板文件以扩展名为 csv 的文件格式存储，一个配方模板文件是通过配方定义模板产生的。配方定义模板用于定义配方中的所有项目名（即配料名）、项目类型、数据变量（与每一个项目名对应）、配方名。每

一个配方指定每一个配料成份所要求的数量大小。

配方定义模板的结构如下：

项目名	变量名	变量类型（项目类型）	配方 1	配方 2	配方 M
配料 1	变量 1	实数型、整数型、离散型或字符串型	11	21	M1
配料 2	变量 2	实数型、整数型、离散型或字符串型	12	22	M2
配料 3	变量 3	实数型、整数型、离散型或字符串型	13	23	M3
配料 4	变量 4	实数型、整数型、离散型或字符串型	14	24	M4
配料 N	变量 N	实数型、整数型、离散型或字符串型	1N	2N	MN

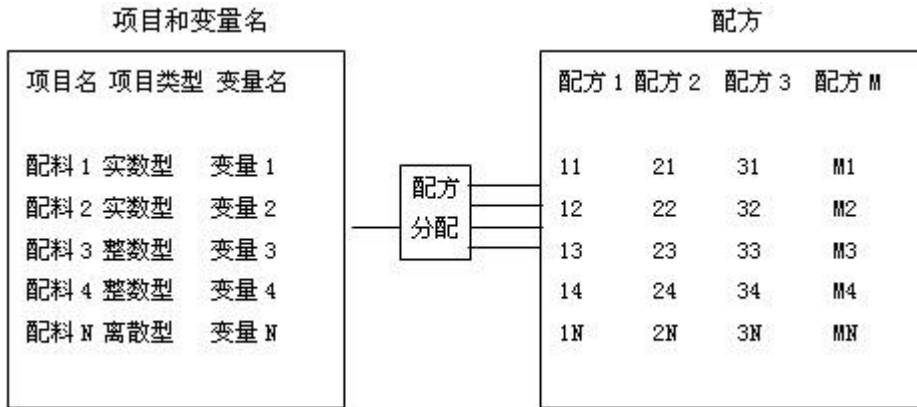
注：配方名由用户设定，具体的配方数表示配料的数量大小，此处的数字是示意性的，并不表示具体意义。

◇ 配方函数

配方函数允许组态王 7.0 运行时对包含在配方模板文件中的各种配方进行选择，修改，创建和删除等一系列操作。

15.1.3 配方的工作原理

配方模板文件中的配方定义模板完成后，在组态王 7.0 运行时可以通过配方函数进行各种配方的调入，修改等。其工作原理结构示意图如下：



配方分配的功能由配方函数来完成，通过配方分配将指定配方(如配方 M)传递到相应的变量中。当调配方 1 时，则配方 1 的数据值 11、12、13、14、1N 分别对应地传送给变量 1、变量 2、变量 3、变量 4、变量 N；同理，当调用配方 M 时，则同样是把配方 M 数据值传送给变量 1、变量 2、变量 3、变量 4、变量 N。

15.2 如何创建配方模板

组态王 7.0 的工程浏览器能够创建和管理配方模板文件，在工程浏览器的目录显示区中，选中大纲项“文件”下的成员“配方”，如图 15.1 所示。

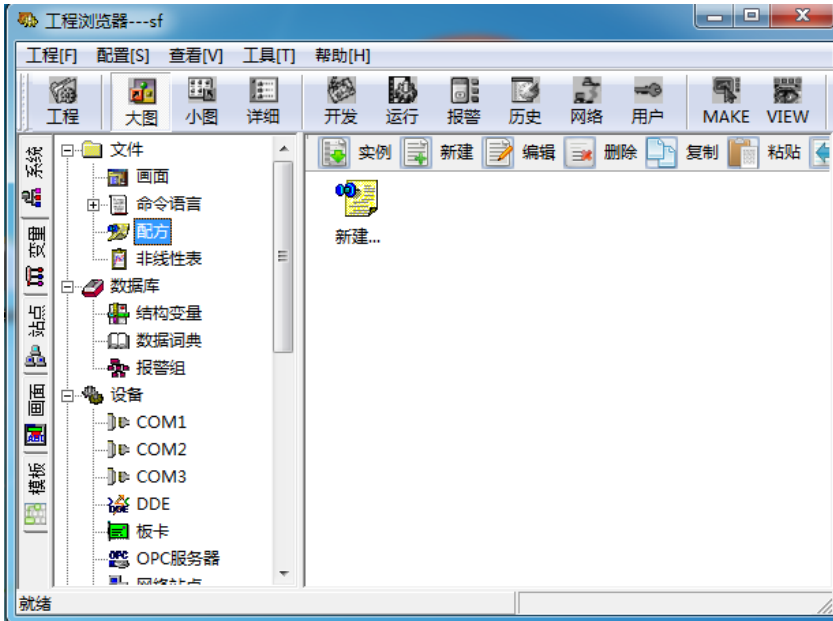


图 15.1 新建配方

内容显示区中用左键双击“新建”图标，或者右键单击“新建”图标，从浮动式菜单中选择命令“新建配方”，则弹出“配方定义”对话框，如图 15.2 所示。



图 15.2 配方定义



注意：

配方定义对话框中的第一行中的第一列和第二列是不可操作的，即不能在这两个单元格中输入任何内容。

15.2.1 菜单栏

15.2.1.1 表格菜单

单击“表格”菜单，弹出下拉式菜单，如图 15.3 所示。



图 15.3

- ◇ **增加行：**选择此键，在输入焦点所在行的位置上面增加一行。
- ◇ **删除行：**选择此键，则删除输入焦点所在的行。
- ◇ **增加列：**选择此键，在输入焦点所在列的位置前面增加一列。
- ◇ **删除列：**选择此键，则删除输入焦点所在的列。
- ◇ **保存：**将配方文件保存到当前工程路径下。对于新建的配方第一次进行保存时，文件默认路径为当前工程路径，如图 15.4 所示。

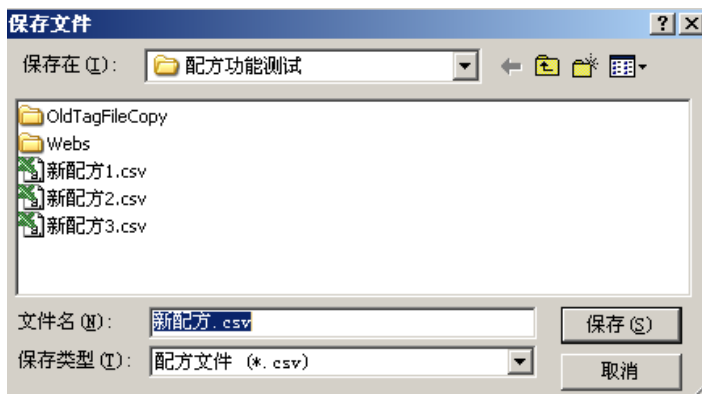


图 15.4 保存配方

- ◇ **另存为:** 将配方文件保存到指定的路径下。当执行该操作时，系统弹出另存为对话框，如图 15.5 所示。选择文件保存路径后，单击“保存”按钮。



图 15.5 另存配方文件

- ◇ **退出:** 退出配方，如果配方没有存盘，则提示存盘，如图 15.6 所示。

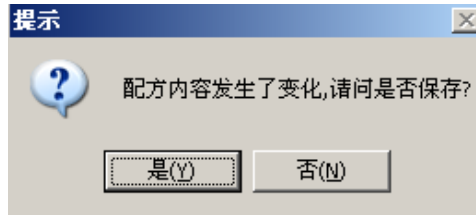


图 15.6 配方保存提示

15.2.1.2 工具菜单

单击“工具”菜单，弹出下拉式菜单，如图 15.7 所示。

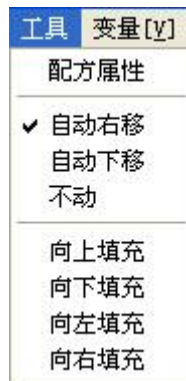



图 15.7

✧ **配方属性：**按下此键，则弹出“定义配方”对话框如图 15.8 所示。



图 15.8 定义配方属性

- 配方名称：要建立的配方名，由用户设定，定义的名称要有实际的意义。
- 配方种类数目：指定配方数目，最大值为 256 种。
- 变量数量：指定与配方中的项目相对应的变量个数，最大值为 999 个。

 注意：

配方种类数目和变量数量要与实际配方中种类数目、变量数量相同，否则运行过程中不能正确调用配方。

- ◇ **自动右移**：这时按下 enter 键，输入焦点自动右移。
- ◇ **自动下移**：这时按下 enter 键，输入焦点自动下移。
- ◇ **不动**：这时按下 enter 键，输入焦点不动。
- ◇ **向上填充**：输入焦点及向上所有被选中的方框都填入与输入焦点的值相同的值。
- ◇ **向下填充**：输入焦点向下所有被选中的方框都填入与输入焦点的值相同的值。
- ◇ **向左填充**：输入焦点向左所有被选中的方框都填入与输入焦点的值相同的值。

◇ **向右填充:** 输入焦点向右所有被选中的方框都填入与输入焦点的值相同的值。

15.2.1.3 变量菜单

单击“变量[V]”菜单，弹出“选择变量名”对话框，如图 15.9 所示，供用户选择数据词典中已定义的变量。

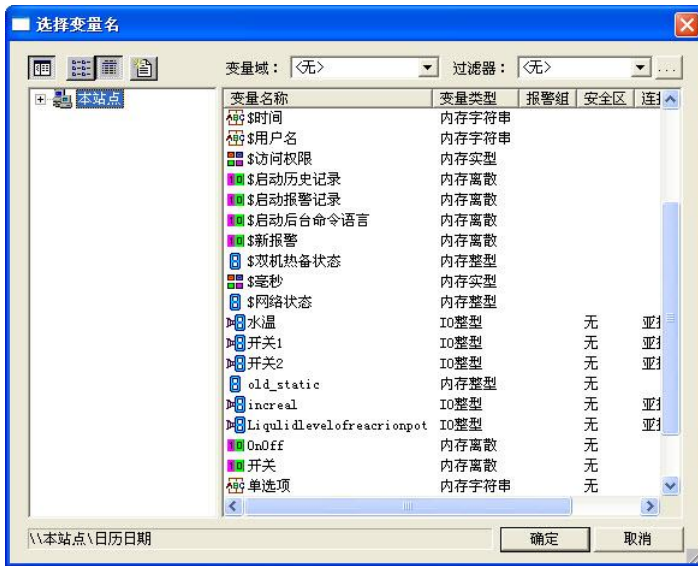


图 15.9 连接变量

15.2.2 创建配方模板

💡 例：下面举一个例子介绍如何创建配方模板。

1、加入变量

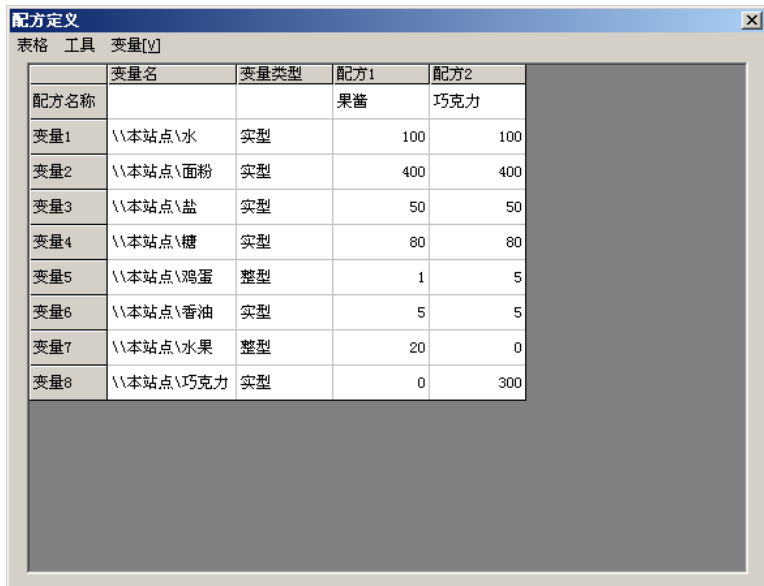
鼠标选中“变量 1”所在列名为“变量名”的单元格，此时“变量[V]”菜单栏变为黑色有效。单击“变量”，弹出“选择变量名”窗口，选中一个已经定义好的组态王 7.0 变量，单击“确定”，完成变量选择。“配方定义”窗口中相应变量的变量类型自动显示出来。如果变量名是由手动输入的，则需要手动输入相应的变量类型。加入多个变量的方法相同。

2、建立配方

在第一行中各个配方名称相应的单元格中输入各种配方的名称。鼠标单击“配方 1”下面的单元格，单元格变为输入状态，输入配方名称即可。接下来在下面对应变量中输入每种配方不同的变量的量值。

3、修改配方属性

编辑完配方之后，鼠标单击“工具”菜单中“配方属性”，定义配方模板的名称为“面包配方”，按照实际配方种类和使用的变量输入数据。定义好的配方模板如图 15.10 所示。



配方名称	变量名	变量类型	配方1	配方2
变量1	\\本站点\水	实型	100	100
变量2	\\本站点\面粉	实型	400	400
变量3	\\本站点\盐	实型	50	50
变量4	\\本站点\糖	实型	80	80
变量5	\\本站点\鸡蛋	整型	1	5
变量6	\\本站点\香油	实型	5	5
变量7	\\本站点\水果	整型	20	0
变量8	\\本站点\巧克力	实型	0	300

图 15.10 定义好的配方模板

“配方定义”窗口中的前两列为变量名、变量类型。

- ◇ **变量名：**为组态王 7.0 中已经定义的数据变量名，定义配方之前必须先和数据词典中定义所有配方中要用到的变量。
- ◇ **变量类型：**为整数型、实数型、离散型、字符串型中的一种，当用户选择变量名后，变量类型会自动加入，不需要用户输入。当用户手动输入变量名后，变量类型不自动加入，需要用户输入。

15.3 如何使用配方

配方的使用是建立配方模板后，通过使用配方命令语言函数实现的。配方命令语言函

数的调用可通过建立操作按钮或是在命令语言中调用来实现。下面首先详细介绍配方命令语言函数，然后再建立一个配方操作按钮的一个实际例子。

15.3.1 配方命令语言函数

◇ RecipeDelete

此函数用于删除指定配方模板文件中当前指定的配方，语法格式使用如下：

```
RecipeDelete( "filename", "recipeName" );
```

参数 filename：指配方模板文件存放的路径和相应的文件名；

recipeName：指配方模板文件中特定配方的名字。

说明：文件名和配方名如果加上双引号，则表示是字符串常量，若不加双引号，则可以是组态王 7.0 中的 DDE 或内存型字符串变量。

例子：RecipeDelete("C:\recipe\北京面包厂.csv", "配方 3")；此语句将配方模板文件“北京面包厂.csv”中的配方 3 删除。

◇ RecipeLoad

此函数将指定配方调入模板文件中的数据变量中，语法格式使用如下：

```
RecipeLoad( "filename", "recipeName" );
```

参数 filename：指配方模板文件存放的路径和相应的文件名；

recipeName：指配方模板文件中特定配方的名字。

说明：文件名和配方名如果加上双引号，则表示是字符串常量，若不加双引号，则可以是组态王 7.0 中的 I/O 型或内存型字符串变量。

例子: RecipeLoad (“C:\recipe\北京面包厂.csv”, “水果香型面包”);

此语句将配方模板文件 “北京面包厂.csv” 中的配方 “水果香型面包” 调入到项目模板定义中的数据变量中。

◇ RecipeSave

此函数用于存放一个新建配方或把对原配方的修改变化存入已有配方模板文件中, 语法格式使用如下:

```
RecipeSave( "filename", "recipeName" );
```

参数 filename: 指配方模板文件存放的路径和相应的文件名。

recipeName: 指配方模板文件中特定配方的名字,

说明: 文件名和配方名如果加上双引号, 则表示是字符串常量, 若不加双引号, 则可以是组态王 7.0 中的 I/O 型或内存型字符串变量。

配方模板文件必须存在, 如果配方模板文件不存在, 则要事先创建配方模板文件, 否则, 调用此函数将失败, 并返回 FALSE。

例子: RecipeSave (“C:\recipe\北京面包厂.csv”, “配方 3”);

此语句将配方的修改变化存入到配方模板文件 “北京面包厂.csv” 中的配方 3 中。如果 “北京面包厂.csv” 中没有配方 3, 则系统自动创建。

◇ RecipeSelectNextRecipe

此函数用于在配方模板文件中选择指定配方的下一个配方, 语法格式使用如下:

```
RecipeSelectNextRecipe( "filename", "recipeName" );
```

参数 filename: 指配方模板文件存放的路径和相应的文件名;

`recipeName`: 是一个字符串变量, 存放工程人员选择的配方名字。

说明: 文件名和配方名如果加上双引号, 则表示是字符串参数, 若不加双引号, 则可以是在组态王 7.0 中的 I/O 型变量或内存型变量。配方创建后是按序存放的。

例子: `RecipeSelectNextRecipe ("C:\recipe\北京面包厂.csv", "配方 3")` ;
此语句运行后读取模板文件中“配方 3”的下一个配方, 如果字符串变量 `recipeName` 的值为空或没有找到, 则返回文件中的第一个配方; 如果变量, `recipeName` 的值为文件中的最后一个配方, 则仍返回此配方。

✧ **RecipeSelectPreviousRecipe**

此函数用于在配方模板文件中选择当前配方的前一个配方, 语法格式使用如下:

```
RecipeSelectPreviousRecipe("filename", "recipeName");
```

参数 `filename`: 指配方模板文件存放的路径和相应的文件名;

`recipeName`: 是一个字符串变量, 存放工程人员选择的当前配方名字。

说明: 文件名和配方名如果加上双引号, 则表示是字符串参数, 若不加双引号, 则可以是在组态王 7.0 中的 I/O 型变量或内存型变量。配方创建后是按序存放的。

例子: `RecipeSelectPreviousRecipe ("C:\recipe\北京面包厂.csv", "配方 3")` ;
此语句运行后读取模板文件中“配方 3”的上一个配方, 如果变量 `recipeName` 的值为空或没有找到, 则返回文件中的最后一个配方; 如果变量 `recipeName` 的值为文件中的第一个配方, 则仍返回此配方。

✧ **RecipeSelectRecipe**

此函数用于在指定的配方模板文件中选取工程人员输入的配方, 运行此函数后, 弹出

对话框，如下图 15.11 所示，工程人员可以输入指定的配方，并把此配方名送入字符串变量中存放。

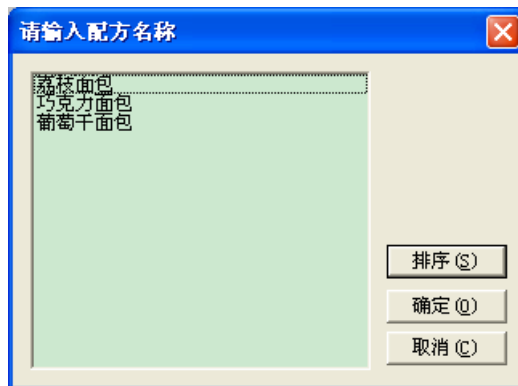


图 15.11 选择配方

语法格式使用如下：

```
RecipeSelectRecipe( "filename", "recipeNameTag", "Mess" );
```

参数 filename: 指配方模板文件存放的路径和相应的文件名；

recipeNameTag: 是一个字符串变量，存放工程人员选择的配方名字；

Mess: 字符串提示信息，由工程人员自己设定。

例子 RecipeSelectRecipe("C:\recipe\北京面包厂.csv", RecipeName, "请输入配方名!");

此语句运行后将弹出一个“选择配方”对话框，给出提示信息“请输入配方名！”，一旦工程人员从对话框中选择一个配方，则此函数将该配方的名字返回到变量 RecipeName 中存放。

◇ **RecipeInsertRecipe**

此函数用于在配方中选定的位置插入一个新的配方。执行该函数后，系统会弹出一个选择插入配方的对话框，如下图 15.12 所示。对话框中列出了当前配方中所有的配方名称，选择要插入的位置，确定后新的配方将被插入到指定配方的前面。

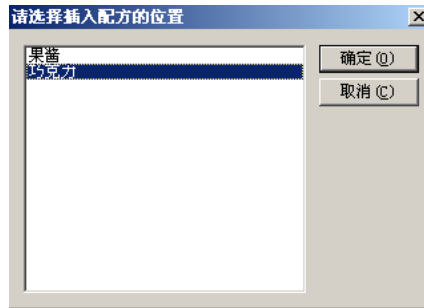


图 15.12 选择插入配方位置

`RecipeInsertRecipe(filename, InsertRecipeName);`

参数说明 filename: 字符型 指配方模板文件存放的路径和相应的文件名

 InsertRecipeName 字符型 要插入的新配方的名称

例子: 在上面的例子中新定义一种配方“牛奶”面包，选择插入到“巧克力”面包之前，在弹出的插入对话框中选择“巧克力”，确定后，新的配方插入到了指定的位置。

`RecipeInsertRecipe("C:\recipe\北京面包厂.csv", "牛奶");`再次选择时，显示配方顺序如下图 15.13 所示。

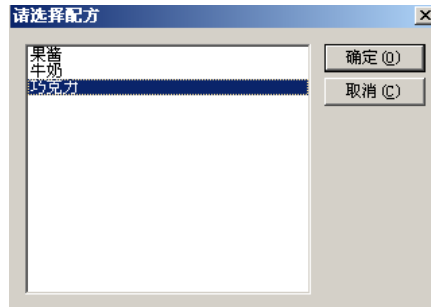


图 15.13 插入后的配方

◇ RecipeManage

此函数用来调用配方管理界面，可以实现对配方各种操作和各种数值的监视，显示管理界面如图 15.14 所示。

- 变量名：配方中对应操作的变量的名称。
- 实时值：工程运行时，配方对应变量的实时数据值。
- 当前配方值新值：装载的配方值或编辑的配方值。
- 当前配方原值：装载的配方值。
- 待装载配方值：从配方文件中选择的配方值或编辑后的配方值。

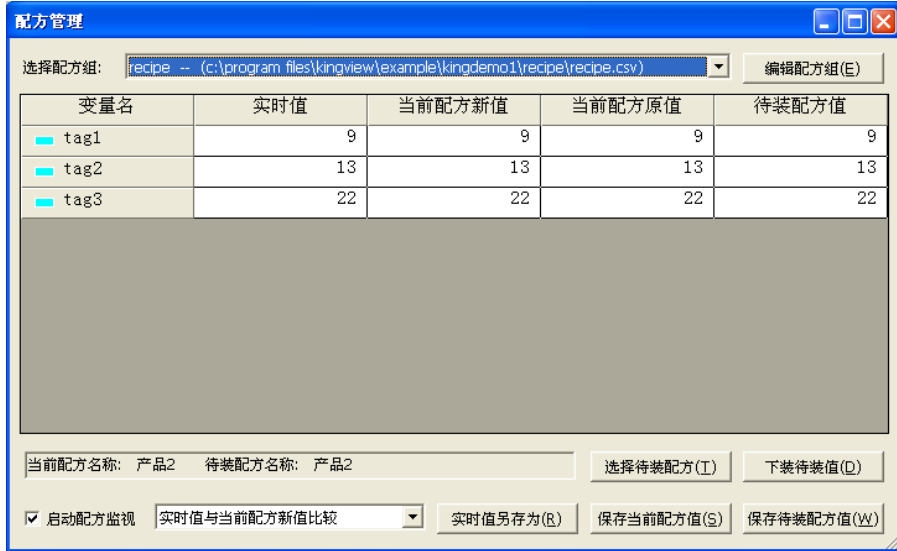


图 15.14 配方管理界面

界面操作说明：

1、选择配方组

从配方组列表中选择一个配方组进行操作，配方组列表显示当前工程中的所有配方文件。

2、编辑配方组：

编辑当前选择的配方文件，弹出配方编辑界面，可以新建、删除、复制、粘贴、打印和保存配方，通过表格转置切换配方的显示视图。



The screenshot shows a software window titled "当前配方组: recipe". The window has a menu bar with the following options: "新建配方 (N)", "删除配方 (D)", "复制配方 (C)", "粘贴配方 (V)", "配方重命名 (R)", "打印 (P)", "表格转置 (T)", and "保存 (S)". Below the menu bar is a table with three columns: "tag1", "tag2", and "tag3". The rows are labeled with product names: "产品1", "产品2", "产品3", "产品4", and "新配方0". The values in the table are as follows:

	tag1	tag2	tag3
产品1	10	12	21
产品2	9	13	22
产品3	8	14	23
产品4	7	15	23
新配方0	5	13	22

图 15.15 编辑配方组



图 15.16 选择配方

3、选择待装配方：

从当前配方文件中，选择一个配方准备下装到变量参数中。如下图，鼠标选择一个配方，点击确定，将选择的配方数据填充到图 15.14 中待装配方值中。

4、下装待装配方：

将待装配方值下载到变量参数中，并更新图 15.14 中当前配方原值显示列。

5、实时值另存为：

将变量的实时值存储到当前配方文件中，可以是新配方或是覆盖原有配方。

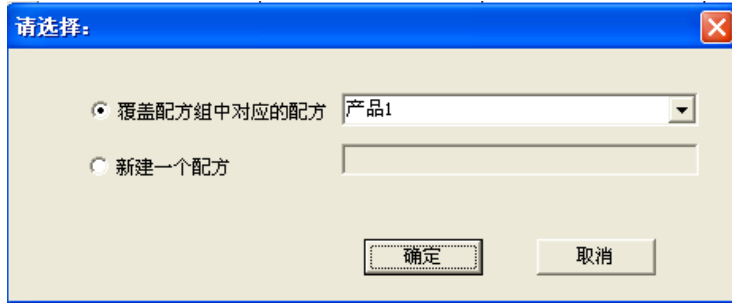


图 15.17 选择配方



6、保存当前配方值：

将变量参数当前使用的配方值保存到当前配方文件中，可存储为新配方，也可覆盖同名的配方。如图 15.17 所示。

7、保存待装配配方值：

将修改后的待装配配方值重新保存到当前配方文件中，可以存储为新配方，也可覆盖原有配方。如图 15.17 所示。

8、启动配方监视：

勾选“启动配方监视”，根据监视选项比较变量实时值和配方值，相同和不同时分别以不同图标标示， 值相同， 值不同。

15.3.2 配方示例

本节建立一个配方操作按钮的实际例子，以 15.2 节中定义的“面包配方.csv”模板文件为实例中的配方模板。

新建“配方管理”画面。在画面上建立配料变量显示，绘制多个按钮，各个按钮中连接配方管理命令语言函数。如图 15.14 所示。



图 15.14 开发系统配方管理画面

1、建立“选择配方”按钮

- ◇ 在画面制作系统绘出按钮，按钮文本字符串为“选择配方”；
- ◇ 给“选择配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeSelectRecipe("D:\配方管理\面包配方.csv", recipeName, "请输入配方名称");
```

```
RecipeLoad("D:\配方管理\面包配方.csv", recipeName);
```

注：其中 recipeName 是在数据词典中定义的内存字符串型的组态王 7.0 变量。

2、建立“调入配方”按钮

- ✧ 在画面制作系统绘出按钮，按钮文本字符串为“调入配方”；
- ✧ 给“调入配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeLoad( "D:\配方管理\面包配方.csv", recipeName );
```

3、建立“保存配方”按钮

- ✧ 在画面制作系统绘出按钮，按钮文本字符串为“保存配方”；
- ✧ 给“存配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeSave( "D:\配方管理\面包配方.csv", recipeName );
```

4、建立“选择下一个配方”按钮

- ✧ 在画面制作系统绘出按钮，按钮文本字符串为“选择下一个”；
- ✧ 给“选择下一个配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeSelectNextRecipe( "D:\配方管理\面包配方.csv", recipeName );
```

```
RecipeLoad( "D:\配方管理\面包配方.csv", recipeName );
```

5、建立“选择上一个配方”按钮

- ✧ 在画面制作系统绘出按钮，按钮文本字符串为“选择上一个”；
- ✧ 给“选择上一个配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeSelectPreviousRecipe( "D:\配方管理\面包配方.csv", recipeName );
```

```
RecipeLoad( "D:\配方管理\面包配方.csv", recipeName );
```

6、建立“删除配方”按钮

- ✧ 在画面制作系统绘出按钮，按钮文本字符串为“删除配方”；

- ◇ 给“删除配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeDelete( "D:\面包配方.csv", recipeName );
```

7、建立“插入配方”按钮

- ◇ 在画面制作系统绘出按钮，按钮文本字符串为“插入配方”；
- ◇ 给“插入配方”按钮进行命令语言连接，命令语言程序如下。

```
RecipeInsertRecipe ( "D:\面包配方.csv", recipeName );
```

配方管理画面就制作好了，保存画面，切换到运行系统中。执行配方操作按钮，对配方进行各种操作。运行系统显示如图 15.15 所示。



图 15.15 运行系统配方管理画面

第十六章 系统安全管理

- ☞ 介绍组态王开发系统安全管理
- ☞ 介绍组态王运行系统安全管理
- ☞ 介绍关于访问权限及口令设置的相关知识
- ☞ 教您使用访问权限和口令设置来增强系统安全性

安全保护是应用系统不可忽视的问题,对于可能有不同类型的用户共同使用的大型复杂应用,必须解决好授权与安全性的问题,系统必须能够依据用户的使用权限允许或禁止其对系统进行操作。组态王提供一个强有力的先进的基于用户的安全管理系统。在“组态王”系统中,在开发系统里可以对工程进行加密。打开工程时只有输入密码正确时才能进入该工程的开发系统。对画面上的图形对象设置访问权限,同时给操作者分配访问优先级和安全区,运行时当操作者的优先级小于对象的访问优先级或不在对象的访问安全区内时,该对象为不可访问,即要访问一个有权限设置的对象,要求先具有访问优先级,而且操作者的操作安全区须在对象的安全区内时,方能访问。组态王以此来保障系统的安全运行。

16.1 组态王开发系统安全管理

16.1.1 如何对工程进行加密

为了防止其他人员对工程进行修改,在组态王开发系统中可以分别对多个工程进行加密。当进入一个有密码的工程时,必须正确输入密码方可进入开发系统,否则不能打开该工程进行修改,从而实现了组态王开发系统的安全管理。

新建组态王工程,首次进入组态王浏览器,系统默认没有密码,可直接进入组态王开发系统。如果要对该工程的开发系统进行加密,执行工程浏览器中“工具\工程加密”命令。弹出“工程加密处理”对话框,如图 16.1 所示。

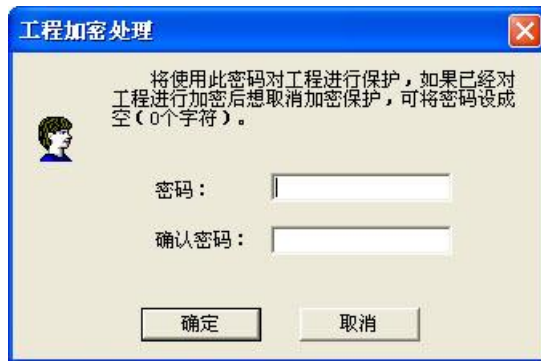


图 16.1 “工程加密处理”对话框

- ◇ **密码**文本框: 输入密码, 密码长度不超过 12 个字节, 密码可以是字母(区分字母大小写)、数字、其它符号等。
- ◇ **确认密码**文本框: 再次输入相同密码进行确认。
- ◇ **取消**按钮: 单击取消按钮将取消对工程实施加密操作;
- ◇ **确定**按钮: 单击确定按钮后, 系统将对工程进行加密。

加密过程中系统会弹出提示信息框,显示对每一个画面分别进行加密处理。当加密操作完成后,系统弹出“操作完成”,如图 16.2 所示。



图 16.2 加密操作成功

退出组态王工程浏览器，每次在开发环境下打开该工程都会出现检查文件密码对话框，要求输入工程密码，如图 16.3 所示。



图 16.3 检查文件密码

密码输入正确后，将打开该工程。否则出现如图 16.4 所示对话框：



图 16.4 密码错误对话框

单击重试按钮将回到检查文件密码对话框，用户可重新输入密码。单击取消按钮，工程将无法打开。

16.1.2 如何去除工程加密

如果想取消对工程的加密，在打开该工程后，单击“工具\工程加密”，弹出“工程加密处理”对话框，将密码设为空，单击确定按钮，则弹出如图 16.5 所示对话框：



图16.5 取消工程加密

单击确定按钮系统将取消对工程的加密。单击取消按钮放弃对工程加密的取消操作。



注意：

如果用户丢失工程密码，无法打开组态王工程进行修改，请小心妥善保管密码！

16.2 组态王运行系统安全管理

16.2.1 运行系统安全管理概述

在“组态王”系统中，为了保证运行系统的安全运行，对画面上的图形对象设置访问权限，同时给操作者分配访问优先级和安全区，当操作者的优先级小于对象的访问优先级或不在对象的访问安全区内时，该对象为不可访问，即要访问一个有权限设置的对象，要求先具有访问优先级，而且操作者的操作安全区须在对象的安全区内时，方能访问。操作者的操作优先级级别从 1~999，每个操作者和对象的操作优先级级别只有一个。系统安全区共有 64 个，用户在进行配置时，每个用户可选择除“无”以外的多个安全区，即一

个用户可有多个安全区权限，每个对象也可有多个安全区权限。除“无”以外的安全区名称可由用户按照自己的需要进行修改。在软件运行过程中，优先级大于 900 的用户还可以配置其他操作者，为他们设置用户名、口令、访问优先级和安全区。

16.2.2 安全管理配置

16.2.2.1 优先级和安全区

组态王采用分优先级和分安全区的双重保护策略。组态王系统将优先级从小到大定为 1 到 999，可以对用户、图形对象、热键命令语言和控件设置不同的优先级。安全区功能在工程中使用广泛，在控制系统中一般包含多个控制过程，同时也有多个用户操作该控制系统。为了方便、安全地管理控制系统中的不同控制过程，组态王引入了安全区的概念。将需要授权的控制过程的对象设置安全区，同时给操作这些对象的用户分别设置安全区，例如工程要求 A 工人只能操作车间 A 的对象和数据，B 工人只能操作车间 B 的对象和数据，组态王中的处理是：将车间 A 的所有对象和数据的安全区设置为包含在 A 工人的操作安全区内，将车间 B 的所有对象和数据的安全区设置为包含在 B 工人的操作安全区内，其中 A 工人和 B 工人的安全区不相同。

应用系统中的每一个可操作元素都可以被指定保护级别（最大 999 级，最小 1 级）和安全区（最多 64 个），还可以指定图形对象、变量和热键命令语言的安全区。对应地，设计者可以指定操作者的操作优先级和工作安全区。在系统运行时，若操作者优先级小于可操作元素的访问优先级，或者工作安全区不在可操作元素的安全区内时，可操作元素是不可访问或操作的。

组态王中可定义操作优先级和安全区的有：

- ◇ **三种用户输入连接：**模拟值输入、离散值输入、字符串输入；
- ◇ **两种滑动杆输入连接：**水平滑动杆输入、垂直滑动杆输入；
- ◇ **五种命令语言输入连接和热键命令语言：**（鼠标或等价键）按下时、按住时、弹起时、鼠标进入、鼠标离开时；
- ◇ **其它：**报警窗、图库精灵、控件（包括通用控件）、自定义菜单；
- ◇ **变量的定义：**每个变量有相应的安全区和优先级。

当用户登录成功后，对于动画连接命令语言和热键命令语言，只有当登录用户的操作优先级不小于该图素或热键规定的操作优先级，并且安全区在该图素或热键规定的安全区内时，方可访问该对象或执行命令语言。命令语言执行时与其中连接的变量的安全区没有关系，命令语言会正常执行。对于滑动杆输入和值输入除要求登录用户的操作优先级不小于对象设置的操作优先级、安全区在对象的安全区内外，其安全区还必须在所连接变量的安全区内，否则用户虽然可以访问对象（使对象获得焦点），但不能操作和修改它的值，在组态王的信息窗口中也会有对连接变量没有修改权限的提示信息。

16.2.2.2 如何配置用户

组态王中可根据工程管理的需要将用户分成若干个组来管理，即用户组。

在组态王工程浏览器目录显示区中，用鼠标双击大纲项系统配置下的用户配置，或从工程浏览器的顶部工具栏中单击“用户”，弹出“用户和安全区管理器”对话框，如图 16.6 所示。

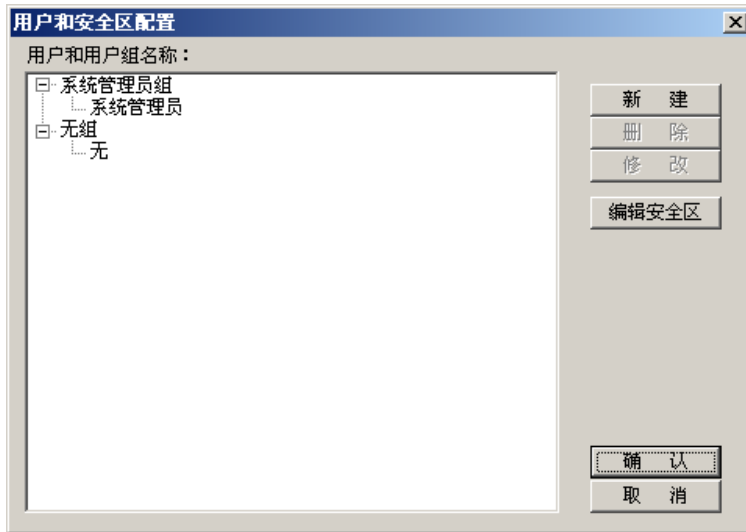


图 16.6 用户和安全区配置对话框

16.2.2.2.1 如何定义用户组

单击“新建”按钮，弹出“定义用户组 and 用户”对话框，选中“用户组”按钮，如图 16.6 所示。

用户组下面可以包含多个用户，在对话框中的“用户组名”中填入所要配置的当前用户组的名称，如“系统维护员”；在“用户组注释”中填入对当前用户组的注释，如“系统维护组成员”。在右侧的“安全区”列表框中选择当前用户组下所有用户的公共安全区，配置完成后，按“确定”返回。

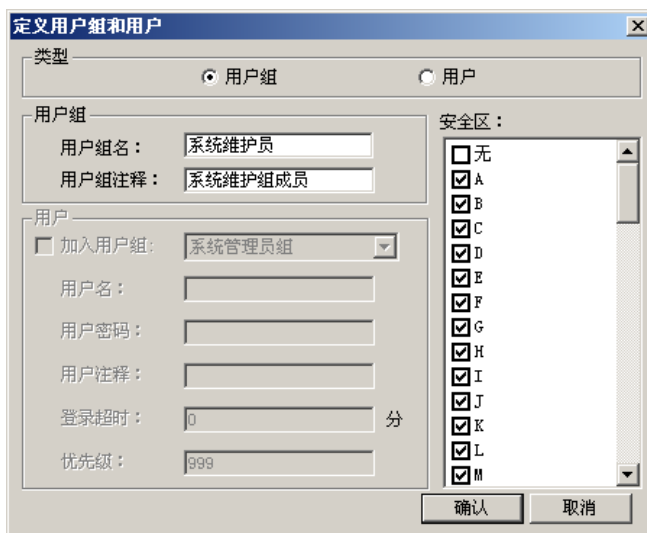


图 16.7 用户组配置对话框

也可对已定义完的用户组进行修改。在“用户和安全区配置”中选择要修改的用户组，单击“修改”按钮，弹出“定义用户组和用户”对话框，可以对用户组名、用户组注释、安全区等进行修改。单击“删除”按钮，可以对选中的用户组进行删除操作，系统会提示用户是否确实要进行删除操作，如果是点击“确定”按钮，否则点击“取消”按钮，取消删除操作。如果该用户组中定义有用户，则“删除”按钮为灰色，该命令无效，不能进行删除操作，只有当用户组为空时才可以删除该用户组。对系统默认生成的“系统管理员组”和“无组”不能进行删除操作，只能对其进行修改操作。

 注意:

用户配置中，用户组名、用户组注释、用户名、用户密码、用户注释最多可输入 31 个字符。

16.2.2.2.2 如何定义用户组下的用户

一个用户组中可以包含多个用户，当建立了一个用户组之后，就可以在该用户组下添加用户了。在“定义用户组 and 用户”界面上，单击“用户”按钮，则“用户”下面的所有选项变为有效。如图 16.8 所示。

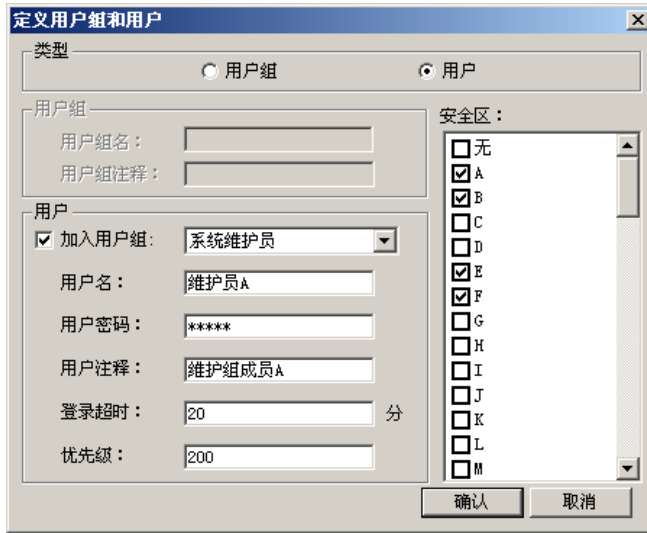


图 16.8 用户组中用户配置对话框

选中“加入用户组”，从下拉列表框中选择用户组名。例如刚才定义的“系统维护员”。在“用户名”中输入当前独立用户的名称，如“维护员 A”；在“用户密码”中输入当前用户的密码，密码输入后显示为“*”；在“用户注释”中输入对当前用户的说明；“登录超时”中输入登录超时时间，即用户登录后，使用权限的时间，当到达规定的时间时，系统权限自动变为“无”，如果登录超时的值为 0，则登录后没有登录超时的限制；在“优先级”中输入当前用户的操作优先级级别；在“安全区”中选择该用户所属安全区。用户

配置完成后单击“确认”按钮。

也可对已定义完的组中的用户进行修改。在“用户和安全区配置”中选择要修改的用户组中的用户，单击“修改”按钮，弹出“定义用户组 and 用户”对话框，可以对用户名、用户密码、用户注释、登录超时、优先级、安全区等进行修改。不能将该用户修改属于其他的用户组。单击“删除”按钮，可以对选中的用户进行删除操作。系统会提示用户是否确实要进行删除操作，如果是点击“确定”按钮，否则点击“取消”按钮，取消删除操作。对系统默认生成的“系统管理员组”和“无组”中的用户“系统管理员”和“无”不能进行删除操作，只能对其进行修改操作。



注意：

增加到组中的用户将继承其组的关于安全区和优先级方面的设置，但用户可以对每个用户的安全区和优先级进行修改。

16.2.2.2.3 如何定义独立用户

对于单独的不需要加入到任何一个用户组的用户，可以定义为独立用户。

在“用户和安全区配置”对话框中，单击“新建”按钮，弹出独立用户配置的“定义用户组 and 用户”对话框，如图 16.9 所示。

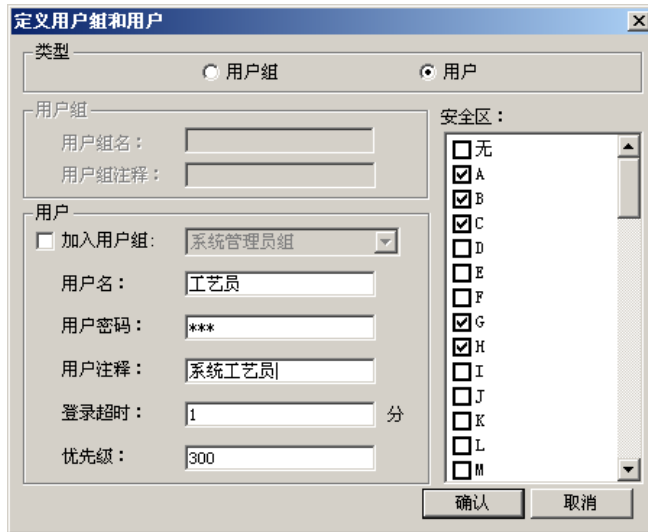


图 16.9 独立用户配置对话框

独立用户不属于任何一个用户组，其本身就是一个用户。在“用户名”中输入当前独立用户的名称，如“工艺员”；在“用户密码”中输入当前用户的密码；在“用户注释”中输入对当前用户的说明；“登录超时”中输入登录超时时间；在“安全区”中选择该用户所属安全区。用户配置完成后单击“确认”按钮。

也可对已定义完的用户进行修改。在“用户和安全区配置”中选择要修改的用户，单击“修改”按钮，弹出“定义用户组 and 用户”对话框，可以对用户名、用户密码、用户注释、登录超时、优先级、安全区等进行修改。

16.2.2.2.4 如何修改安全区

安全区的默认名称为“A, B, C...”，用户可通过“用户和安全区管理器”中的“编辑安全区”按钮来修改各个安全区的名称。单击“编辑安全区”按钮，弹出“安全区配置”

对话框，如图 16.10 所示。

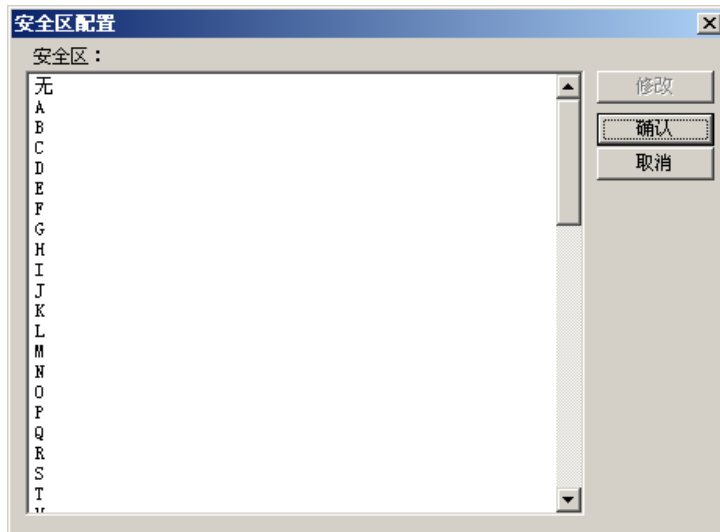


图 16.10 安全区配置对话框

用鼠标单击选择一个除“无”外的要修改的安全区名称，“修改”按钮由灰色不可用变为黑色可用，单击“修改”按钮，弹出“更改安全区名”对话框，如图 16.11 所示。

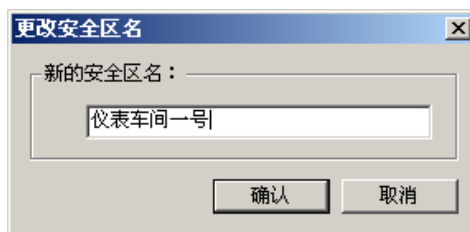


图 16.11 更改安全区名称对话框


在文本框中输入安全区的名称，单击“确认”按钮完成修改，照此方法，可修改所有的安全区名称。在其它的安全区选择列表框中选择安全区时，安全区的名称变成了修改后

的名称，方便操作。

16.2.2.3 如何设置对象的安全属性

16.2.2.3.1 设置图形对象的安全属性

当用户登录成功后，对于图形的动画连接命令语言，只有当登录用户的操作优先级不小于该图形规定的操作优先级，并且安全区在该图形规定的安全区内时，方可访问该图形或执行命令语言。命令语言执行时与其中连接的变量的安全区没有关系，命令语言会正常执行。对于滑动杆输入和值输入除要求登录用户的操作优先级不小于对象设置的操作优先级、安全区在对象的安全区内外，其安全区还必须在所连接变量的安全区内，否则用户虽然可以访问对象（使对象获得焦点），但不能操作和修改它的值，在组态王的信息窗口中也会有对连接变量没有修改权限的提示信息。

在组态王开发系统中双击画面上的某个对象，如矩形，弹出“动画连接”对话框，如图 16.12 所示。选择具有数据安全动画连接中的一项，如命令语言连接。则“优先级”和“安全区”选项变为有效，在“优先级”中输入访问的优先级级别；单击“安全区”后的  按钮选择安全区，如图 16.12 所示。弹出“选择安全区”对话框，如图 16.13 所示。

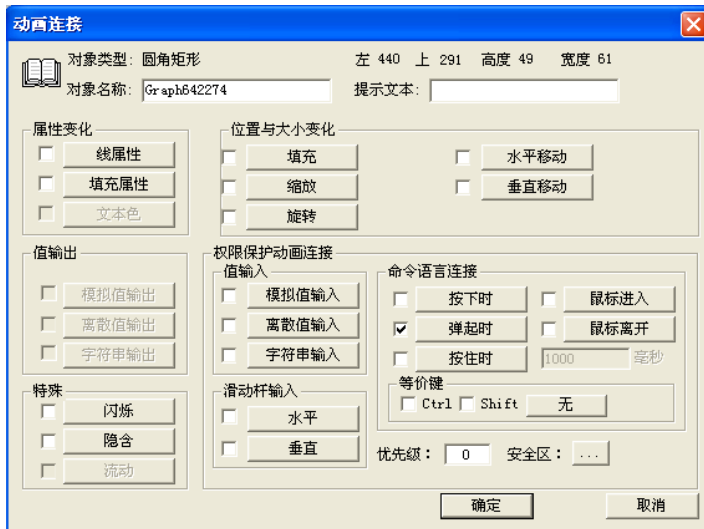


图 16.12 动画连接访问权限设置

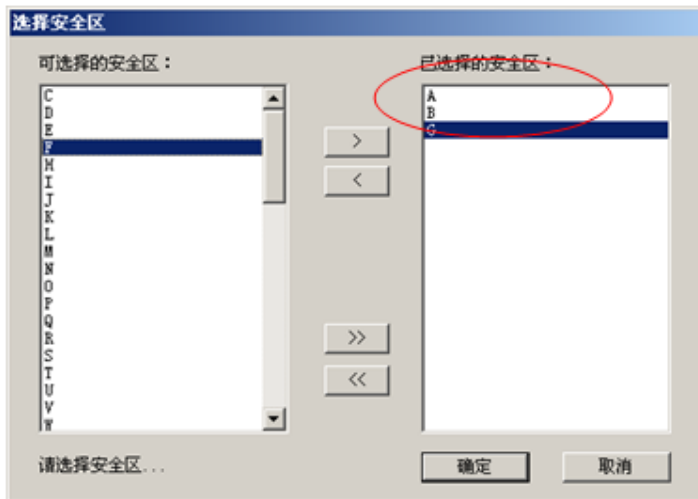


图 16.13 安全区选择对话框

设置安全区的方法为：单击左侧“可选择的安全区”列表框中的安全区名称，然后单击“>”按钮，即可将该安全区名称加入右侧的“已选择的安全区”列表框中，若一次选择连续排列的多个安全区，可以利用 shift 键或按下并同时拖动鼠标，来选择所有需要选中的多个安全区，若选择非连续排列的多个安全区，可以利用 Ctrl 键或者单个多次加入。若需加入左侧“可选择的安全区”列表框中的全部安全区，请使用“》”按钮。取消安全区的方法为：选中“已选择的安全区”列表框中的安全区名称，单击“<”按钮即可，选中多个的方法与上同。若需取消右侧“可选择的安全区”列表框中的全部安全区，请使用“《”按钮。选择完毕后，单击“确定”返回。

16.2.2.3.2 设置热键命令语言的安全属性

当用户登录成功后，对于热键命令语言，只有当登录用户的操作优先级不小于热键规定的操作优先级，并且安全区在热键规定的安全区内时，方可执行热键命令语言。命令语言执行时与其中连接的变量的安全区没有关系，命令语言会正常执行。

在工程浏览器的目录显示区，选择“文件\命令语言\热键命令语言”，在右边的内容显示区出现“新建”图标，用左键双击此图标，则弹出“命令语言”对话框，如图 16.14 所示。设置热键并在操作优先级输入栏内输入优先级级别，在安全区选择列表中选择热键的安全区。只有优先级高于该级别和安全区在该热键安全区内的用户登录后按下热键时，才会执行这段命令语言。热键的优先级级别和安全区设置与图形对象优先级级别和安全区设置相同，请参见“16.2.2.3.1 设置图形对象的安全属性”一节。

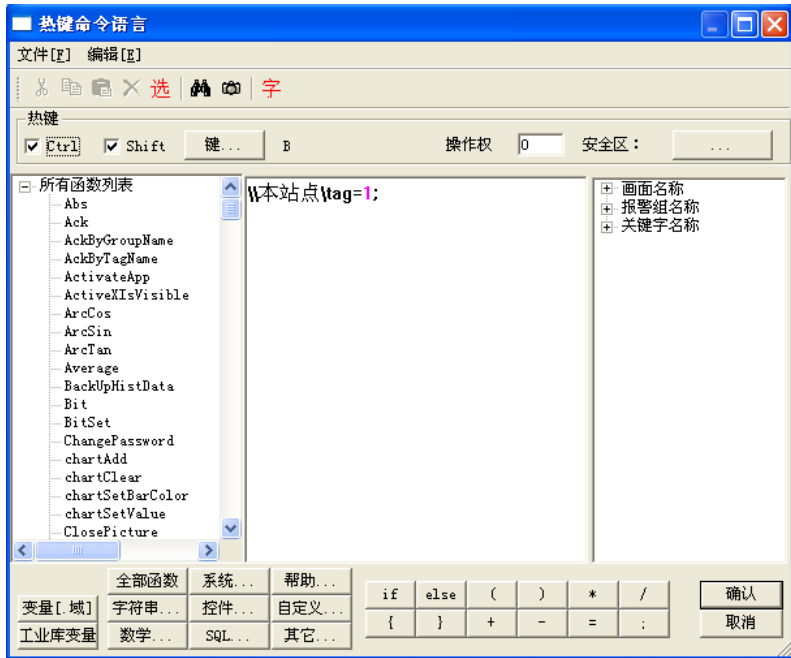


图 16.14 热键优先级和安全区设置

16.2.2.3.3 如何设置变量的安全属性

在工程浏览器“数据词典”中新建变量时，弹出“变量属性”对话框，定义好变量后，单击“记录和安全区”属性页标签，进入记录和安全区配置对话框，如图 16.15 所示。

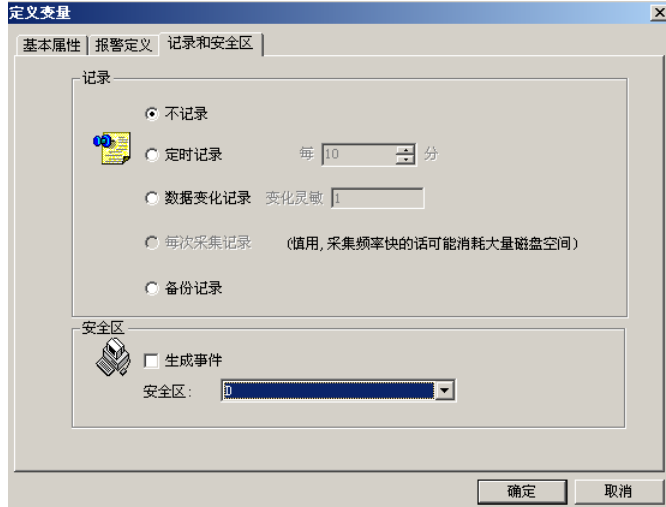


图 16.15 变量属性——安全区设置对话框

工程人员根据工程设计需要在安全区列表框中选择一个安全区名称。选择完后单击“确定”按钮完成。

16.2.2.3.4 如何设置控件的安全属性

对于组态王的控件，只有趋势曲线类控件中的温控曲线和 X-Y 轴曲线、窗口控制类控件（包括列表框、组合框、复选框、编辑框、单选按钮）和超级文本显示控件可以设置访问权限，这些控件没有安全区设置。只与相连接的变量的安全区有关。

对于 Active X 控件，既可以设置优先级，也可以设置安全区。在组态王开发系统中双击画面上某个控件（为第十四章列举的 控件），弹出“动画连接属性”对话框，如图 16.16 所示。“优先级”输入栏输入该控件的访问优先级级别（1~999），单击“安全区选择”按钮弹出安全区选择对话框，选择需要的安全区后，在控件的“动画连接属性”对话框

框中的“安全区”文本框中显示出已经选择的安全区名称。

 注意：

组态王的可以设置操作权限的控件只能设置优先级。

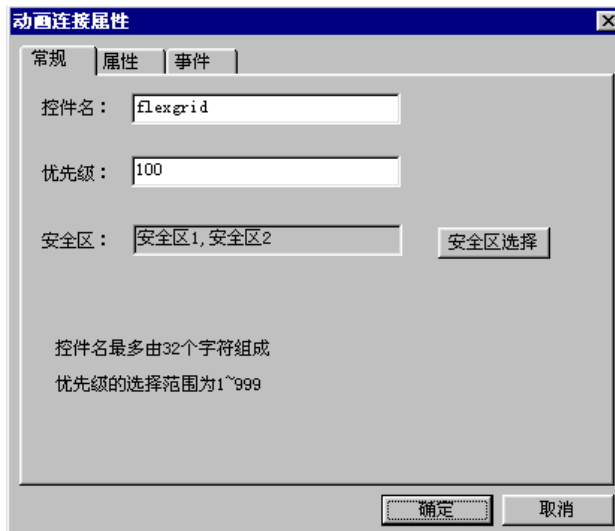


图 16.16 设置控件的访问优先级和安全区

16.2.3 运行时如何登录用户

16.2.3.1 登录用户

在 Touchview 运行环境下，操作人员必须以自己的身份登录才能获得一定的操作权。在运行系统中打开菜单“特殊\登录开”菜单项，则弹出如图 16.17 所示。

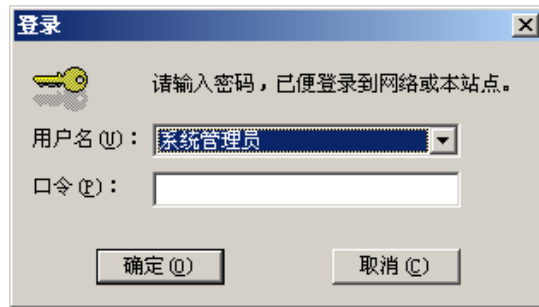


图 16.17 软件运行时登录用户

单击用户名下拉列表框显示在开发系统中定义的所有用户的用户名称，从中选择一个；在“口令”文本框中正确输入口令，然后单击“确定”按钮。如果登录无误，使用者将获得一定的操作权。否则系统显示“登录失败”的信息。

“登录开”的操作还可以通过命令语言来实现。假设给按钮“用户登录”设置命令语言连接：`LogOn()`；程序运行后，当操作者单击按钮时，将弹出“登录”对话框。如果在组态王工程浏览器中选择了菜单命令“配置\运行系统”，而且在弹出的“运行系统设置”对话框中的“特殊”卡片上使“运行时使用模拟键盘”有效，则“登录”对话框弹出后，点击密码对话框将同时显示模拟键盘。用户用鼠标在键盘窗口内选择字母或数字，如同使用真正的键盘一样。如图 16.18 所示。



图 16.18 模拟键盘

为了加强运行系统的安全性，组态王运行系统还提供用户操作双重验证功能。在运行过程中，当用户希望进行一项操作时（如分闸或合闸），为防止误操作，需要进行双重认证。即在身份认证对话框中，既要输入操作者的名称和密码，又要输入监控者的姓名和密码，两者验证无误时方可操作。实现双重验证通过调用 `PowerCheckUser()` 函数实现，函数具体使用方法如下：

```
bool PowerCheckUser(string OperatorName, string MonitorName);
```

其中：OperatorName：返回的操作者姓名；MonitorName：返回监控者姓名。

返回值：1：验证成功；0：验证失败。

运行时执行该函数后，弹出身份验证对话框，如图 16.19 所示。

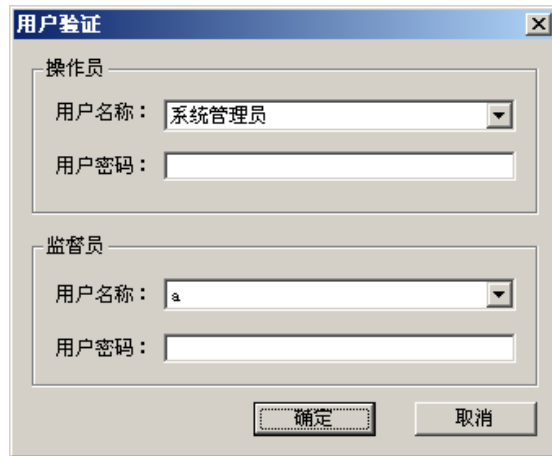


图16.19 用户验证对话框

在“操作员”用户栏中将默认显示当前登录的用户；在“监督员”栏中将默认显示上次登录的用户。可通过下拉框选择已经在组态王中定义的用户。对于操作员和监督员，不能以相同的用户名称进行登录。当点击“确定”按钮时，如果用户的名称，以及用户的密码完全正确，将完成此次的用户验证，但是用户的验证将不影响工程用户登录的情况。当用户取消此次登录，将返回登录失败的信息，不进行任何的操作。操作者和监控者具有不同的权限和类型。在组态王中建议采用变通方法，两者均为组态王用户即可。

组态王运行系统中还提供对加密锁的加密方式。通过调用 `GetKey();` 函数来规定某个工程只能使用某一个加密锁，从而起到加密作用。运行系统中执行 `GetKey();` 函数，可以得到当前插在计算机上加密锁的序列号。函数使用格式如下：

`GetKey();`

此函数没有参数。

返回值为字符串型：加密锁的序列号。

16.2.3.2 退出登录

用户完成操作离开微机时，有必要退出登录，以免非法用户侵入系统。退出登录只须选择菜单“特殊\登录关”即可。

同样使用函数LogOff()；的功能与菜单命令“特殊\登录关”相同。假设给按钮“用户登录关”设置命令语言连接：LogOff()；程序运行后，当操作者单击按钮时，将退出登录的用户。

16.2.4 运行时如何重新设置口令和权限

在运行环境下，“组态王”还允许任何登录成功的用户（访问权限无限制）修改自己的口令。首先进行用户登录，具体操作方法请参见“16.2.3.1 登录用户”一节。然后执行“特殊\修改口令”菜单，则弹出如图 16.20 所示。

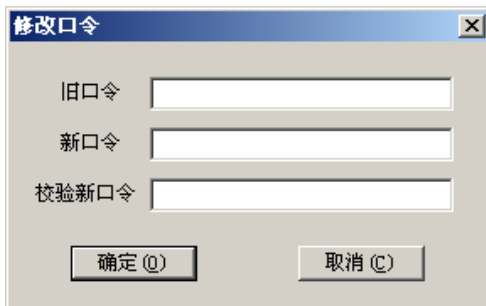


图 16. 20 修改口令

在“旧口令”输入栏中输入旧的口令，在“新口令”输入栏中输入新的口令，在“校验新口令”输入栏中同样输入新的口令，给用户一次核实的机会。最后单击“确定”按钮，然后旧的口令将被新的口令所代替。

修改口令也可以通过命令语言进行。函数 `ChangePassWord()` 的功能和菜单命令“特殊\修改口令”相同。假设给按钮“修改口令”设置命令语言连接：`ChangePassWord()`；程序运行后，当操作者单击按钮时，将弹出“修改口令”对话框。

运行系统中，对于操作权限大于 900 的用户还可以对用户权限进行修改，可以添加、删除或修改各个用户的优先级和安全区。如果登录用户权限小于 900，执行“特殊\配置用户”命令时，系统弹出如图 16.21 所示窗口。



图 16.21 不能配置用户提示

如果登录用户权限大于或等于 900，执行“特殊\配置用户”命令时，系统弹出“用户和安全区配置”对话框。可以修改用户的优先级和安全区。具体使用的方法和开发系统中配置用户的方法是一样的，请参见“16.2.2.2 如何配置用户”一节。在运行系统中配置完成用户后，系统将会自动记住，打开组态王开发系统用户配置，显示的是新配置完成的用户。

同样使用函数EditUsers();的功能与菜单命令“特殊\配置用户”相同。假设给按钮“配置用户”设置命令语言连接: EditUsers();程序运行后,当操作者单击按钮时,用户权限大于或等于900时,系统弹出“用户和安全区配置”对话框。

16.2.5 与安全管理相关的系统变量和函数

16.2.5.1 系统变量

与安全管理有关的系统变量有两个:“\$用户名”和“\$访问权限”。

◇ “\$用户名”

是内存字符串型变量,在程序运行时记录当前用户的名字。若没有用户登录或用户已退出登录,“\$用户名”为“无”。

◇ “\$访问权限”

是内存实型变量,在程序运行时记录着当前用户的访问权限。若没有用户登录或用户已退出登录,“\$访问权限”为1,安全区为“无”。

16.2.5.2 函数

与安全管理有关的函数有:

◇ ChangePassWord()

此函数用于显示“修改口令”对话框,允许登录用户修改他们的口令。

调用格式: ChangePassWord();

此函数无参数。

◇ **EditUsers()**

此函数用于显示“用户和安全区配置”对话框，允许权限大于 900 的用户配置用户和安全区。

调用格式: EditUsers();

此函数无参数。

◇ **GetKey()**

此函数用于系统运行时获取组态王加密锁的序列号。

调用格式: GetKey();

此函数无参数。

返回值为字符串型: 加密锁的序列号。

◇ **LogOn()**

此函数用于在 Touchview 运行系统中登录。

调用格式: LogOn();

此函数无参数。

◇ **LogOff()**

此函数用于在 Touchview 运行系统中退出登录。

调用格式: LogOff();

此函数无参数;

◇ **PowerCheckUser()**

此函数用于运行系统中进行身份双重认证。

调用格式: `Result = PowerCheckUser (OperatorName, MonitorName);`

参数: **OperatorName**: 返回的操作者姓名; **MonitorName**: 返回监控者姓名。

返回值: **Result=1**: 验证成功; **Result=0**: 验证失败。

第十七章 报表系统

- ☞ 介绍报表系统的创建和格式设置
- ☞ 介绍报表函数
- ☞ 教您如何制作报表
- ☞ 介绍报表模板
- ☞ 教您如何使用报表向导

数据报表是反应生产过程中的数据、状态等，并对数据进行记录的一种重要形式。是生产过程必不可少的一部分。它既能反映系统实时的生产情况，也能对长期的生产过程进行统计、分析，使管理人员能够实时掌握和分析生产情况。

组态王提供内嵌式报表系统，工程人员可以任意设置报表格式，对报表进行组态。组态王为工程人员提供了丰富的报表函数，实现各种运算、数据转换、统计分析、报表打印等。既可以制作实时报表，也可以制作历史报表。组态王还支持运行状态下单元格的输入操作，在运行状态下通过鼠标拖动改变行高、列宽。另外，工程人员还可以制作各种报表模板，实现多次使用，以免重复工作。

组态王报表提供向导工具，该工具可以以组态王的历史库或 KingHistorian 为数据源，快速建立所需的班报表、日报表、周报表、月报表、季报表和年报表。此外，还可以实现值的行列统计功能。

17.1 创建报表

17.1.1 如何创建报表窗口

进入组态王开发系统，创建一个新的画面，在组态王工具箱按钮中，用鼠标左键单击“报表窗口”按钮，如图 17.1 所示，此时，鼠标箭头变为小“+”字形，在画面上需要加入报表的位置按下鼠标左键，并拖动，画出一个矩形，松开鼠标键，报表窗口创建成功，如图 17.2 所示。鼠标箭头移动到报表区域周边，当鼠标形状变为双“+”字型箭头时，按下左键，可以拖动表格窗口，改变其在画面上的位置。将鼠标挪到报表窗口边缘带箭头的小矩形上，这时鼠标箭头形状变为与小矩形内箭头方向相同，按下鼠标左键并拖动，可以改变报表窗口的大小。当在画面中选中报表窗口时，会自动弹出报表工具箱，不选择时，报表工具箱自动消失。



图 17.1 工具箱按钮

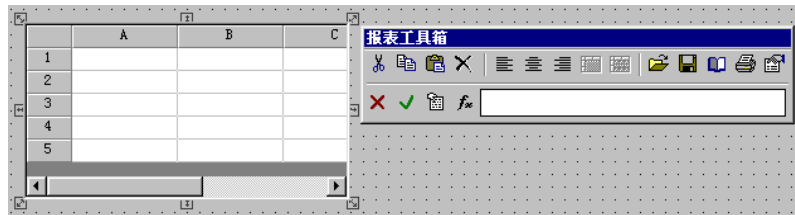


图 17.2 创建后的报表窗口

17.1.2 配置报表窗口的名称及格式套用

组态王中每个报表窗口都要定义一个唯一的标识名,该标识名的定义应该符合组态王的命名规则,标识名字符串的最大长度为 31。

用鼠标双击报表窗口的灰色部分(表格单元格区域外没有单元格的部分),弹出“报表设计”对话框,如图 17.3 所示。该对话框主要设置报表的名称、报表表格的行列数目以及选择套用表格的样式。



图 17.3 报表设计对话框

“报表设计”对话框中各项的含义为:

- ◇ **报表名称:** 在“报表控件名”文本框中输入报表的名称,如“实时数据报表”。

第十七章 报表系统



注意:

报表名称不能与组态王的任何名称、函数、变量名、关键字相同。

- ◇ **表格尺寸:** 在行数、列数文本框中输入所要制作的报表的大致行列数（在报表组态期间均可以修改）。默认为 5 行 5 列，行数最大值为 20000 行；列数最大值为 128 列。行用数字“1、2、3…”表示，列用英文字母“A、B、C、D…”表示。单元格的名称定义为“列标+行号”，如“a1”，表示第一行第一列的单元格。列标使用时不区分大小写，如“A1”和“a1”都可以表示第一行第一列的单元格。
- ◇ **套用报表格式:** 用户可以直接使用已经定义的报表模板，而不必再重新定义相同的表格格式。单击“表格样式”按钮，弹出“报表自动调用格式”对话框，如图 17.4 所示。如果用户已经定义过报表格式的话，则可以在左侧的列表框中直接选择报表格式，而在右侧的表格中可以预览当前选中的报表的格式。套用后的格式用户可按照自己的需要进行修改。在这里，用户可以对报表的套用格式列表进行添加或删除。
 - **添加报表套用格式:** 单击“请选择模板文件:”后的“…”按钮，弹出文件选择对话框，用户选择一个自制的报表模板 (*.rt1 文件)，单击“打开”，报表模板文件的名称及路径显示在“请选择模板文件:”文本框中。在“自定义格式名称:”文本框中输入当前报表模板被定义为表格格式的名称，如“格式 1”。单击“添加”按钮将其加入到格式列表框中，供用户调用。
 - **删除报表套用格式:** 从列表框中选择某个报表格式，单击“删除”按钮，即可删除不需要的报表格式。删除套用格式不会删除报表模板文件。

- **预览报表套用格式：**在格式列表框中选择一个格式项，则其格式显示在右边的表格框中。



图 17.4 报表自动套用格式对话框

定义完成后，单击“确认”完成操作，单击“取消”取消当前的操作。“套用报表格式”可以将常用的报表模板格式集中在这里，供随时调用，而不必在使用时再去一个个的查找模板。

套用报表格式的作用类似于报表工具箱中的“打开”报表模板功能。二者都可以在报表组态期间进行调用。

17.2 报表组态

17.2.1 认识报表工具箱与快捷菜单

报表创建完成后，呈现出的是一张空表或有套用格式的报表，还要对其进行加工——

第十七章 报表系统

报表组态。报表的组态包括设置报表格式、编辑表格中显示内容等。进行这些操作需通过“报表工具箱”中的工具或单击鼠标右键弹出的快捷菜单来实现。



图 17.5 报表工具箱和快捷菜单

报表工具箱中的按钮的含义如下：



剪切选中的一个或多个单元格中的内容，不包括单元格格式。剪切后，原单元格中的内容会被清除。剪切方法：

- 选中一个或多个单元格，从报表工具箱中单击“剪切”按钮
- 在报表上单击右键，从弹出的菜单中选择“剪切”项



复制选中的一个或多个单元格中的内容，不包括单元格格式。复制方法：

- 选中一个或多个单元格，从报表工具箱中单击“复制”按钮
- 在报表上单击右键，从弹出的菜单中选择“复制”项



将复制或剪切的单元格内容依次粘贴到当前单元格向右向下方向的单元格中。

粘贴方法:

- 选中一个或多个单元格, 从报表工具箱中单击“粘贴”按钮
- 在报表上单击右键, 从弹出的菜单中选择“粘贴”项

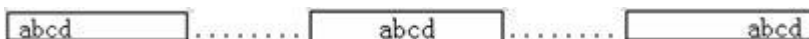


删除选中的一个或多个单元格中的内容, 单元格格式不变。删除方法:

- 选中一个或多个单元格, 从报表工具箱中单击“删除”按钮
- 在报表上单击右键, 从弹出的菜单中选择“清除内容”项



单元格显示内容的**对齐方式**: 靠左、居中、靠右, 如下图 17.6 所示。



17.6 单元格对齐方式



选中两个以上的单元格时**合并单元格**, 将所选择的单元格围成的矩形区域内的所有单元格合并为一个单元格, 合并后的单元格的内容及格式为所选择区域的左上角单元格的内容及格式。合并单元格方法:

- 选中两个以上的单元格, 单击报表工具箱中的“合并单元格”按钮
- 在报表上单击右键, 从弹出的菜单中选择“合并单元格”项



将选中的一个合并过的单元格**撤消合并**, 分解为基本单元格, 撤消合并后的各个单元格的内容及格式与合并单元格的内容及格式相同。撤消合并单元格方法:

- 选中一个合并过的单元格, 单击报表工具箱中的“撤消合并单元格”

第十七章 报表系统

按钮

- 在报表上单击右键，从弹出的菜单中选择“撤消合并单元格”项



打开一个报表模板到当前报表窗口中。单击该按钮后，弹出文件选择对话框，如图 17.7 所示，选择一个报表模板文件 (*.rtl)，单击打开，报表模板将加载到当前的报表中。

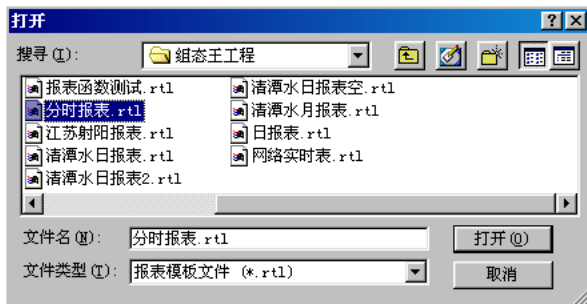


图 17.7 打开一个报表模板



将当前设计的报表**存储**为一个报表模板，单击该按钮，弹出文件存储对话框，如图 17.8 所示，选择存储路径，并输入要存储的报表模板的文件名，单击“保存”，模板文件存储为“*.rtl”文件。

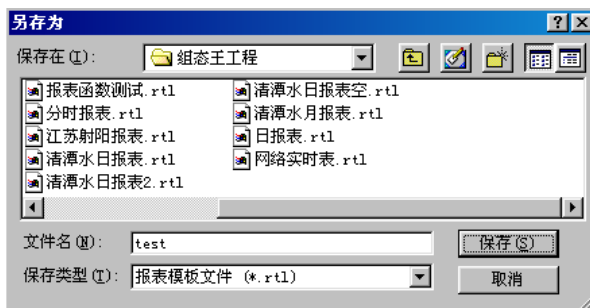


图 17.8 保存报表为一个报表模板



报表的**页面设置**，单击该按钮，弹出“页面设置”对话框，如图 17.9 所示，用户可以设置默认打印机、纸张大小、纸张来源、纸张方向、边距等。还可以设置报表的页眉、页脚的内容。单击“页眉”、“页脚”的下列列表框，如图 17.10 所示，从列表项中选择页眉、页脚要显示的内容。这里是报表在开发系统中的页面设置，在组态王运行系统中，可以通过函数实现页面设置，请参见报表函数一节。



第十七章 报表系统

图 17.9 对报表进行页面设置



图 17.10 页眉、页脚选项



报表打印预览：在开发系统中对设计好的报表进行打印预览，查看打印后的效果，进行打印预览时，系统会自动隐藏组态王的开发系统和运行系统。在打印预览中，也可以进行页面设置。执行打印预览时，有打印预览工具条，如下图所示。



- **打印：**弹出“打印属性”对话框，选择打印选项。
- **下一页、上一页：**如果报表比较大，超过了两页，选择翻页预览。
- **两页/一页：**当前以两页或一页来预览。两者切换进行。
- **放大/缩小：**放大/缩小预览页面。
- **设置：**对报表进行页面设置。



打印报表，单击该按钮，弹出“打印”对话框，如图 17.11，打印当前设计的报表（这里是开发环境下的打印，运行环境下使用报表打印函数 ReportPrint2()）。

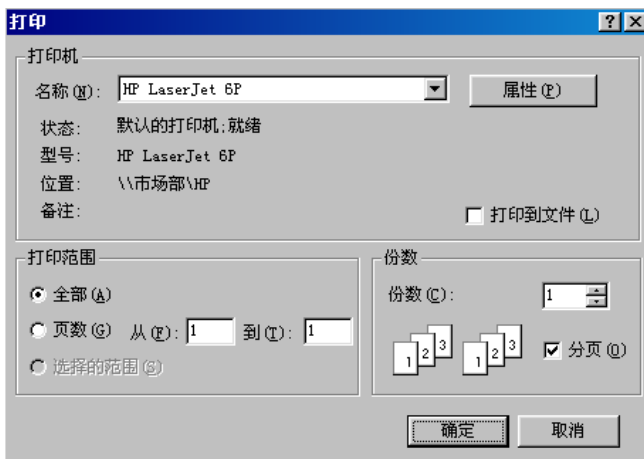


图 17.11 报表打印对话框



设置选中的**单元格格式**，包括：单元格格式，如数字型、日期型等；字体；对齐方式；单元格边框样式；单元格图案。单击该按钮，弹出“单元格格式设置”对话框。报表格式设置请参见 17.2.4 节。启动设置单元格格式方法：

- 选中要设置格式的单元格，从报表工具箱中单击“设置单元格格式”按钮
- 在报表上单击右键，从弹出的菜单中选择“设置单元格格式”项



取消上次对报表单元格的输入操作。



将报表工具箱中文本编辑框的内容**输入**到当前单元格中，当把要输入到某个单元格中的内容写到报表工具箱中的编辑框时，必须单击该按钮才能将文本

第十七章 报表系统

输入到当前单元格中。当用户选中一个已经有内容的单元格时，单元格的内容会自动出现在报表工具箱的编辑框中。

输入文本、公式等到单元格的方法：

- 将选择的组态王变量、报表公式、文本等写到报表工具箱中的编辑框中，然后单击“输入”按钮。
- 直接双击要编辑内容的单元格，使文本输入光标位于该单元格中，直接进行编辑。



注意：

在单元格中输入组态王变量、引用函数或公式时必须在其前加“=”。



插入组态王变量，单击该按钮，弹出组态王变量选择对话框。例如要在报表单元格中显示“\$时间”变量的值，首先在报表工具箱的编辑栏中输入“=”号，然后选择该按钮，在弹出的变量选择器中选择该变量，单击“确定”关闭变量选择对话框，这时报表工具箱编辑栏中的内容为“=\$时间”，单击工具箱上的“输入”按钮，则该表达式被输入到当前单元格中，运行时，该单元格显示的值能够随变量的变化随时自动刷新。



插入报表函数，单击该按钮弹出报表内部函数选择对话框。如图 17.12 所示，



图 17.12 报表内部函数选择对话框

在报表中快速**插入一行和一列**：在报表编辑过程中，如果需要在报表的某个位置插入一行，则选择插入位置行的任意一个单元格；如果需要在报表的某个位置插入一列，则选择插入位置列的任意一个单元格。例如在如图 17.13 所示的报表窗口中，要在“插入行”单元格的位置新插入一行，则选中该单元格，单击右键在弹出的快捷菜单中选择“插入”，弹出“插入”对话框，如图 17.14 所示，在对话框上选择“整行”选项，单击“确定”，关闭对话框。则在报表中的原第三行插入了一行，如图 17.15 所示，原第三行变为了第四行，其它行依次向下移动。插入的行的单元格格式继承上一行（本例中第二行）的格式。要在“插入列”的单元格位置插入一列，同样选中该单元格，单击右键选择快捷菜单上的“插入”项，在弹出的图 17.14 的对话框上选择“整列”，单击确定关闭对话框，则在“插入列”单元格的位置（本例中 C 列）的位置插入了新的一列，如图 17.16 所示，则原有的“插入列”

第十七章 报表系统

位置后的列依次向右移动。新的列的单元格格式继承其上一列的格式。如果在插入行、列的中间遇有合并的单元格，则该合并的单元格自动扩大一行或一列。



	A	B	C	D	E
1					
2					
3		插入行			
4			插入列		
5					

图 17.13 插入行列原表格

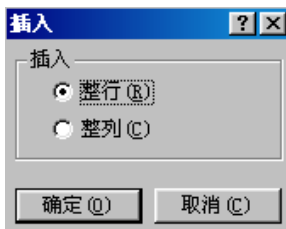


图 17.14 插入对话框



	A	B	C	D	E
1					
2					
3					
4		插入行			
5			插入列		
6					

图 17.15 插入行后的表格



	A	B	C	D	E	F
1						
2						
3						
4		插入行				
5				插入列		
6						

图 17.16 插入列后的表格

在报表中快速**删除一行或一列**：在报表编辑过程中，如果其中的某一行或某一列不再需要，则选中该行或该列的任意一个单元格，单击右键在弹出的菜单中选择“删除”项，弹出“删除”对话框，如图 17.17 所示，如果要删除一行，则选择“整行”选项，如果删除整列，则选择“整列”选项，选择后单击“确定”按钮，关闭对话框。则选中的行或列整个被删除。删除后其它行或列依次向上或向右移动，行号列标也发生变化。

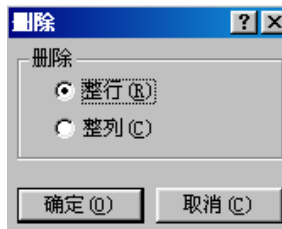


图 17.17 删除行列对话框

设置单元格只读属性：在报表中的某单元格或是用鼠标拖动选中多个单元格后，单击鼠标右键，在快捷菜单中选“只读”，则所选单元格在运行系统中不可编辑。

第十七章 报表系统

17.2.2 定义报表单元格的保护属性

组态王报表在系统运行过程中，用户可以直接在报表单元格中输入数据，修改单元格内容。为防止用户修改不允许修改单元格的内容，报表提供了一个保护属性——只读。

在开发环境中进行报表组态时，选择要保护的单元格区域，单击鼠标右键，在弹出的快捷菜单中选择“只读”，被保护的单元格在系统运行时不允许用户修改单元格内容。要查看某个单元格是否被定义为只读属性，方法为在单元格上单击鼠标右键，如果快捷菜单上的“只读”项前有“√”符号，则表明该单元格被定义了只读属性。再次选择该菜单项时取消保护属性。

 注意：

用户在系统运行过程中在修改含有表达式的单元格的内容后，会在当前运行画面清除原表达式。只有重新关闭、打开画面后才能恢复该表达式。

17.2.3 报表的其它快捷编辑方法

报表的其它编辑方法有：

- ◇ 鼠标左键单击某个单元格后拖动则为选择多个单元格。区域的左上角为当前单元格。
- ◇ 鼠标左键单击固定行或固定列（报表中标识行号列标的灰色单元格）为选择整行或整列。单击报表左上角的灰色固定单元格为全选报表单元格。
- ◇ 单击报表左上角的固定单元格为选择整个报表。

✧ 允许在获得焦点的单元格直接输入文本。用鼠标左键单击单元格或双击单元格使输入光标位于该单元格内，输入字符。按下回车键或鼠标左键单击其它单元格为确认输入，<ESC>键取消本次输入。

✧ 允许通过鼠标拖动改变行高、列宽。将鼠标移动到固定行或固定列之间的分割线上，鼠标形状变为双向黑色箭头时，按下鼠标左键，拖动，修改行高、列宽。当处于运行状态时，可以通过快捷键来对单元格高度和宽度进行设置。

具体操作如下：

1. 在运行态点选表格；
2. 按“Ctrl+f”，单元格的宽度和高度就被锁定，即不能被修改；
3. 按“Ctrl+j”，解除锁定。

✧ 单元格文本的第一个字符若为“=”，则其它的字符为组态王的表达式，该表达式允许由已定义的组态王的变量、函数、报表单元格名称等组成；否则为字符串。

17.2.4 设置报表格式

在报表工具箱中单击“设置单元格格式”按钮或在菜单中选择“设置单元格格式”项，弹出“设置单元格格式”对话框，如图 17.18 所示。

第十七章 报表系统



图 17.18 设置单元格格式对话框

“设置单元格格式”对话框包括数字、字体、对齐、边框、图案等五个属性页。下面详细介绍。

1、数字属性页：

设置选中的单元格中的数值的类型，从“分类”列表框中选择，运行时有效：

- ◇ 常规：如图 17.18 所示，单元格不包括任何特殊的格式。
- ◇ 数值：如图 17.19 所示，单元格中为数字时，可以设置显示时的数值的小数点位数，选择“千分位分割符”选项，当数值位数较多时，自动添加千分位分割符。仅在单元格显示数字时有效。

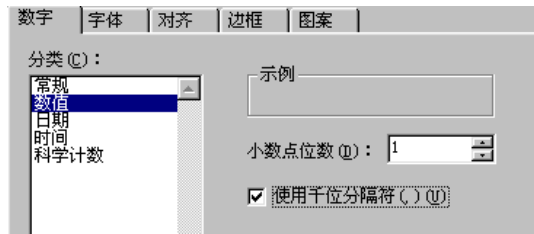


图 17.19

- ◇ 日期：如图 17.20 所示，当单元格需要显示日期的特殊形式，仅在设置了日期函数 Date（参见组态王报表函数）的单元格中有效，可以任选其一。



图 17.20

- ◇ 时间：如图 17.21 所示，当单元格需要显示时间的特殊形式，仅在设置了时间函数 Time（参见组态王报表函数）的单元格中有效，可以任选其一。

第十七章 报表系统

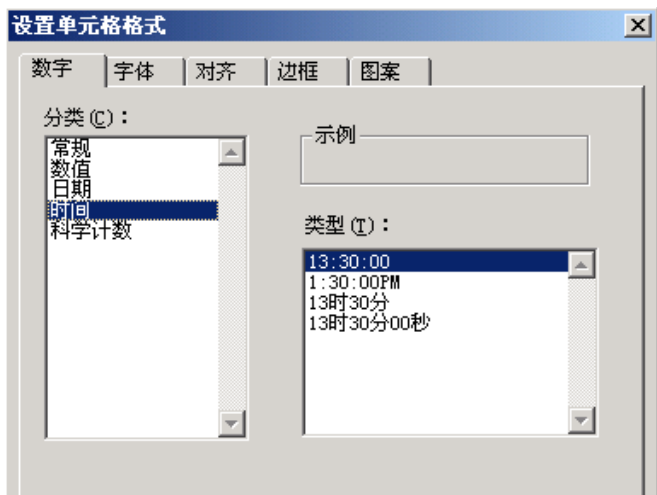


图 17.21



注意：

对于组态王的系统变量\$时间和\$日期在单元格显示时不能改变其显示形式。

- 科学计数：如图 17.22 所示，将单元格的数值显示为科学计数型。仅在单元格显示数字时有效。



图 17.22

2、字体属性页：

设置选中的单元格的字体。如图 17.23 所示。可以任选字体、字形、字号。单击颜色标签下的带颜色按钮，从弹出的调色板中选择字体颜色。选择字体的特殊效果：删除线、

下划线。可以在“预览”窗口中看到设置的字体的效果。



图 17.23 字体属性页

3、对齐属性页：

设置选中的单元格中的内容的对齐方式，分为水平、垂直两项。如图 17.24 所示。

- ◇ 水平有常规（左对齐）、靠左、居中、靠右、两端对齐等方式；
- ◇ 垂直有常规（靠上）、靠上、居中、靠下、两端对齐等选项。



图 17.24 对齐属性页

4、**边框属性页**：设置选中的单元格的边框样式、线形、边框颜色等。如图 17.25 所示。

◇ **预置**：A、无：设置所有选中的单元格为没有边框。

B、外边框：用选定的线形和线条颜色设置所选择的单元格矩形区域的区域周边的单元格的边框。

C、内部：用选定的线形和线条颜色设置所选择的单元格矩形区域的区域内的单元格的边框。

◇ **线条**：选择设置单元格边框所需要的线条样式和线条颜色。在线条列表中直接单击相应项选择线条样式，单击颜色标签下的带颜色按钮，从弹出的调色板中选择线条颜色。

◇ **边框**：单独设置单元格的某个边或添加斜线，设置单元格边框时，在边框项中可以

进行预览。

创建报表后，默认的表格线为灰色，只能显示，不能打印。

 注意：

定义表格的边框时，先选择线条样式、线条颜色，然后选择添加边框的位置。



图 17.25 边框属性页

4、图案属性页

设置单元格底部颜色和单元格填充样式。如图 17.26 所示。单击“单元格底纹”项的颜色按钮，从弹出的调色板中选择单元格的背景填充颜色从“单元格图案”中的“图案”按钮中选择单元格填充样式：依次为全填充、不填充、左斜线、水平垂直网状线、斜网状线、右斜线、水平线、垂直线填充等。单击“颜色”按钮，弹出调色板，选择填充的颜色。所选择的单元格图案填充效果可从右边的“示例”中进行预览。

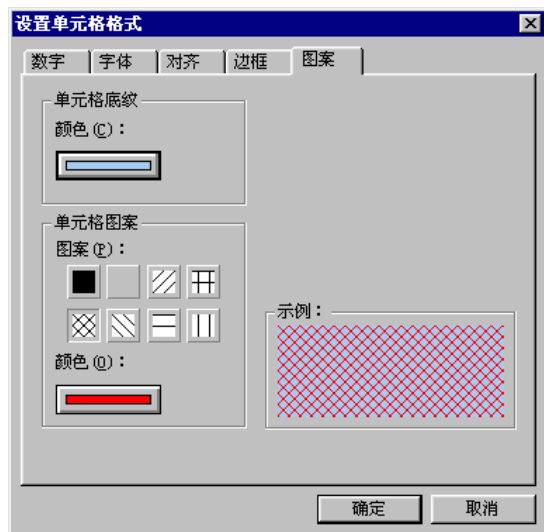


图 17.26 图案属性页

17.3 报表函数

报表在运行系统中单元格中数据的计算、报表的操作等都是通过组态王提供的一整套报表函数实现的。报表函数分为报表内部函数、报表单元格操作函数、报表存取函数、报表历史数据查询函数、统计函数、报表打印函数等。

17.3.1 报表内部函数

报表内部函数是指只能在报表单元格内使用的函数，有数学函数、字符串函数、统计函数等。其基本上都是来自于组态王的系统函数，使用方法相同，只是函数中的参数发生了变化，减少了用户的学习量，方便学习和使用。组态王的报表函数中的参数和有关用报

表单元格作为参数的函数，其中的参数引用均为这种方法。当参数为多个单元格时：

- ◇ 如果是任选多个单元格，则使用方法为用逗号将各个单元格的表示分隔，如：
‘a1, b3, c6, h10’。
- ◇ 如果选择的为连续的单元格时，可以输入第一个单元格标识和最后一个单元格标识，中间用冒号分割。如选择了 a1 到 c10 间的单元格区域：‘a1:c10’。
- ◇ 报表内部函数中的单元格参数可以使用组态王变量代替，即报表支持的组态王系统函数可以直接在报表中使用。
- ◇ 合并单元格中的数值不论对齐方式如何，在进行函数运算时，合并单元格中的数值都将放在左上角被运算。



例如：

	A	B	C
1			
2	aa0	=\\本站点\ a0	=\\本站点\ a0
3	aa1	=\\本站点\ a1	

其中：B3 和 C3 为合并单元格；

\\本站点\ a0=1； \\本站点\ a1=1；

组态王运行时，求和的结果为：

$\text{Sum}('b2:b3')=2$

$\text{Sum}('c2:c3')=1$



注意：

第十七章 报表系统

1. 在单元格中使用报表函数时，必须在函数前加“=”号，否则按字符串处理。
 2. 函数中将单元格作为参数时，单元格参数须用单引号括起来。
 3. 除特殊标明的外，报表内部函数（用单元格作参数）只能用于报表的单元格中，不能用于命令语言中。
-

下面详细介绍报表函数的功能和使用方法：

✧ Abs:

函数功能：计算某个单元格内的数据的绝对值，并显示到当前单元格中。

使用格式：=Abs（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 b10 的数值的绝对值：=Abs（‘b10’）

✧ ArcCos:

函数功能：计算某个单元格内的数据的反余弦值，显示到当前单元格中。

使用格式：=ArcCos（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a6 的数值的反余弦值：=ArcCos（‘a6’）

✧ ArcSin:

函数功能：计算某个单元格内的数据的反正弦值，显示到当前单元格中。

使用格式：=ArcSin（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a5 的数值的反余弦值：=ArcSin（‘a5’）

✧ **ArcTan:**

函数功能：计算某个单元格内的数据的反正切值，显示到当前单元格中。

使用格式：=ArcTan（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a5 的数值的反正切值：=ArcTan（‘a5’）

✧ **Bit:**

函数功能：获取某个单元格的整型或实型数据的某一位的值（0 或 1），显示到当前单元格中。

使用格式：=Bit（‘数据源单元格列标行号’，位的序号） 位的序号：取值为 1 至 16

返回值：0 或 1

举例：取单元格 a5 的数值的第 5 位：=Bit（‘a5’，5）

✧ **Cos:**

函数功能：计算某个单元格内的数据的余弦值，显示到当前单元格中。

使用格式：=Cos（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a5 的数值的余弦值：=Cos（‘a5’）

✧ **Date:**

函数功能：根据单元格给出的年、月、日整型数，返回与当前单元格定义格式相

第十七章 报表系统

同的日期字符串，显示在当前单元格中。默认格式为：年-月-日。单元格日期格式在“设置单元格格式”中的“常规”属性页中设置。

使用格式：=Date（‘a1’，‘a2’，‘a3’）或=Date（long a，long b，long c）

举例：如年、月、日所在的单元格分别为：b5、b6、b7，在 b8 中显示有以上三个整数决定的日期字符串，则在 b8 单元格中输入：=Date（‘b5’，‘b6’，‘b7’）。

✧ Exp:

函数功能：计算以某个单元格内的数据为指数的 e^x 的计算结果，显示到当前单元格中。

使用格式：=Exp（‘数据源单元格列标行号’）

返回值：实型

举例：取单元格 a5 的数值为指数的 e^x 的计算结果：=Exp（‘a5’）

✧ Int:

函数功能：对某个单元格内的数据进行取整，显示到当前单元格中。

使用格式：=Int（‘数据源单元格列标行号’）

返回值：整型

举例：对单元格 a5 的数值进行取整：=Int（‘a5’）

✧ LogE:

函数功能：计算某单元格内的数据的对数函数 Log_e^x 的值，显示到当前单元格中。

使用格式：=LogE（‘数据源单元格列标行号’）

返回值：实型

举例：取单元格 a5 的数值的对数函数 Log_e^x 的值：=LogE（‘a5’）

✧ **LogN:**

函数功能：计算某单元格内的数据的以 N 为底的对数的值，显示到当前单元格中。

使用格式：=LogN（‘数据源单元格列标行号’，底数）底数：对数中做底的数

返回值：实型

举例：取单元格 a5 的数值以 10 为底的对数函数 Log_e^x 的值：=LogN（‘a5’，10）

✧ **Max:**

函数功能：对两个以上的单元格内的数据进行比较，取出最大值，显示到当前单元格中。

使用格式：=Max（‘单元格区域’）

举例：

A) 对单元格 a1、b3、c8、i10 的数值进行比较取出最大值：=Max（‘a1’，‘b3’，‘c8’，‘i10’）

B) 对单元格 b5 至 b12 之间的单元格进行比较取最大值：=Max（‘b5:b12’）

✧ **Min:**

函数功能：对两个以上的单元格内的数据进行比较，取出最小值，显示到当前单元格中。

使用格式：=Min（‘单元格区域’）

举例：

A) 对单元格 a1、b3、c8、i10 的数值进行比较取出最小值：=Min（‘a1’，‘b3’，

第十七章 报表系统

‘c8’ , ‘i10’)

B) 对单元格 b5 至 b12 之间的单元格进行比较取最小值: =Min (‘b5:b12’)

❖ Pow:

函数功能: 计算某个单元格内的数据的任意次幂, 显示到当前单元格中。

使用格式: =Pow (‘数据源单元格列标行号’ , 指数)

举例: 计算单元格 a5 为底的数值的 4 次幂: =Pow (‘a5’ , 4)

❖ Sgn:

函数功能: 返回某个单元格内的数据的符号, 显示到当前单元格中。

使用格式: =Sgn (‘数据源单元格列标行号’)

返回值: 如果为正数, 返回 1

如果为 0, 返回 0

如果为负数, 返回-1

举例: 返回单元格 a5 的数值的符号: =Sgn (‘a5’)

❖ Sin:

函数功能: 计算某个单元格内的数据的正弦值, 显示到当前单元格中。

使用格式: =Sin (‘数据源单元格列标行号’)

返回值: 整型或实型

举例: 取单元格 a5 的数值的正弦值: =Sin (‘a5’)

❖ Sqrt:

函数功能: 计算某个单元格内的数据的平方根, 显示到当前单元格中。

使用格式：=Sqrt（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a5 的数值的平方根：=Sqrt（‘a5’）

✧ **StrAscII:**

函数功能：返回某单元格的字符串首字符的 AscII 码值，显示到当前单元格中。

使用格式：=StrAscII（‘数据源单元格列标行号’）

举例：取单元格 a5 的字符串首字符的 AscII 码值：= StrAscII（‘a5’）

✧ **StrChar:**

函数功能：返回某个单元格的 AscII 码对应的字符，显示到当前单元格中。

使用格式：=StrChar（‘数据源单元格列标行号’）

举例：取单元格 a5 的 AscII 码对应的字符：= StrChar（‘a5’）

✧ **StrFromInt:**

函数功能：将某个单元格内的**整数**数值转换为另一进制下的字符串，显示到当前单元格中。

使用格式：=StrFromInt（‘数据源单元格列标行号’，进制）
进制：用来转换的进制

举例：取单元格 a5 整数转换为 8 进制字符串：= StrFromInt（‘a5’，8）

✧ **StrFromReal:**

函数功能：将某单元格内的**实数**数值转换为字符串，显示到当前单元格中。

使用格式：=StrFromReal（‘数据源单元格列标行号’，小数位，表示类型）小

第十七章 报表系统

数位：指定要显示多少个小数位

表示类型：“f”按照浮点数显示

“e”按照小写“e”的指数制显示

“E”按照大写“E”的指数制显示

举例：取单元格 a5 的实数转换为 2 位小数的以浮点数表示的字符串：=

StrFromReal（‘a5’，2，“f”）

✧ StrFromTime:

函数功能：将某个单元格内的**时间值**（1970年1月1日起，以秒为单位）转换为字符串，显示到当前单元格中。

使用格式：=StrFromTime（‘数据源单元格列标行号’，字符串格式）

字符串格式：1、以 WINDOWS 控制面板相同的格式显示日期

2、以 WINDOWS 控制面板相同的格式显示时间

3、同时显示日期和时间

举例：取单元格 a5 的时间值转为时间字符串：= StrFromTime（‘a5’，2）

✧ StrInStr:

函数功能：返回要查找的字符串在数据源单元格中第一次出现的位置，显示到当前单元格中。

使用格式：=StrInStr（‘数据源单元格’，要查找的字符串，开始查找的位置，是否区分大小写）

举例：在单元格 a5 的字符串中从第 2 位开始查找字符串“abc”在其中出现的

位置, 区分大小写: = StrInStr (‘a5’ , “abc”, 2, 1)

✧ **StrLeft:**

函数功能: 返回数据源单元格中最左边指定数目的字符串, 显示到当前单元格中。

使用格式: =StrLeft (‘数据源单元格’, 个数) 个数: 返回字符的数目, 如果为 0, 则返回全部字符串

举例: 在单元格 a5 字符串中取出左边 4 个字符: = StrLeft (‘a5’ , 4)

✧ **StrLen:**

函数功能: 返回数据源单元格中字符串的长度, 显示到当前单元格中。

使用格式: =StrLen (‘数据源单元格’)

举例: 返回单元格 a5 的字符串的长度: = StrLen (‘a5’)

✧ **StrLower:**

函数功能: 将数据源单元格中的字符串中的大写字母全部转换为小写字母, 显示到当前单元格中, 其它小写字母、数字和特殊字符不受影响。

使用格式: =StrLower (‘数据源单元格’)

举例: 将单元格 a5 中的字符串内的大写字母转换为小写字母: = StrLower (‘a5’)

✧ **StrMid:**

函数功能: 从数据源单元格中的字符串中指定的位置开始取出指定长度的字符串, 显示到当前单元格中。

使用格式: =StrMid (‘数据源单元格’, 开始位置, 长度)

第十七章 报表系统

举例：从单元格 a5 的字符串中第 2 为开始取出 4 个字符：=StrMid（‘a5’，2，4）

✧ StrReplace:

函数功能：将数据源单元格中的字符串中指定的字符串替换为给定的字符，显示到当前单元格中。

使用格式：=StrReplace（‘数据源单元格’，参数 1，参数 2，参数 3，参数 4，参数 5）

参数 1：要被替换的目标字符串

参数 2：替换字符串

参数 3：是否区分大小写（0：不区分； 1：区分）

参数 4：确定替换的次数（0 为全部替换）

参数 5：是否全字匹配（0：否； 1：是）

举例：从单元格 a5 中的字符串中的字符串“abc”替换为“mn”，区分大小写，替换 2 次，全字匹配：= StrReplace（‘a5’，“abc”，“mn”，1,2,1）

✧ StrRight:

函数功能：返回数据源单元格最右边指定数目的字符串，显示到当前单元格中。

使用格式：=StrRight（‘数据源单元格’，个数）个数：返回字符的数目，如果为 0，则返回全部字符串

举例：在单元格 a5 的字符串中取右边 4 个字符：= StrRight（‘a5’，4）

✧ StrSpace:

函数功能：在字符串给定的位置处产生若干空格，显示到当前单元格中。

使用格式: =StrSpace (产生个数)

举例: A) = “值班” +StrSpace (2) + “人员”, 结果为 “值班 人员”

B) = ‘a5’ StrSpace (2) + ‘a6’ , 结果为单元格 a5 的字符串加两个空格在加单元格 a6 的字符串

❖ **StrToInt:**

函数功能: 将数据源单元格中的由数字组成的字符串转换为整型数值, 显示到当前单元格中。转换取得的数值直到在源字符串中遇到非数字字符为止, 若首字符为非数字字符, 则返回值为 0。

使用格式: =StrToInt (‘数据源单元格’)

举例: =StrToInt (‘a5’)

❖ **StrToReal:**

函数功能: 将数据源单元格中的由数字组成的字符串转换为实型数值, 显示到当前单元格中。转换取得的数值直到在源字符串中遇到非数字字符为止, 若首字符为非数字字符, 则返回值为 0。

使用格式: =StrToReal (‘数据源单元格’)

举例: =StrToReal (‘a5’)

❖ **StrTrim:**

函数功能: 去掉数据源单元格的字符串中指定位置的空格, 显示到当前单元格中。

使用格式: =StrTrim (‘数据源单元格’, 删除类型)

删除类型: 1、字符串左边的空格

第十七章 报表系统

2、字符串右边的空格

3、删除字符串中单词间单个空格以外的空格

举例：如单元格 a5 的字符串为“ This is a test! ”，使用函数：=StrTrim ('a5' , 2)，结果为“ This is a test!”。

❖ StrType:

函数功能：检测数据源单元格中字符串的首字符是否为指定类型，判断结果显示到当前单元格中（1：是； 0：否）。

使用格式：=StrType（‘数据源单元格’，检测类型）

检测类型：1、字母、数字（A~Z、a~z、0~9）

2、数字（0~9）

3、字母（A~Z、a~z）

4、大写字母（A~Z）

5、小写字母（a~z）

6、标点符号（0x21~0x2F）

7、ASCII 码字符

8、十六进制字符（A~F、a~f、0~9）

9、可打印字符（0x20~0x7E）

10、控制字符（0x00~0x1F 或 0x20）

11、空白符（0x09~0x0D 或 0x20）

举例：判断单元格 a5 的字符串首字符是否为大写：=StrType（‘a5’，4）

✧ **StrUpper:**

函数功能：将数据源单元格中的字符串中的小写字母全部转换为大写字母，显示到当前单元格中，其它大写字母、数字和特殊字符不受影响。

使用格式：=StrUpper（‘数据源单元格’）

举例：将单元格 a5 中的字符串内的小写字母转换为大写字母：= StrUpper(‘a5’)

✧ **Tan:**

函数功能：计算某个单元格内的数据的正切值，显示到当前单元格中。

使用格式：=Tan（‘数据源单元格列标行号’）

返回值：整型或实型

举例：取单元格 a5 的数值的正切值：=Tan（‘a5’）

✧ **Text:**

函数功能：将某单元格内的数据设定为特殊格式，显示到当前单元格中。

使用格式：=Text（‘数据源单元格列标行号’，指定格式）

举例：设定单元格 a5 的数值格式为“0#.00”：=Text（‘a5’，“0#.00”）

✧ **Time:**

函数功能：根据单元格给出的时、分、秒整型数，返回与当前单元格定义格式相同的时间字符串，显示在当前单元格中。默认格式为：时：分：秒。单元格时间格式在“设置单元格格式”中的“常规”属性页中设置。

使用格式：=Time（‘a1’，‘a2’，‘a3’）或=Time（long a，long b，long c）

举例：如时、分、秒所在的单元格分别为：b5、b6、b7，在 b8 中显示有以上三

第十七章 报表系统

个整数决定的时间字符串,则在 b8 单元格中输入:=Time(‘b5’,‘b6’,‘b7’)。

✧ Trunc:

函数功能: 将某个单元格内的数据删除小数点右边的部分取得一个实数, 显示到当前单元格中。

使用格式: =Text (‘数据源单元格列标行号’)

举例: 将单元格 a5 的数值截取实数: =Trunc (‘a5’)

逻辑运算符:

逻辑运算符请参见 10.2.1 节“运算符”一节, 使用方法相同。

以上介绍了组态王的报表内部函数, 其余参数的设置可参见《组态王命令语言函数速查手册》。

17.3.2 报表的单元格操作函数

运行系统中, 报表单元格是不允许直接输入的, 所以要使用函数来操作。单元格操作函数是指可以通过命令语言来对报表单元格的内容进行操作, 或从单元格获取数据的函数。这些函数大多只能用在命令语言中。

✧ ReportSetCellValue

Long nRet = ReportSetCellValue(String szRptName, long nRow, long nCol, float fValue)

函数功能: 将指定报表的指定单元格设置为给定值。

返回值: 整型 0---成功;

 -1---行列数小于等于零;

-2---报表名称错误;

-3---设置值失败

参数说明: szRptName: 报表名称

Row: 要设置数值的报表的行号(可用变量代替)

Col: 要设置数值的报表列号(这里的列号使用数值,可用变量代替)

Value: 要设置的数值

举例: 根据组态王实型变量“压力”的数据变化设置报表“实时数据报表”的第2行第4列为变量“压力”的值,并且返回设置是否成功结果“实数设置结果”(组态王变量),在数据改变命令语言中输入:

实数设置结果=ReportSetCellValue(“实时数据报表”,2,4,压力);

◇ ReportSetCellString

Long nRet = ReportSetCellString(String szRptName, long nRow, long nCol, StringsValue)

函数功能: 将指定报表的指定单元格设置为给定字符串。

返回值: 整型 0---成功;

-1---行列数小于等于零;

-2---报表名称错误;

-3---设置文本失败

参数说明: szRptName: 报表名称

Row: 要设置数值的报表的行号(可用变量代替)

Col: 要设置数值的报表列号(这里的列号使用数值,可用变量代替)

第十七章 报表系统

Value: 要设置的文本

举例：根据组态王实型变量“压力”的数据变化设置报表“实时数据报表”的第2行第5列为字符串变量“压力说明”的值，并且返回设置是否成功结果“字符串设置结果”（组态王变量），在数据改变命令语言中输入：

字符串设置结果=ReportSetCellString(“实时数据报表”, 2, 5, 压力说明);

◇ **ReportSetCellValue2**

Long nRet = ReportSetCellValue2(String szRptName, long nStartRow, long nStartCol, long nEndRow, long nEndCol, float fValue)

函数功能：将指定报表的指定单元格区域设置为给定值。

返回值：整型

- 0---成功;
- 1---行列数小于等于零;
- 2---报表名称错误;
- 3---设置值失败

参数说明：szRptName：报表名称

StartRow：要设置数值的报表的开始行号（可用变量代替）

StartCol：要设置数值的报表开始列号（这里的列号使用数值，可用变量代替）

EndRow：要设置数值的报表的结束行号（可用变量代替）

EndCol：要设置数值的报表结束列号（这里的列号使用数值，可用变量代替）

Value：要设置的数值

举例：根据组态王实型变量“压力”的数据变化设置报表“实时数据报表”的第3行第

4 列到第 6 行第 7 列区域为变量“压力”的值，并且返回设置是否成功结果“实数设置结果 2”（组态王变量），在数据改变命令语言中输入：

实数设置结果 2=ReportSetCellValue2(“实时数据报表”, 3, 4, 6, 7, 压力);

◇ ReportSetCellString2

Long nRet = ReportSetCellString2(String szRptName, long nStartRow, long nStartCol, long nEndRow, long nEndCol, String szValue)

函数功能：将指定报表指定单元格设置为给定字符串。

返回值：整型 0---成功；

-1---行列数小于等于零；

-2---报表名称错误；

-3---设置文本失败

参数说明：szRptName：报表名称

StartRow：要设置数值的报表的开始行号（可用变量代替）

StartCol：要设置数值的报表开始列号（这里的列号使用数值，可用变量代替）

StartRow：要设置数值的报表的开始行号（可用变量代替）

StartCol：要设置数值的报表开始列号（这里的列号使用数值，可用变量代替）

Value：要设置的文本

举例：根据组态王实型变量“压力”的数据变化设置报表“实时数据报表”的第 8 行第 5 列到第 10 行第 7 列为字符串变量“压力说明”的值，并且返回设置是否成功结果“字符串设置结果 2”（组态王变量），在数据改变命令语言中输入：

第十七章 报表系统

字符串设置结果 2=ReportSetCellString2(“实时数据报表”, 8, 5, , 10, 7, 压力说明);

◇ **ReportGetCellValue**

```
float fValue = ReportGetCellValue(String szRptName, long nRow, long nCol)
```

函数功能：获取指定报表的指定单元格的数值。

返回值：实型

参数说明：szRptName：报表名称

Row：要获取数据的报表的行号（可用变量代替）

Col：要获取数据的报表列号（这里的列号使用数值，可用变量代替）

举例：获取报表“实时数据报表”中的第 2 行第 4 列的数值，赋给实型变量“数值”：数

```
值= ReportGetCellValue(“实时数据报表”, 2, 4);
```

◇ **ReportGetCellString**

```
String szValue = ReportGetCellString(String szRptName, long nRow, long nCol,)
```

函数功能：获取指定报表的指定单元格的文本。

返回值：字符串型

参数说明：szRptName：报表名称

Row：要获取文本的报表的行号（可用变量代替）

Col：要获取文本的报表列号（这里的列号使用数值，可用变量代替）

举例：获取报表“实时数据报表”中的第 2 行第 5 列的文本，赋给字符串型变量“文本”：

```
文本= ReportGetCellString(“实时数据报表”, 2, 5);
```

◇ **ReportGetRows**

Long nRows = ReportGetRows(String szRptName)

函数功能：获取指定报表的行数

参数说明：szRptName：报表名称

举例：获取报表“实时数据报表”的行数，赋给变量“行数”：行数= ReportGetRows(“实时数据报表”);

◇ ReportGetColumns

Long nCols = ReportGetColumns(String szRptName)

函数功能：获取指定报表的列数

参数说明：szRptName：报表名称

举例：获取报表“实时数据报表”的列数，赋给变量“列数”：列数= ReportGetColumns(“实时数据报表”);

◇ ReportGetFocusCell

ReportGetFocusCell(const char *pReportName);

函数功能：此函数用于在使用报表时，获取指定报表选中单元格的行列数。

参数说明：pReportName:报表名称。

返回值说明：

返回值为 X-XXXXX-XXX 格式的字符串。其中第一个有效字符只有两个值：1 和 0；1 表示函数调用成功，0 表示函数调用失败。第 2 个有效字符串表示行数，固定占 5 个字符，不够时高位填 0；第 3 个有效字符串表示列数，固定占 3 个字符，不够时高位填 0。

举例：

ReportGetFocusCell(“日报表”) 返回值为：1-00007-003

第十七章 报表系统

这个函数表示获取日报表的选择的单元格，返回值表示：获取成功，当前日报表被选中的单元格为第 7 行第 3 列的单元格。

注意事项：若选择多个单元格时，返回的是报表中与其他被选中单元格颜色不同的单元格

◇ ReportSetRow

ReportSetRows(String szRptName, long RowNum)

函数功能：设置指定报表的行数

参数说明：szRptName：报表名称

RowNum：要设置的行数

举例：将“实时数据报表”报表的行数设置为 1000 行：

```
ReportSetRows(“实时数据报表”, 1000);
```

◇ ReportSetColumns

ReportSetColumns(String szRptName, long ColumnNum)

函数功能：设置指定报表的列数

参数说明：szRptName：报表名称

ColumnNum：要设置的列数

举例：将“实时数据报表”报表的列数设置为 1000 列：`ReportSetColumns (“实时数据报表”, 1000);`

17.3.3 存取报表函数

存取报表函数主要用于存储指定报表和打开查阅已存储的报表。用户可利用这些函数

保存和查阅历史数据、存档报表。

◇ **ReportSaveAs**

Long nRet = ReportSaveAs(String szRptName, String szFileName)

函数功能：将指定报表按照所给的文件名存储到指定目录下，ReportSaveAs 支持将报表文件保存为 rtl、xls、csv 格式。保存的格式取决于所保存的文件的后缀名。

参数说明：szRptName：报表名称

szFileName：存储路径和文件名称，

返回值：返回存储是否成功标志 0 - 成功

举例：将报表“实时数据报表”存储为文件名为“数据报表 1”，路径为“C:\My Documents”，

返回值赋给变量“存文件”：

存文件 = ReportSaveAs(“实时数据报表”, ”C:\My Documents\数据报表 1.Rtl”);

◇ **ReportLoad**

Long nRet = ReportLoad(String szRptName, String szFileName)

函数功能：将指定路径下的报表读到当前报表中来。ReportLoad 支持读取 rtl 格式的报表文件。报表文件格式取决于所保存的文件的后缀名。

参数说明：szRptName：报表名称

szFileName：报表存储路径和文件名称

返回值：返回存储是否成功标志 0 - 成功

举例：将文件名为“数据报表 1”，路径为“C:\My Documents”的报表读取到当前报表中，

返回值赋给变量“读文件”：读文件 = ReportLoad(“实时数据报表”, ”C:\My

第十七章 报表系统

Documents\数据报表 1. Rtl”);

17.3.4 报表统计函数

◇ Average:

函数功能：对指定单元格区域内的单元格进行求平均值运算，结果显示在当前单元格内。

使用格式：=Average（‘单元格区域’）

举例：=Average（‘a1’，‘b2’，‘r10’）任意单元格选择求平均值

=Average（‘b1:b10’）连续的单元格求平均值

◇ Sum:

函数功能：将指定单元格区域内的单元格进行求和运算，显示到当前单元格内。单元格区

域内出现空字符、字符串等都不会影响求和。

使用格式：=Sum（‘单元格区域’）

举例：=Sum（‘a1’，‘b2’，‘r10’）； 任意单元格选择求和

=Sum（‘b1:b10’）； 连续的单元格求和

17.3.5 报表历史数据查询函数

报表历史数据查询函数将按照用户给定的起止时间和查询间隔，从组态王历史库或工业库中查询数据，并填写到指定报表上。

◇ ReportSetHistData()

ReportSetHistData(String szRptName, String szTagName, Long nStartTime, Long

nSepTime, String szContent) ;

函数功能：按照用户给定的参数查询历史数据

参数说明：

szRptName：要填写查询数据结果的报表名称

szTagName：所要查询的变量名称

StartTime：数据查询的开始时间，该时间是通过组态王 HTConvertTime 函数转换的以 1970 年 1 月 1 日 8: 00: 00 为基准的长整型数，所以用户在使用本函数查询历史数据之前，应先将查询起始时间转换为长整型数值。

SepTime：查询的数据的时间间隔，单位为秒

szContent：查询结果填充的单元格区域

举例：查询变量“压力”自 2001 年 5 月 1 日 8: 00: 00 以来的数据，查询间隔为 30 秒，数据报表填充的区域为” a2:a100”：

```
long StartTime; (StartTime 为自定义变量)
```

```
StartTime=HTConvertTime(2001, 5, 1, 8, 0, 0);
```

```
ReportSetHistData(“历史数据报表”, “压力”, StartTime, 30,  
“a2:a100”);
```



注意：

当查询的数据行数大于报表设计的行数时，系统将自动将添加行数，满足数据填充的需要，最大行数可到 20000 行。

第十七章 报表系统

◇ ReportSetTime

```
ReportSetTime(String szRptName, Long nStartTime, Long nSepTime, String  
szContent)
```

函数功能：向报表设置连续的时间字符串，配合函数 ReportSetHisData 设置返回的历史数据的时间。

参数说明：szRptName：要填写查询数据结果的报表名称。

szTagName：所要查询的变量名称。

StartTime：数据查询的开始时间，该时间是通过组态王 HTConvertTime：函数转换的以 1970 年 1 月 1 日 8: 00: 00 为基准的长整型数，所以用户在使用本函数查询历史数据之前，应先将查询起始时间转换为长整型数值。

SepTime：查询的数据的时间间隔，单位为秒。

szContent：查询结果填充的单元格区域。

举例：在报表中插入的时间为自 2001 年 5 月 1 日 8: 00: 00 以来的时间段，时间段间隔为 30 秒，数据报表填充的区域为” a2:a100”：

```
long StartTime; (StartTime 为自定义变量)
```

```
StartTime=HTConvertTime(2001, 5, 1, 8, 0, 0);
```

```
ReportSetTime (“历史数据报表”, StartTime, 30, “a2:a100”);
```

◇ ReportSetHistData2()

```
ReportSetHistData2(StartRow, StartCol);
```

函数参数：StartRow：指定数据查询后，在报表中开始填充数据的起始行

StartCol：指定数据查询后，在报表中开始填充数据的起始列这两个参数可以省略不写（应同时省略），省略时默认值都为 1。函数功能：使用该函数，不需要任何参数，系统回自动弹出历史数据查询对话框，如图 17.27 所示。

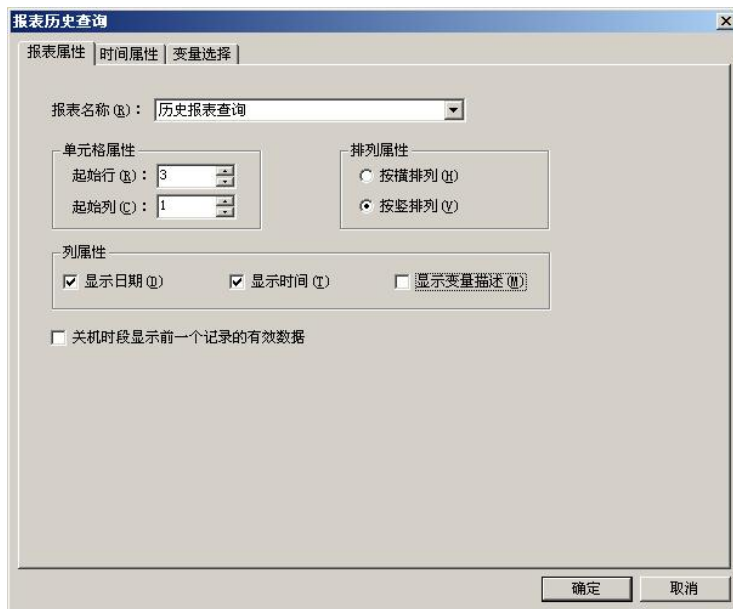


图 17.27 历史数据查询—报表属性页

该对话框共有三个属性页，用于定义历史数据查询的参数，分别介绍如下：

1、“报表属性”属性页：

报表名称：报表名称列表框中列出当前画面中所有报表的名称，用户通过单击列表框向下箭头后弹出的下拉表选择执行查询后的数据填充的报表名称。

第十七章 报表系统


单元格属性：选择查询后的数据在报表中填充开始的位置，输入起始行数、列数。

排列属性：确定数据在报表中的填充方向：横向填充、竖向填充。

列属性：有两个选项：“显示日期”、“显示时间”。当用户需要在查询数据的数据报表中同时显示数据被采集的日期和时间时，可以选择该项，或按实际需要任选一项。

显示变量描述：是否在变量名下显示变量描述。

显示关机时段数据：可以控制变量关联的设备通讯失败，变量的质量戳为坏，运行系统退出时期的数据在报表中的显示方式。如果不选中此项，在报表中显示“---”，如果在此项前打勾显示最后记录的数据。

 注意：

当查询的数据行数大于报表设计的行数时，系统将自动将添加行数，满足数据填充的需要，最大行数可到 20000 行。

2、“时间属性”属性页：

设置数据查询的时间属性。如图 17.28 所示。

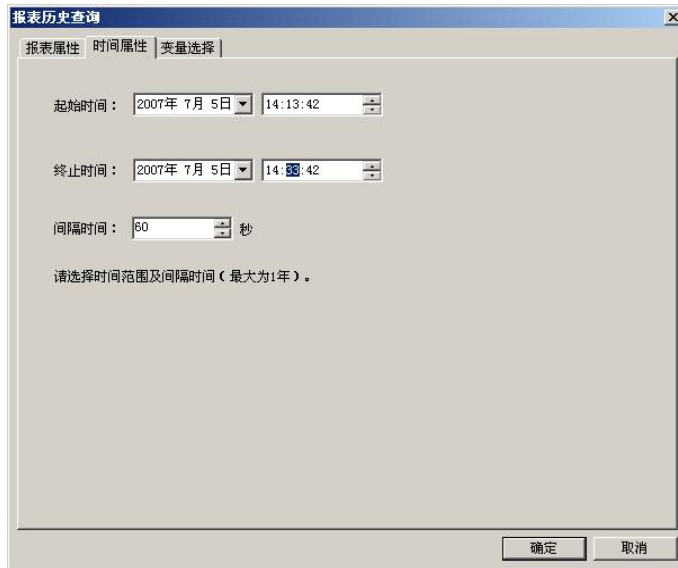


图 17.28 历史数据查询—时间属性页

起始时间: 定义所查询的历史数据的起始点时间, 包括起始日期和起始时间。修改起始日期的方法有两种:

- 用鼠标左键单击日期编辑框中需要修改的部分, 该部分被加亮选中, 然后直接使用键盘输入日期数值。如选择图 17.28 中的起始日期为 5 月 24 日, 先用鼠标单击如图 17.28 中起始时间编辑框“2001 年 6 月 25 日”中的“6”, 该字被加亮显示, 然后直接输入“5”; 用鼠标单击“25”, 该字被加亮显示, 然后直接输入“24”。
- 单击日期编辑框右侧向下的箭头, 弹出日历控件, 如图 17.29 所示。单击日历中顶部向左或向右的箭头按钮, 选择月份, 然后直接用鼠标单击选择某一天即可。

起始时间中时间的选择方法也有两种:

第十七章 报表系统

- 方法一同日期选择的方法 1。
- 用鼠标左键单击时间编辑框中需要修改的部分，该部分被加亮选中，然后用鼠标单击编辑框右侧的增加、减少的箭头，修改时间。

终止时间：定义所查询的历史数据的截止点时间，包括终止日期和终止时间。定义方法同“起始时间”。

时间间隔：定义查询历史数据时，查询的数据点间的时间间隔。可直接输入，或通过增加、减少按钮修改。



图 17.29 使用日历选择日期

3、“变量属性”属性页：

定义所要查询的变量。如图 17.30 所示。

第十七章 报表系统

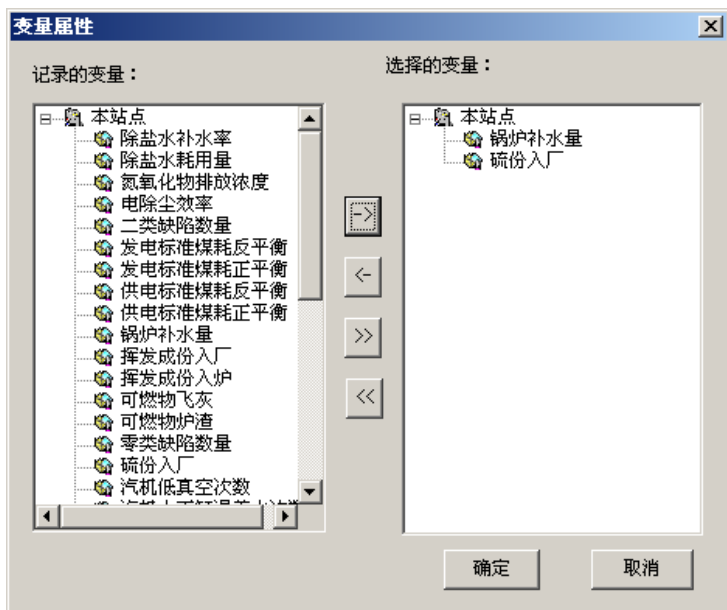


图 17.31 历史库数据查询—变量选择界面

“记录的变量”：该列表框中列出了当前工程中所有定义了数据历史记录了的变量。变量显示支持变量组方式，如上图 17.31 所示。

“选择的变量”：该列表框将显示用户选择的需要进行历史数据查询的变量。

变量选择箭头：

- 一>：在“记录的变量”列表框中选择一个变量，单击此按钮，将选择的变量加入到右侧的“选择的变量”列表框中。
- 一>：在“选择的变量”列表框中选择一个已经选择的变量，单击此按钮，将被选择的变量放回到左侧的“记录的变量”列表框中。
- >>：将左侧“记录的变量”列表框中所有的项添加到右侧的“选择的变量”



图 17.33 工业库选择

点击“选择变量”按钮，在弹出的对话框中选择变量，如图 17.34 所示。

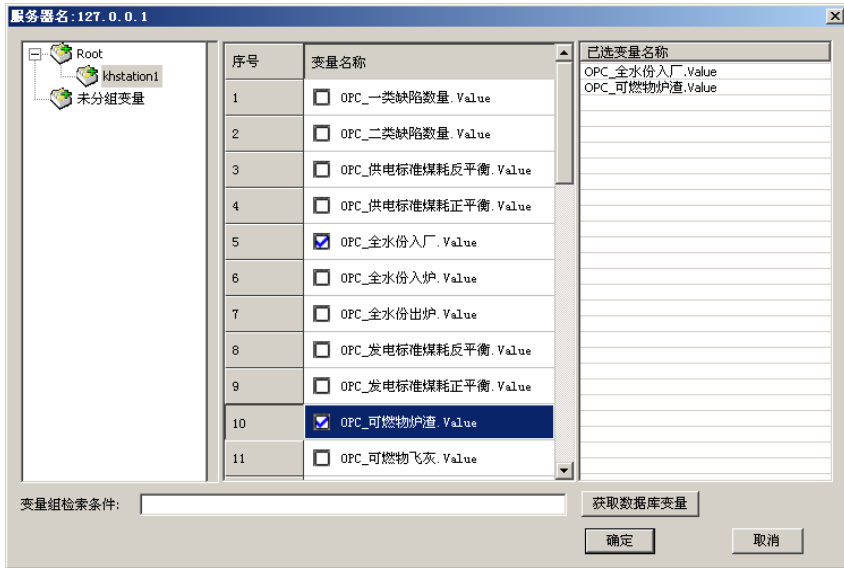


图 17.34 工业库数据查询—变量选择界面

确定后，在该工业库下面列出所选变量，如图 17.35 所示。



图17.35 工业库数据变量选择结果

点击“确定”按钮，变量列入**变量选择**属性页，如图 17.36 所示。

删除变量：在对话框中选中已存在的变量，点击“删除变量”按钮，可以删除该变量。

若选中某工业库服务器，点击“删除变量”，则可一次删除该服务器中的所有选择的变量。

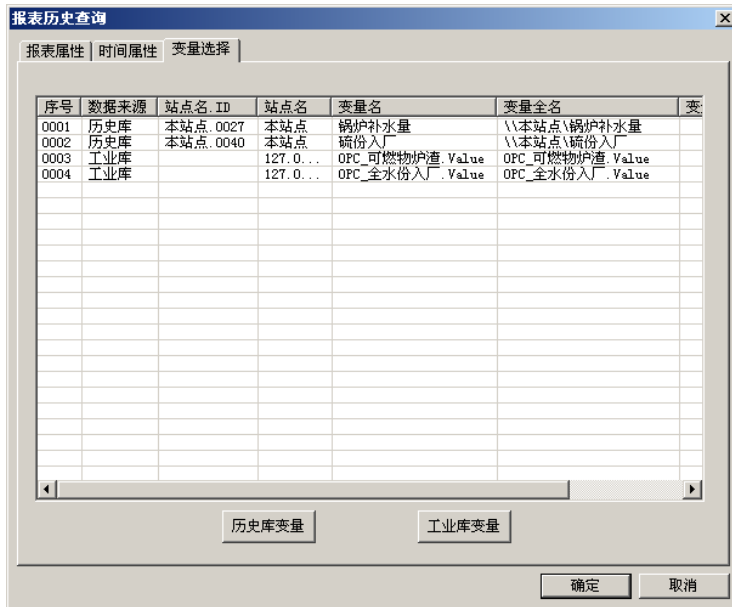


图 17.36 变量选择结果

设置完成后，单击“确定”按钮，所查询的数据便填充到指定的报表中，如图 17.37 所示。

历史报表查询						
日期	时间	锅炉补水量	硫份入厂	OPC_可燃物炉渣.Value	OPC_全水份入厂.Value	
07/07/05	14:13:42	357.25	458.00	270.00		382.00
07/07/05	14:14:42	307.00	408.00	203.00		304.00
07/07/05	14:15:42	358.00	459.00	237.00		327.00
07/07/05	14:16:42	307.75	408.00	271.00		349.00
07/07/05	14:18:42	308.50	409.00	238.00		394.00
07/07/05	14:19:42	359.50	460.00	272.00		316.00
07/07/05	14:20:42	309.25	410.00	205.00		339.00
07/07/05	14:21:42	360.25	461.00	238.00		361.00
07/07/05	14:22:42	310.00	411.00	272.00		383.00
07/07/05	14:23:42	361.00	462.00	206.00		306.00
07/07/05	14:24:42	310.75	412.00	239.00		328.00
07/07/05	14:25:42	361.75	463.00	273.00		351.00
07/07/05	14:26:42	311.50	412.00	206.00		373.00
07/07/05	14:27:42	362.50	463.00	240.00		396.00
07/07/05	14:28:42	312.25	413.00	273.00		318.00
07/07/05	14:29:42	325.50	425.00	217.00		311.00

图 17.37 历史报表查询结果

第十七章 报表系统

举例：在查询历史数据的按钮命令语言中使用函数：

```
ReportSetHistData2( )
```

在运行系统中弹出的图 17. 27 中的对话框中单元格属性的起始行和起始列都为 1。

如果将函数定义为：

```
ReportSetHistData2(2, 1);
```

则对话框中的默认起始行的值为 2，默认起始列为 1。

◇ ReportSetHistData3()

```
ReportSetHistData3("ReportName", "TagName", StartTime, SepTime, "szContent", bShow  
InvalidData);
```

函数功能：可以控制变量关联的设备通讯失败，变量的质量戳为坏，运行系统退出时期的数据在报表中的显示方式。

函数参数：ReportName：要填写查询数据结果的报表名称

szTagName：所要查询的变量名称

StartTime：数据查询的开始时间，该时间是通过组态王 HTConvertTime 函数转换的以 1970 年 1 月 1 日 8：00：00 为基准的长整型数，所以用户在使用本函数查询历史数据之前，应先将查询起始时间转换为长整型数值。

SepTime：查询的数据的时间间隔，单位为秒

szContent：查询结果填充的单元格区域

bShowInvalidData：0：在报表中不显示变量通讯失败或者关机时段的数据，显示“---”；1：在关机或通讯失败时段显示该时段之前

的最近一个有效数据的值。

◇ **ReportSetHistDataEx()**

```
ReportSetHistDataEx("ReportName", QueryFrom, "ServerName", "TagName", StartTime,  
SepTime, "szContent", bShowInvalidData);
```

函数功能：按照用户给定的参数从组态王历史库或工业库查询历史数据，**兼容 ReportSetHistData、ReportSetHistData3 的功能。**

函数参数：ReportName：要填写查询数据结果的报表名称

QueryFrom：当 QueryFrom== 1 时，从历史库中查询；当 QueryFrom== 2 时，从工业库中查询。

ServerName：可以为工业库的站点名，也可以是历史库查询时的远程站点名或“本站点”。从历史库查询时，如果 TagName 是全名的形式，即：站点名\变量名，此项可省略。

TagName：为工业库或历史库中的变量名

StartTime：数据查询的开始时间，该时间是通过组态王 HTConvertTime 函数转换的以 1970 年 1 月 1 日 8: 00: 00 为基准的长整型数，所以用户在使用本函数查询历史数据之前，应先将查询起始时间转换为长整型数值

SepTime：查询的数据的时间间隔，单位为秒

szContent：查询结果填充的单元格区域

bShowInvalidData：0：在报表中不显示变量通讯失败或者关机时段数据，1：

第十七章 报表系统

在报表中显示最后记录的数据。

举例：在报表中插入的时间为自 2007 年 7 月 3 日 13: 27: 00 以来的时间段，时间段间隔为 2 秒，数据报表填充的区域为” a2:a50” 数据来自本站点历史库；数据报表填充的区域为” c2:c50” 数据来自本机工业库：

```
long StartTime;

StartTime=HTConvertTime(2007, 7, 3, 13, 27, 0);

ReportSetHistDataEx("Report1", 1, "本站点", "nI0Dec50", StartTime, 2,
"a2:a50", 0);

ReportSetHistDataEx("Report1", 1, " ", "\\本站点\nI0Dec40", StartTime, 2,
"b2:b50", 0);

ReportSetHistDataEx("Report1", 2, "127.0.0.1", "ZFYLYP_nI0Dec50.Value",
StartTime, 20, "c2:c50", 0);
```

◇ ReportSetHistData4

```
ReportSetHistData4("ReportName", "TagName", StartTime, EndTime, StartRow, StartCol
, bTimeVisible);
```

函数功能：从历史库中，查询某变量 (TagName) 一段时间范围内 (StartTime-- EndTime) 的所有值，在报表 (ReportName) 中显示出来；如果 bTimeVisible=1，从 (StartRow, StartCol) 开始显示 2 列，左列显示时间，右列显示值；如果 bTimeVisible=0，从 (StartRow, StartCol) 开始显示 1 列，只显示数值。

参数:

ReportName --string 报表名称

TagName --string 变量名称

StartTime --long 数据查询的开始时间

EndTime --long 数据查询的结束时间

该时间是通过组态王 HTConvertTime 函数转换的以 1970 年 1 月 1 日 8: 00: 00(东八区)为基准的长整型数, 所以用户在使用本函数查询历史数据之前, 应先将查询起始时间转换为长整型数值

StartRow --long 查询的数据在报表中填充的起始行, 小于 20000。

StartCol --long 查询的数据在报表中填充的起始列, 小于 128。

bTimeVisible --bool 查询的数据在报表中是否显示时间列。0, 不显示时间列; 1, 显示时间列。

返回值: 0, 查询失败;

1, 查询成功;

2, 参数超限;

3, 报表名错误;

4, 变量名错误;

举例: 在报表中插入的时间为自 2013 年 6 月 3 日 10: 20: 00 到 2013 年 6 月 5 日 10: 20: 00 时间段内, 管道流量、管道压力的所有历史数据, 报表填充的区域为从第 1

第十七章 报表系统

列第 3 行开始，其中第一列显示时间，第二列显示‘管道流量’数据，第三列显示‘管道压力’数据。数据来自本站点历史库；

```
long StartTime;  
  
long EndTime;  
  
StartTime=HTConvertTime(2013, 6, 3, 10, 20, 0);  
  
EndTime =HTConvertTime(2013, 6, 5, 10, 20, 0);  
  
ReportSetHistData4("Report1", "管道流量", StartTime, EndTime,  
3, 1, 1);  
  
ReportSetHistData4("Report1", "管道压力", StartTime, EndTime,  
3, 3, 0);
```

17.3.6 报表打印类函数

1、报表打印函数

报表打印函数根据用户的需要有两种使用方法，一种是执行函数时自动弹出“打印属性”对话框，供用户选择确定后，再打印；另外一种是在执行函数后，按照默认的设置直接输出打印，不弹出“打印属性”对话框，适用于报表的自动打印。报表打印函数原型为：

✧ ReportPrint2

```
ReportPrint2(String szRptName) 或者 ReportPrint2(String szRptName,  
EV_LONG|EV_ANALOG|EV_DISC) ;
```

函数功能：将指定的报表输出到打印配置中指定的打印机上打印。

参数说明：szRptName：要打印的报表名称

EV_LONG|EV_ANALOG|EV_DISC：整型或实型或离散型的一个参数，当该参数不为0时，自动打印，不弹出“打印属性”对话框。如果该参数为0，则弹出“打印属性”对话框。

举例：自动打印“实时数据报表”：ReportPrint2(“实时数据报表”)； 或
ReportPrint2(“实时数据报表”，1)；手动打印时，弹出“打印属性”对话框：
ReportPrint2(“实时数据报表”，0)；

2、报表页面设置函数

开发系统中可以通过报表工具箱对报表进行页面设置，运行系统中则需要通过调用页面设置函数来对报表进行设置。页面设置函数的原型为：

◇ ReportPageSetup

ReportPageSetup(ReportName)；

函数功能：设置报表页面属性，如纸张大小，打印方向、页眉页脚设置等。执行该函数后，会弹出“页面设置”对话框，如图 17.9 所示。

参数说明：szRptName：要打印的报表名称

举例：对“实时数据报表”进行页面设置：ReportPageSetup(“实时数据报表”)；

3、报表打印预览函数

运行中当页面设置好以后，可以使用打印预览查看打印后的效果。打印预览函数原型如下：

第十七章 报表系统

◇ ReportPrintSetup

ReportPrintSetup(ReportName);

函数功能：对指定的报表进行打印预览。

参数说明：szRptName：要打印的报表名称

举例：对“实时数据报表”进行打印预览：

ReportPrintSetup (“实时数据报表”); 执行打印预览时，系统会自动隐藏组态王的开发系统和运行系统窗口结束预览后恢复。

17.3.7 报表向导函数

报表向导函数用于从工业库或组态王的历史库中获得某个时刻的历史数据、某段时间的统计数据。还可以重新设置报表查询的起始时间（运行状态下）。

◇ GetHistData

GetHistData("variable name", year,mon,day,hour,min,sec);

函数功能：取得某个时间点上存储于历史库的历史数据，返回值类型为字符串

参数说明：const char* str variable name:变量名称，格式为“\\站点名\变量名”，比如“\\本站点\RAD”；short year:年、short mon:月、short day:日、short hour:时、short min:分、short sec:秒。

举例：GetHistData (“\\本站点\RAD” ,2010,11,1, 12,00,00) 获取本站点上的变量 RAD 在 2010 年 11 月 1 日 12:00:00 时刻的数据。（如果当前时刻没有记录，则自动往前时刻寻找最近的数据；如果之前一直没有数据或者找到的数据质量戳不为 GOOD，

则返回“---”。

◇ GetStatisData

GetStatisData("variable name", "StartTime", timespan, statistic type, "EndTime")

函数功能：获得某时间段的统计数据，可以在用户脚本中使用。

参数说明： variable name: 变量名称，格式为“\\站点名\变量名”，比如[\\本站点\RAD](#)

StartTime: 查历史库数据的开始时间，格式为“年-月-日-时-分-秒”，如“2010-6-23-12-00-00”为2010年6月23日12时00分00秒。输入其他格式提示错误。

short timespan: 取值时间长度,单位为秒。

short StatisticType: 统计类型。0-最小值、1-最大值、2-平均值。输入其它值提示错误。

EndTime: 查历史库数据的结束时间，格式为“年-月-日-时-分-秒”，如“2010-6-23-23-00-00”为2010年6月23日23时00分00秒。

举例：GetStatisData(“\\本站点\RAD”，“2010-11-1-12-00-00”，10, 2, “2010-11-1-12-5-0”),
表示2010-11-1 12:0:0至2010-11-1 12:5:0时间长度为5分钟每隔10秒取一个本站点变量RAD的瞬时值，再用这些值求平均值。

注意：

如果开始与结束时间间隔过大，取值时间长度过小，导致需要查询的数据量过大，程序等待结果的时间将变长，无法进行其他操作。请选取好开始时间，结束时

第十七章 报表系统

间与取值时间长度。

◇ GetRDBData

GetRDBData("variable name", year,mon,day,hour,min,sec);

函数功能：取得某个时间点上存储于工业库的历史数据。

参数说明：const char* str variable name:变量名称，格式为“\\服务器名\变量名”，比如

\\127.0.0.1\aa_RAD_ss; short year:年、short mon:月、short day:日、short hour:时、
short min:分、short sec:秒。

举例：GetRDBData (“\\127.0.0.1\aa_RAD_ss”,2010,11,1, 12,00,00) ;

获取服务器名为 127.0.0.1 上的变量 aa_RAD_ss 在 2010 年 11 月 1 日 12:00:00 时刻的数据。（如果当前时刻没有记录，则自动往前时刻寻找最近的数据；如果之前一直没有数据或者找到的数据质量戳不为 GOOD，则返回“---”）。

◇ GetRDBStatisData

GetRDBStatisData("variable name", "StartTime", timespan, statistictype, "EndTime")

函数功能：用来获得存储在工业库中某时间段的统计数据，可以在用户脚本中使用。

返回值：类型为字符串。

参数说明：variable name: 变量名称，格式为“\\服务器名\变量名”，比如：

“\\127.0.0.1\aa_RAD_ss”

StartTime: 查工业库数据的开始时间，格式为“年-月-日-时-分-秒”，如：

“2010-6-23-12-00-00”为 2010 年 6 月 23 日 12 时 00 分 00


秒。输入其他格式提示错误。

short Timespan: 取值时间长度, 单位为秒。

short Statistype: 统计类型。0-最小值、1-最大值、2-平均值。输入其它值提示错误。

EndTime: 查工业库数据的结束时间, 格式为“年-月-日-时-分-秒”, 如: “2010-6-23-23-00-00”为2010年6月23日23时00分00秒。

举例: GetRDBStatisData (“\\127.0.0.1\aa_RAD_ss”, “2010-11-1-12-00-00”, 10, 2, “2010-11-1-12-5-0”), 表示2010-11-1 12:0:0至2010-11-1 12:5:0时间长度为5分钟每隔10秒取一个127.0.0.1上变量aa_RAD_ss的瞬时值, 再用这些值求平均值。

 注意: 如果开始与结束时间间隔过大, 取值时间长度过小, 导致需要查询的数据量过大, 程序等待结果的时间将变长, 无法进行其他操作。请选取好开始时间, 结束时间与取值时间长度。

◇ ReportSetStartTime

ReportSetStartTime(const char *pReprotName)

函数功能: 用于在使用了报表向导功能时, 在运行系统中重新设置报表查询的起始时间, 可以在按钮弹起的脚本中使用。运行后将弹出设置报表起始时间的对话框。

参数说明: ReportName: 报表名称。

◇ ReportRelationDBGuide

第十七章 报表系统

ReportRelationDBGuide("ReportName")

函数功能：调取报表连接向导

参数说明：ReportName：报表名称。

◇ ReportRelationDBRefresh

ReportRelationDBRefresh("ReportName");

函数功能：更新关系库报表连接数据。

参数说明：ReportName：报表名称。

17.3.8 报表行、列设置和锁定函数

◇ ReportSetRowColResize

ReportSetRowColResize("ReportName", bRow, bCol)

函数功能：用来设置报表的行和列是否可以调整大小，设置成功返回 1，失败返回 0。

参数说明："ReportName"：报表名称

bool bRow：行设定，0 为不能调整，1 为可以调整

bool bCol：列设定，0 为不能调整，1 为可以调整

◇ ReportSetLock

ReportSetLock (const CHAR* ReportName, BOOL is_enable, int nRow, int nCol)

函数功能：用于锁定报表的前 nRow 行和 nCol 列。

函数参数说明：

ReportName：报表名称。

is_enable: 是否锁定行列, 为 TRUE 执行锁定操作; 为 Fales 时, 执行取消锁定操作。

nRow: 锁定的行数, nCol: 锁定的列数。

例: ReportSetLock (“值班表”, 1,1,2): 对值班表执行锁定操作, 锁定第 1 行和前 2 列。



注意:

1. 锁定操作只能在报表没有锁定时执行, 如果报表已锁定, 必须先取消锁定, 才能执行锁定。

2. 锁定操作必须在报表滚动条没有滚动过的情况下执行, 否则无效。

17.4 如何套用报表模板

一般情况下, 工程中同一行业的报表基本相同或类似。如果工程人员在每做一个工程时, 都需要重新制作一个报表, 而其中大部分的工作是重复性的, 无疑是增大了工作量和开发周期, 特别是比较复杂的报表。而利用已有的报表模板, 在其基础上做一些简单的修改, 将是一个很好的途径, 使工作快速、高效的完成。

组态王在开发和运行系统中都提供了报表的保存功能, 即将设计好的报表或保存有数据的报表保存为一个模板文件 (扩展名为.rtl), 工程人员需要相似的报表时, 只需先建立一个报表窗口, 然后在报表工具箱中直接打开该文件, 则原保存的报表便被加载到了工程里来。如果不满意, 还可以直接修改或换一个报表模板文件加载。

套用报表模板时, 有两种方式, 第一种是使用报表工具箱上的“打开”按钮, 如图 17.38 所示, 系统会弹出文件选择对话框, 在其中选择已有的模板文件 (*.rtl), 打开后,

第十七章 报表系统

当前报表窗口便自动套用了选择的模板格式。具体参见 17.2.2 节。



图 17.38 使用报表工具箱套用模板

第二种方法是使用“报表设计”中的“表格样式”，首先建立一些常用的格式，然后在使用时，直接选择表格样式即可自动套用模板。具体操作参见 17.1.2 节。

17.5 如何制作实时数据报表

实时数据报表主要是来显示系统实时数据。除了在表格中实时显示变量的值外，报表还可以按照单元格中设置的函数、公式等实时刷新单元格中的数据。在单元格中显示变量的实时数据一般有两种方法。

17.5.1 单元格中直接引用变量


在报表的单元格中直接输入“=变量名”，既可在运行时在该单元格中显示该变量的数值，当变量的数据发生变化时，单元格中显示的数值也会被实时刷新。如图 17.39 所示，例如在单元格“B4”中要实时显示当前的登录“用户名”，在“B4”单元格中直接输入“=\本站点\\$用户名”，切换到运行系统后，该单元格中便会实时显示登录的用户的名称，如“系统管理员”登录，则会显示“系统管理员”。

	A	B	C
1			
2			
3			
4	用户名	=\\本站点\\\$用户名	
5			

用户名	系统管理员		

图 17.39 直接引用变量

这种方式适用于表格单元格中的显示固定变量的数据。如果单元格中要显示不同变量的数据或值的类型不固定，则最好选择单元格设置函数。

 注意：

只有当报表画面被打开时其中的数据才会被刷新。

17.5.2 使用单元格设置函数

如果单元格中显示的数据来自于不同的变量，或值的类型不固定时，最好使用单元格设置函数。当然，显示同一个变量的值也可以使用这种方法。单元格设置函数有：ReportSetCellValue（）、ReportSetCellString（）、ReportSetCellValue2（）、ReportSetCellString2（），这些函数的使用方法请参见本章 17.3.2 节，或组态王函数手册。如 17.5.1 节中在“B4”中设置用户名，也可以在数据改变命令语言中使用 ReportSetCellString（）函数设置数据，如图 17.40 所示。这样当系统运行时，用户登录后，用户名就会被自动填充指定单元格中。

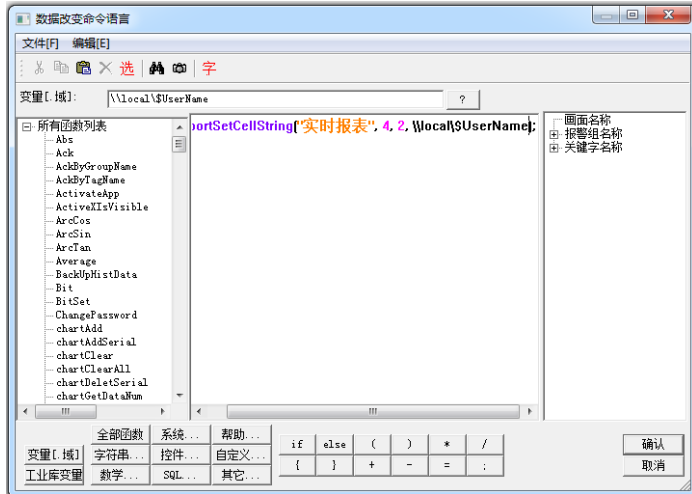


图 17.40 使用单元格设置函数

17.6 如何制作历史数据报表

历史报表记录了以往的生产记录数据，对用户来说是非常重要的。历史报表的制作根据所需数据的不同有不同的制作方法，这里介绍两种常用的方法。

17.6.1 向报表单元格中实时添加数据

例如要设计一个锅炉功耗记录表，该报表为 8 小时生成一个（类似于班报），要记录每小时最后一刻的数据作为历史数据，而且该报表在查看时应该实时刷新。

对于这个报表就可以采用向单元格中定时刷新数据的方法实现。报表设计如图 17.41 所示，按照规定的时间，在不同的小时里，将变量的值定时用单元格设置函数如 ReportSetCellValue() 设置到不同的单元格中，这时，报表单元格中的数据会自动刷新，

而带有函数的单元格也会自动计算结果，当到换班时，保存当前添有数据的报表为报表文件，清除上班填充的数据，继续填充，这样就完成了要求。这样就好比是操作员每小时在记录表上记录一次现场数据，当换班时，由下一班在新的记录表上开始记录一样。

可以另外创建一个报表窗口，在运行时，调用这些保存的报表，查看以前的记录，实现历史数据报表的查询。

	A	B	C	D	E	F	G	H
1	系统锅炉房功耗总报表							
2	NO!1		报表时间:	=Time(\$...				
3	日期	时间	1#热水锅炉	1#采暖锅炉	泵	总功耗	供电单价(元)	总电价(元)
4						=sum('c4:j4')		=k4'*g4'
5						=sum('c5:j5')		=k5'*g5'
6						=sum('c6:j6')		=k6'*g6'
7						=sum('c7:j7')		=k7'*g7'
8						=sum('c8:j8')		=k8'*g8'
9						=sum('c9:j9')		=k9'*g9'
10						=sum('c10:j10')		=k10'*g10'
11						=sum('c11:j11')		=k11'*g11'
12						=sum('f4:f11')		=sum('h4:h11')
13	制表单位:					值班员:		

图 17.41 锅炉功耗报表

这种制作报表的方式既可以作为实时报表观察实时数据，也可以作为历史报表保存。用户可以参照组态王演示工程中的实时报表。

17.6.2 使用历史数据查询函数

使用历史数据查询函数从组态王历史库或工业库中按指定的起始时间和时间间隔查询指定变量的数据，历史数据查询函数的使用方法请参见本章 17.3.5 节，或组态王函数手册。

第十七章 报表系统

如果用户在查询时，希望弹出一个对话框，可以在对话框上随机选择不同的变量和时间段来查询数据，可使用函数 `ReportSetHistData2(StartRow, StartCol)`。该函数已经提供了方便、全面的对话框供用户操作。但该函数会将指定时间段内查询到的所有数据都填充到报表中来，如果报表不够大，则系统会自动增加报表行数或列数，对于使用固定格式报表的用户来说不太方便。那么可以用下面一种方法。

如果用户想要一个定时自动查询历史数据的报表，而不是弹出对话框，或者历史报表的格式是固定的，要求将查询到的数据添到固定的表格中，多余查询的数据不需要添到表中，这时可以使用函数 `ReportSetHistData`，`ReportSetHistData3` 或 `ReportSetHistDataEx`。使用这些函数时，用户需要指定查询的起始时间，查询间隔，和变量数据的填充范围。

组态王报表拥有丰富而灵活的报表函数，用户可以使用报表制作一些数据存储、求和、运算、转换等特殊用法。如将采集到的数据存储于报表的单元格中，然后将报表数据赋给曲线控件来制作一段分析曲线等，既可以节省变量，简化操作，还可重复使用。总之，报表的其它用法还很多，有待用户按照自己的实际用途灵活使用。

17.7 如何使用报表向导

上面两节讲述了如何制作实时、历史数据报表。下面将介绍如何使用报表向导以组态王的历史库或 `KingHistorian` 为数据源来快速创建所需的班报表、日报表、周报表、月报表、季报表和年报表。此外，还可以实现值的行列统计功能。

17.7.1 制作各种报表


创建一个报表窗口（详情请参考“17.1.1 如何创建报表窗口”），选中新建的报表窗口，弹出报表工具箱对话框。单击报表工具箱的  按钮，弹出如图 17.42 所示的对话框。



图 17.42 数据库类型选择

17.7.1.1 制作亚控数据库报表

在上面的对话框中，可以选择亚控数据库或关系数据库。选择亚控数据库，点击下一步，弹出如下数据源选择对话框：



图 17.43 从工业库添加变量

选中将要添加变量的工业库，这里是 127.0.0.1。单击“添加变量”按钮，弹出如图 17.44 所示的对话框，该窗口列举了被选的工业库服务器中的所有变量。



图 17.44 工业库变量

第十七章 报表系统

选择要添加的变量（即在变量前的复选框打√），单击“确定”按钮返回图 17.43 的界面。在图 17.43 所示的界面中，选择可使用的工业库再次添加变量或单击“确定”按钮返回图 17.42 所示的对话框，则选中的变量将会在该对话框的列表中显示。

历史库变量的添加：

单击图 17.42 的“添加历史库变量”按钮，弹出“从历史库添加变量”对话框，如图 17.45。



图 17.45 从历史库添加变量

在记录的变量列表中选择要添加的历史库变量，单击形如“->”的按钮将其添加到“选择的变量”列表中，或单击形如“>>”按钮添加所有变量。选择完成后，单击“确定”按钮，添加的变量将会在图 17.42 的列表中显示。

选择好变量后，单击图 17.42 的“下一步”按钮，将会弹出下图所示的对话框。在该对话框中，设置报表的基本属性。

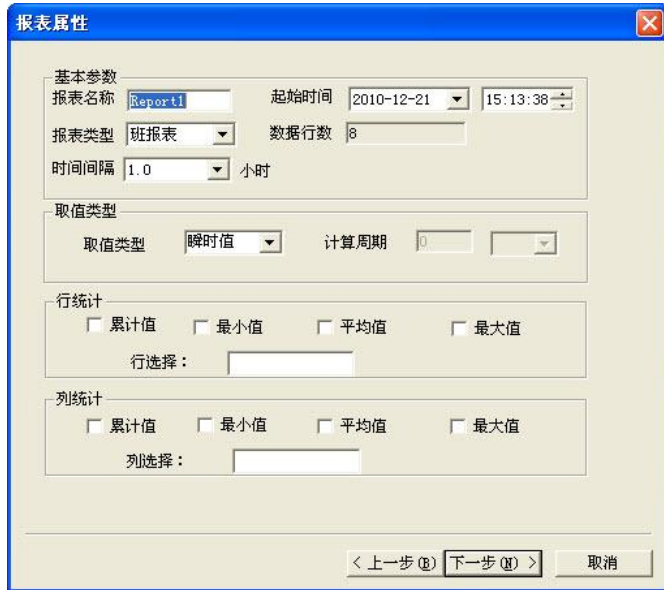


图 17.46 报表参数设置页

- ◇ **报表名称：**报表的唯一标识，不能与已存在的报表重名。
- ◇ **报表类型：**包括班报、日报、周报、月报、季报、年报和自由报。
- ◇ **报表的列数和行数：**所有报表的列数都根据上一步所选择的变量数目固定。
自由报表可以自由设置行数，并根据设置的起始时间和时间间隔，确定自由报表的数据统计范围。其他报表根据统计时间和时间间隔确定数据行数。其中，日报表统计 24 小时的数据，班报表统计 8 个小时的数据。
举例：若选择班报表，时间间隔设置为 1.0 小时，则行数为：班报表统计时间/时间间隔=8 小时/1.0 小时=8，因此行数为 8 行。
- ◇ **起始时间：**开始数据统计的起始时间。

第十七章 报表系统

- ◇ **时间间隔：**报表统计的时间间隔。根据所选择的报表类型的不同，时间间隔的选择也不同。周报表、月报表、季报表和年报表时间间隔一定，所以当选择这几种报表时，时间间隔选项不可用。班报表和日报表以小时为单位，共有 0.5 小时，1 小时，2 小时三种时间间隔可以选择。自由报表比较灵活，可以选择不同的单位，写入不同的数字，来组成时间间隔。
- ◇ **取值类型：**有瞬时值、最大值、最小值和平均值四个选项，默认为瞬时值。当选择为最大值，最小值或平均值时，会根据设置的计算周期，在每个时间间隔内，当到达计算周期时，取数据的瞬时值，进行最大，最小和平均值的计算，并输出。
- ◇ **计算周期：**当取值类型选为最大值、最小值和平均值时，才可用。
- ◇ **行统计和列统计：**选择是否需要行末、列末统计，并填写需要统计的行与列。这里行，列的选择格式是一定的：必须从小到大排列，非连续数目用“，”隔开，连续的数字用“-”连接开始和结束的数目。如：1,3-5,7-8。表示选择了第 1 行，第 3 到 5 行，第 7 到 8 行数据进行统计。

设置好报表的参数后，单击“下一步”按钮，弹出如图 17.47 所示的对话框。



图 17.47 报表的时间属性

- ◇ **时间显示格式选择：**提供“2010-6-8 15:21:00”，“2010/6/8 15:21:00”和“2010年6月8日 15:21:00”三种格式供选择。
- ◇ **故障数据显示：**该功能未实现，目前只能和报表其他函数一样，在故障数据处显示：“---”。
- ◇ **行高和列宽：**设置报表的行高和列宽，单位可以选择像素或者毫米，默认为像素。
- ◇ **标题起始行和起始列：**设置标题的起始行数和起始列数。
- ◇ **表头类型：**有固定表头和自由表头两个选项。若选择固定表头，则在报表内容较多的情况下，用滚动条翻页时，每屏数据的第一行都会存在表头；若选择

第十七章 报表系统

自由表头，只在首屏的第一行存在表头。

完成报表显示属性的设置后，单击“完成”按钮返回开发系统界面并显示生成的报表，如图 17.48 所示。

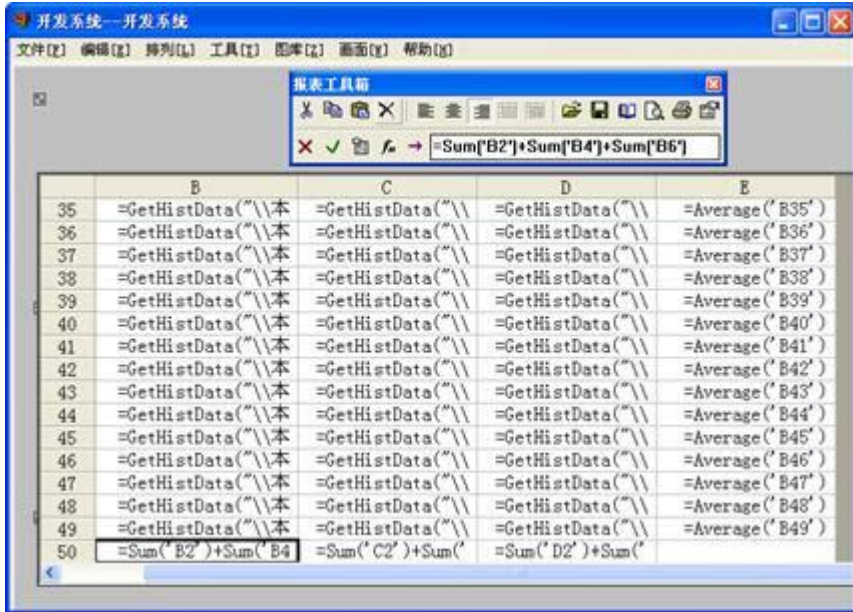


图 17.48 用报表向导生成的报表

17.1.1.2 制作关系数据库报表

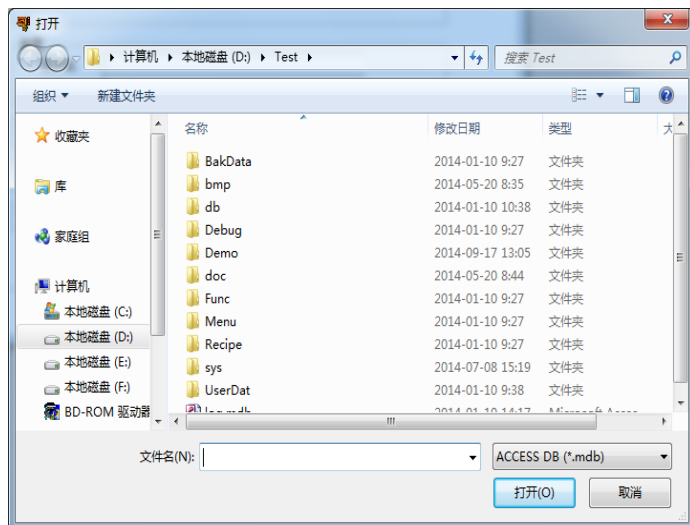
在上面的对话框中，可以选择亚控数据库或关系数据库。选择关系数据库，点击下一步，弹出如下数据库配置对话框：



数据库配置

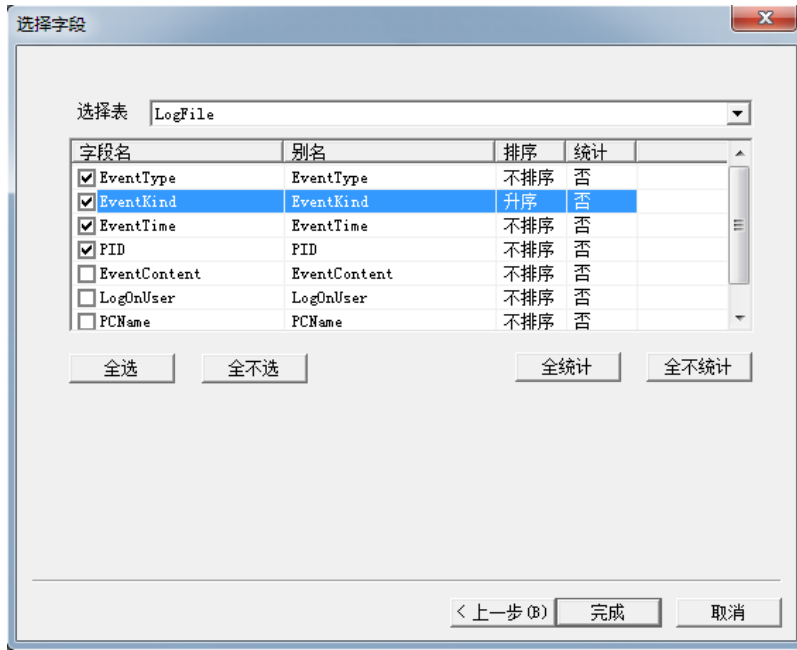
- ◇ **类型**：选择哪种类型的关系数据库，如：ACCESS、SqlServer、Oracle 等。
- ◇ **数据库**：在数据库存放目录下选择某数据库。具体方法是点击路径选择按钮，弹出如下图所示的对话框。通过该对话框选择某数据库。

第十七章 报表系统



- ◇ **用户名：**数据库的用户名。
- ◇ **密码：**数据库密码。
- ◇ **行统计和列统计：**选择是否需要行末统计，并填写需要统计的行。这里行的选择格式是一定的：必须从小到大排列，非连续数目用“，”隔开，连续的数字用“-”连接开始和结束的数目。如：1,3-5,7-8。表示选择了第1行，第3到5行，第7到8行数据进行统计。
- ◇ **统计条件：**输入要统计的条件，如 `cell() != 0 and cell() != 9999`。
- ◇ **其它：**根据需要设置是否显示字段名

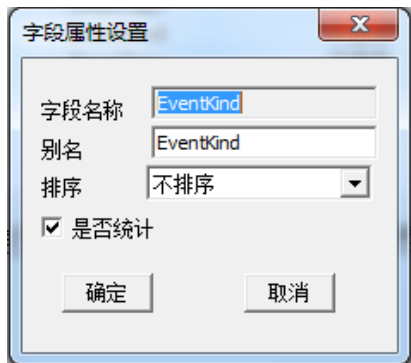
在完成上述配置后单击“下一步”，则弹出选择字段对话框，从数据表下拉框中选择需要查询的数据表，在字段名中勾选需要查询的字段。



选择字段

- ◇ **数据表：**在指定的数据库下选择某数据表
- ◇ **字段选择列表：**显示指定数据表下的字段及相关信息，包括：字段名、字段别名、是否排序、是否统计。双击某行，弹出字段属性设置框，如下图所示：

第十七章 报表系统



通过该设置框，可以对字段选择列表中的字段设置别名、是否排序、是否统计等信息。


- ◇ **全选按钮：**左键点击全选按钮，字段列表中的字段被全部选中。
- ◇ **全不选按钮：**左键点击全不选按钮，全部选中的字段取消选中。
- ◇ **全统计按钮：**左键点击全统计按钮，字段列表中的选中字段设置全统计。
- ◇ **全不统计按钮：**左键点击全不统计按钮，全统计的字段取消统计。

完成上述设置后，单击“完成”按钮返回开发系统界面并显示生成的报表，如下图所示：

	A	B	C	D	E
1	EventT	EventK	EventT	PID	
2	0	0	2014-0	7	
3	0	0	2014-0	8	
4	0	0	2014-0	7	
5	0	0	2014-0	2	
6	0	0	2014-0	7	
7	0	0	2014-0	1	
8	0	0	2014-0	7	
9	0	0	2014-0	7	
10	0	0	2014-0	1	
11	0	0	2014-0	7	
12	0	0	2014-0	7	
13	0	0	2014-0	1	
14	0	0	2014-0	7	
15	0	0	2014-0	7	
16	0	0	2014-0	2	
17	0	0	2014-0	2	
18	0	0	2014-0	2	

17.7.2 重设报表查询时间

在运行由报表向导生成的报表时，可以调用 `ReportSetStartTime(const char *pReprotName)`重新设置报表的查询起始时间，`const char* pReportName` 为报表名称。该函数可以在按钮弹起的脚本中使用，运行后将弹出设置报表起始时间的对话框，如图 17.49 所示

 注意：

该函数仅对用报表向导生成的报表起作用。



图 17.49 设置起始时间

17.8 报表锁定行列功能

报表的锁定行列功能对组态王运行系统中的报表适用。

17.8.1 锁定报表的行列

在组态王运行系统的报表中，选择要锁定行列交叉处的单元格，如图 17.50 所示。同时按下 {Ctrl}+{L} 键，可以锁定选定单元格上侧的所有行和左侧的所有列，如图 17.51 所示。被锁定区域边界以蓝色线条为界线。

第十七章 报表系统

锁定行列中的单元格不能被编辑。如果需要对锁定的单元格进行编辑，必须先对锁定部分进行解锁操作。

17.8.2 用函数锁定报表的行列

◇ ReportSetLock

ReportSetLock (const CHAR* ReportName, BOOL is_enable, int nRow, int nCol) 可以用于锁定报表的前 nRow 行和 nCol 列。

函数参数说明：

ReportName: 报表名称。

is_enable: 是否锁定行列，为 TRUE 时执行锁定操作；为 Fales 时，执行取消锁定操作。

nRow: 锁定的行数

nCol: 锁定的列数。

例: ReportSetLock (“值班表”, 1, 1, 2): 对值班表执行锁定操作，锁定第 1 行和前 2 列。




注意：

1. 锁定操作只能在报表没有锁定时执行，如果报表已锁定，必须先取消锁定，才能执行锁定。

2. 锁定操作必须在报表滚动条没有滚动过的情况下执行，否则无效。

17.8.3 解除锁定

当报表的某些行和列被锁定后，在表格中单击任意单元格，同时按下 {Ctrl} + {U} 键，可以解除锁定。


 注意：

无论锁定行列，还是解除锁定，在按下快捷键前，要用鼠标点击报表的单元格，使焦点位于报表上。否则可能出现操作无效的情况。

第十八章 组态王 7.0 与其它应用程序的动态数据交换 (DDE)

- ☞ 介绍 DDE 的概念
- ☞ 举例说明组态王 7.0 与 EXCEL、VB 服务程序之间的动态数据交换
- ☞ 介绍组态王 7.0 中的 I/O 变量如何进行 DDE 访问

“组态王 7.0”支持动态数据交换 (DDE: Dynamic Data Exchange)，能够和其他支持动态数据交换的应用程序方便地交换数据。通过 DDE，工程人员可以利用 PC 机丰富的软件资源来扩充“组态王 7.0”的功能，比如用电子表格程序从“组态王 7.0”的数据库中读取数据，对生产作业执行优化计算，然后“组态王 7.0”再从电子表格程序中读出结果来控制各个生产参数；可以利用 VISUAL BASIC 开发服务程序，完成数据采集、报表打印、多媒体声光报警等功能，从而很容易组成一个完备的上位机管理系统；还可以和数据库程序、人工智能程序、专家系统等进行通信。

 注意:

“组态王 7.0”和其他支持 DDE 应用程序的数据交换与此类似。

18.1 动态数据交换的概念

DDE (Dynamic Data Exchange: 动态数据交换) 是 WINDOWS 平台上的一个完整的通信协议, 它使支持动态数据交换的两个或多个应用程序能彼此交换数据和发送指令。DDE 始终发生在客户应用程序和服务器应用程序之间。DDE 过程可以比喻为两个人的对话, 一方问另一方提出问题, 然后等待回答。提问的一方称为“顾客”(Client), 回答的一方称为“服务器”(Server)。一个应用程序可以同时是“顾客”和“服务器”: 当它向其他程序中请求数据时, 它充当的是“顾客”; 若有其他程序需要它提供数据, 它又成了“服务器”。DDE 对话的内容是通过三个标识名来约定的:

- ◇ **应用程序名 (application):** 进行 DDE 对话的双方的名称。商业应用程序的名称在产品文档中给出。“组态王 7.0”运行系统的程序名是“VIEW”; Microsoft Excel 的应用程序名是“Excel”; Visual Basic 程序使用的是可执行文件的名称。
- ◇ **主题 (topic):** 被讨论的数据域 (domain)。对“组态王 7.0”来说, 主题规定为“tagname”; Excel 的主题名是电子表格的名称, 比如 sheet1、sheet2 ……; Visual Basic 程序的主题由窗体 (Form) 的 LinkTopic 属性值指定。
- ◇ **项目 (item):** 这是被讨论的特定数据对象。在“组态王 7.0”的数据词典里, 工程人员定义 I/O 变量的同时, 也定义项目名称(参见第五章 变量定义和管理)。Excel 里的项目是单元, 比如 r1c2 (r1c2 表示第一行、第二列的单元)。对 Visual Basic 程序而言, 项目是一个特定的文本框、标签或图片框的名称。

建立 DDE 之前, 客户程序必须填写服务器程序的三个标识名。为方便使用, 列表如下:

应用程序名		主题		项目	
规定	例子	规定	例子	规定	例子

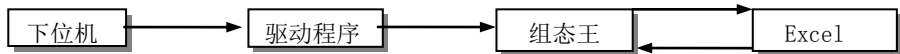
组态王	VIEW		tagname		工程人员 自己定义	温度
Excel	Excel		电子表格名	sheet1	单元	r2c2
VB	执行文 件名	vbdde	窗体的 LinkTopic属 性	Form1	控件的名 称	Text

18.2 组态王 7.0 与 Excel 间的数据交换

为了建立 DDE 连接，需要在“组态王 7.0”的数据词典里新建一个 I/O 变量，并且登记服务器程序的三个标识名。当 Excel 作为“顾客”向“组态王 7.0”请求数据时，要在 Excel 单元中输入远程引用公式：=VIEW|TAGNAME!设备名.寄存器名。

◇ 此设备名.寄存器名指的是“组态王 7.0”数据词典里 I/O 变量的设备名和该变量的寄存器名。设备名和寄存器名的大小写一定要正确。

18.2.1 组态王 7.0 访问 Excel 的数据



在本例中，假设“组态王 7.0”访问 Excel 的数据，“组态王 7.0”作为客户程序向 Excel 请求数据。数据流向如下图所示。

“组态王 7.0”作为客户程序，需要在定义 I/O 变量时设置服务器程序 Excel 的三个标识名，即：服务程序名设为 Excel，话题名设为电子表格名，项目名设置成 Excel 单元

格名。具体步骤如下。

1、在“组态王 7.0”中定义 DDE 设备

在工程浏览器中，从左边的工程目录显示区中选择“设备\DDE”，然后在右边的内容显示区中双击“新建”图标，则弹出“设备配置向导”（DDE 设备的配置请参见第六章 I/O 设备管理），已配置的 DDE 设备的信息总结列表框如图 18.1 所示。



图 18.1 利用设备安装向导定义 DDE 设备

定义的连接对象名为 EXCEL (也即连接设备名), 定义 I/O 变量时要使用此连接设备。

2、在“组态王 7.0”中定义变量

在工程浏览器左边的工程目录显示区中，选择“数据库\数据词典”，然后在右边的目录内容显示区中用左键双击“新建”图标，弹出“变量属性”对话框，在此对话框中建立一个 I/O 实型变量。如图 18.2 所示。变量名设为 FromExceltoView，项目名设

为 r2c1, 表明此变量将和 Excel 第二行第一列的单元进行连接。

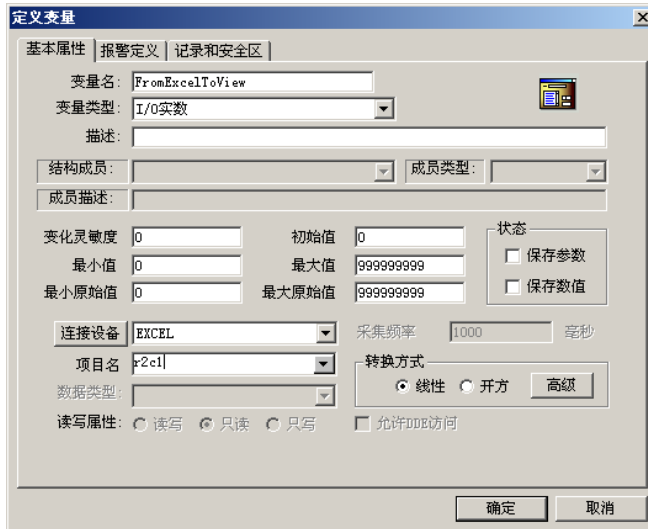


图 18.2 组态王 7.0 定义变量并与 Excel 进行连接

3、创建“组态王 7.0”画面

新建组态王 7.0 画面名为 test, 如图 18.3 所示。

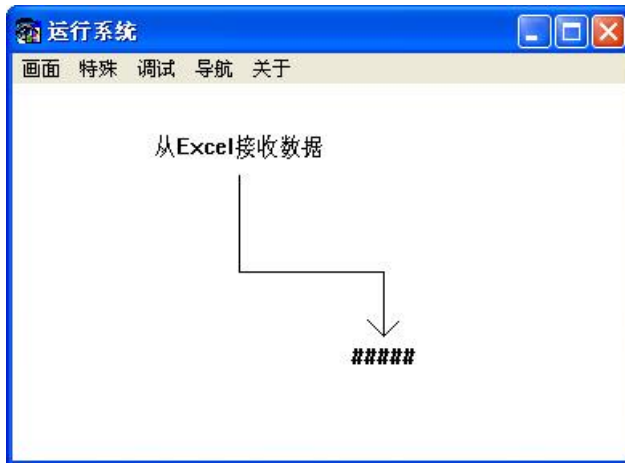


图 18.3 组态王 7.0 运行系统输出变量

为文本对象“#####”设置“模拟值输出”的动画连接，如图 18.4 所示。



图 18.4 对变量输出进行动画连接

设置完成后，选择菜单“文件\全部存”命令，保存画面。在工程浏览器中选择菜单“配置\运行系统”，弹出“运行系统配置”对话框，从对话框中选择主画面配置

卡片，将画面 test 设置为主画面。

4、启动应用程序

首先启动 Excel 程序，然后启动组态王 7.0 运行系统。TouchView 启动后，TouchView 就自动开始与 Excel 连接。在 Excel 的 A2 单元（第二行第一列）中输入数据，可以看到，TouchView 中的数据也同步变化。如图 18.5 所示。

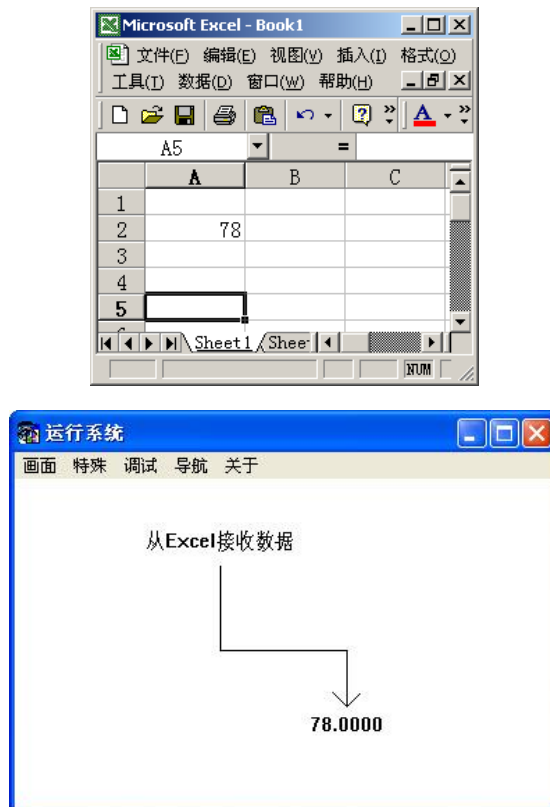
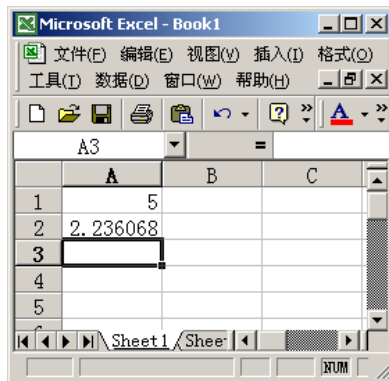


图 18.5 组态王 7.0 访问 Excel 交换数据

特别地，在 Excel 的 A1 单元中输入一个数值，A2 单元中输入公式=SQRT(A1) 公式的含义是：A2 单元的值是 A1 单元值的平方根。利用 Excel 的功能，默认情况下，每当任一数据发生变化时，公式的值都将重新计算。单元 A1 的值改变时，单元 A2 的值也将同时变化，并传给“组态王 7.0”中与之有连接的变量。如图 18.6 所示。在“组态王 7.0”中就可以得到这个变量的平方根，好像“组态王 7.0”在独立计算一样。用类似的方法，可以极大地扩充“组态王 7.0”的功能。（在“组态王 7.0”中也有求平方根函数，此处仅以 Excel 的函数为例）。



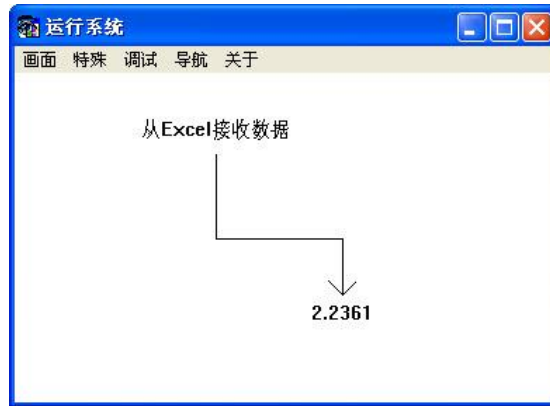
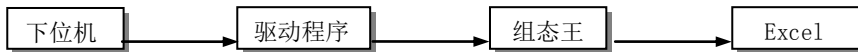


图 18.6 组态王 7.0 访问 Excel 交换数据

18.2.2 Excel 访问组态王 7.0 的数据

在本例中，假设“组态王 7.0”通过驱动程序从下位机采集数据，Excel 又向“组态王 7.0”请求数据。“组态王 7.0”既是驱动程序的“客户”，又充当了 Excel 的服务器，Excel 访问组态王 7.0 的数据。数据流向如下图所示。



具体步骤如下：

1、在“组态王 7.0”中定义设备

在工程浏览器中，从左边的工程目录显示区中选择“设备”，然后在右边的内容显示区中双击“新建”图标，则弹出“设备配置向导”（设备的配置可请参见第六章 I/O 设备管理，在这里比如建立了 OMRON 的 PLC），已配置的设备的信息总结列表框如图 18.7 所示。



图 18.7 设备安装向导

定义连接对象名为 OMRON（也就是连接设备名），定义 I/O 变量时要使用此连接设备。

2、在“组态王 7.0”中定义 I/O 变量

在工程浏览器左边的工程目录显示区中，选择“数据库\数据词典”，然后在右边的目录内容显示区中用左键双击“新建”图标，弹出“变量属性”对话框，在此对话框中建立一个 I/O 实型变量。如图 18.8 所示。变量名设为 FromViewToExcel，这个名称由工程人员自己定义。必须选择“允许 DDE 访问”选项。该选项用于组态王 7.0 能够从外部采集来的数据传送给 VB 或 EXCEL 或其它应用程序使用。该变量的项目名为“OMRON.AR001”。变量名在“组态王 7.0”中使用，项目名是供 Excel 引用的。连接设备为 OMRON，用来定义服务器程序的信息。

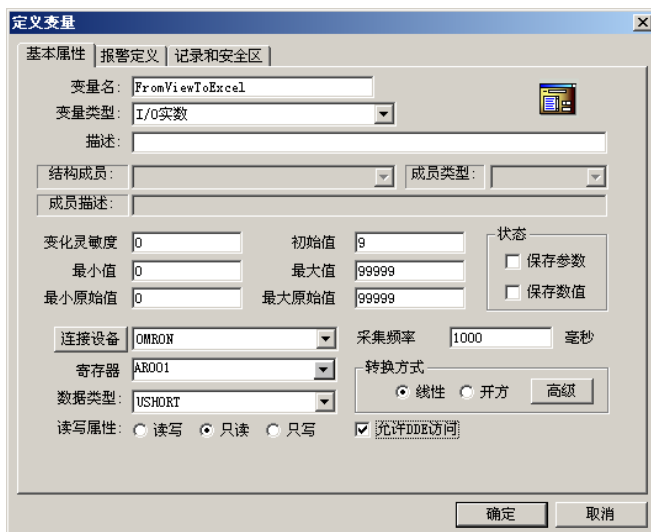


图18.8 定义IO变量



注意:

在定义变量时必须要选择“允许 DDE 访问”，否则在客户应用程序不能访问到组态王 7.0 的变量。

3、创建画面

在组态王 7.0 开发系统中建立画面 test1，如图 18.9 所示。

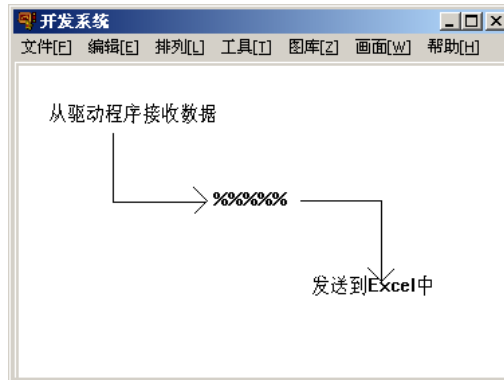


图 18.9 组态王 7.0 运行系统输出该变量

为文本对象“%%%"设置“模拟值输出”动画连接，如图 18.10 所示。

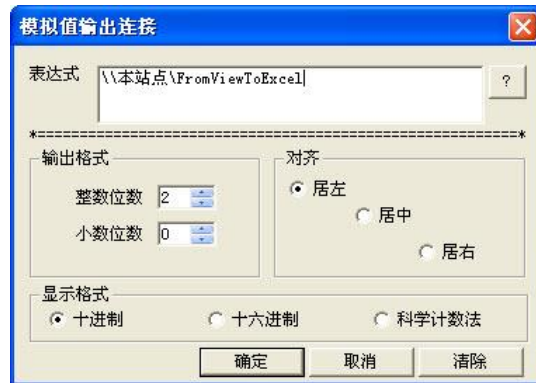


图 18.10 建立变量输出动画连接

选择菜单“文件\全部存”，保存画面。在工程浏览器中选择菜单“配置\运行系统”，弹出“运行系统配置”对话框，从对话框中选择主画面配置卡片，将画面 test1 设置为主画面。

4、启动应用程序

启动“组态王 7.0”画面运行系统 TouchView。TouchView 启动后，如果数据词典内定义的有 I/O 变量，TouchView 就自动开始连接。然后启动 Excel。如图 18.11 所示，选择 Excel 的任一单元，比如 r1c1，输入远程公式:=VIEW|tagname!OMRON.AR001

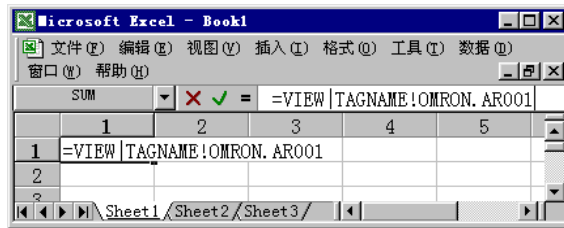


图 18.11 Excel 中引用组态王 7.0 变量

- VIEW和tagname分别是“组态王 7.0”运行系统的应用程序名和主题名。
- OMRON.AR001是“组态王 7.0”中的 I/O 变量 FromViewToExcel 的项目名。在 Excel 中只能引用项目名，不能直接使用“组态王 7.0”的变量名。输入完成后，Excel 进行连接。若连接成功，单元格中将显示数值。如图 18.12 所示。

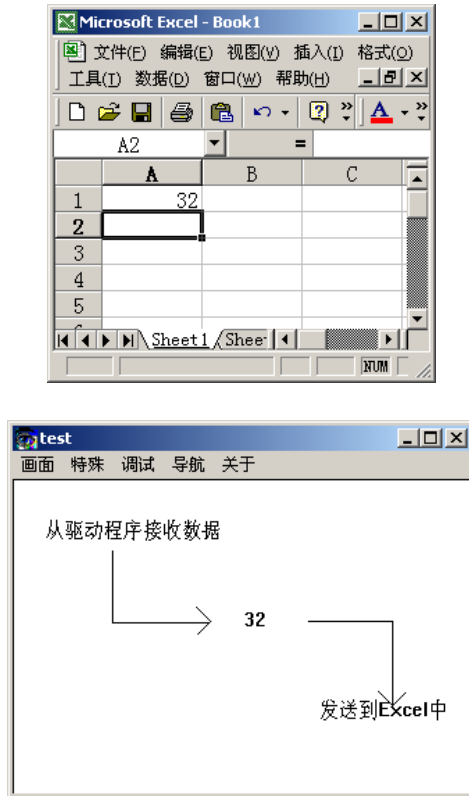


图 18.12 组态王 7.0 运行系统输出

18.3 组态王 7.0 与 VB 间的数据交换

在 Visual Basic 可视化编程工具中，DDE 连接是通过控件的属性和方法来实现的。对于作“顾客”的文本框、标签或图片框，要设置 LinkTopic、LinkItem、LinkMode 三个属性。

◇ control.LinkTopic=服务器程序名|主题名

◇ control.LinkItem=项目名

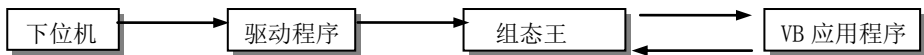
其中, control 是文本框、标签或图片框的名字。

◇ control.LinkMode 有四种选择: 0=关闭 DDE; 1=热连接; 2=冷连接; 3=通告连接。

如果“组态王 7.0”作为“顾客”向 VB 请求数据, 需要在定义变量时说明服务器程序的三个标识名, 即: 应用程序名设为 VB 可执行程序的名字, 把话题名设为 VB 中窗体的 LinkTopic 属性值, 项目名设为 VB 控件的名字。

18.3.1 组态王 7.0 访问 VB 的数据

在本例中, 假设“组态王 7.0”访问 VB 的数据, “组态王 7.0”作为客户程序向 VB 请求数据。数据流向如下图所示。



使 VB 成为“服务器”很简单, 需要在“组态王 7.0”中设置服务器程序的三个标识名, 并把 VB 应用程序中提供数据的窗体的 LinkMode 属性设置为 1。

1、运行可视化编程工具 Visual Basic

选择菜单“File\New Project”, 显示新窗体 Form1。设计 Form1, 将窗体 Form1 的 LinkMode 属性设置为 1(source), 如图 18.13 所示。

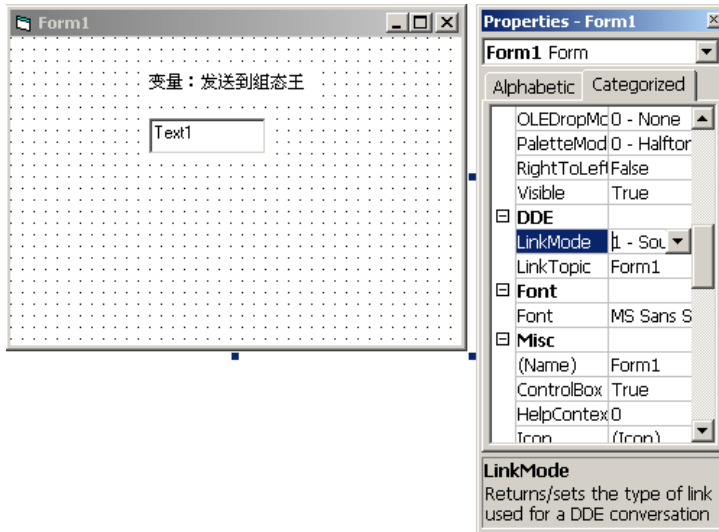


图 18.13 VB 中建立窗体和控件

修改 VB 中窗体和控件的属性：

窗体 Form1 属性:LinkMode 属性设置为 1 (source);LinkTopic 属性设置为 FormTopic, 这个值将在“组态王 7.0”中引用。

文本框 Text1 属性: Name 属性设置为 Text_To_View, 这个值也将将在“组态王 7.0”中被引用。

2、生成 vbdde.exe 文件

在 Visual Basic 菜单中选择“File\Save Project”，为工程文件命名为 vbdde.vbp，这将使生成的可执行文件默认名是 vbdde.exe。选择菜单“File\Make EXE File”，生成可执行文件 vbdde.exe。

3、在“组态王 7.0”中定义 DDE 设备

在工程浏览器中，从左边的工程目录显示区中选择“设备\DDE”，然后在右边的内容显示区中双击“新建”图标，则弹出“设备配置向导”（DDE 设备的配置请参见第六章 I/O 设备管理），已配置的 DDE 设备的信息总结列表框如图 18.14 所示。定义 I/O 变量时要使用定义的连接对象名 VBDDE（也就是连接设备名）。



图18.14 组态王7.0 中定义DDE 设备

4、在工程浏览器中定义新变量

定义新变量，变量名为 FromVBToView，项目名设为服务器程序中提供数据的控件名，此处是文本框 Text_To_View，连接设备为 VBDDE。“变量属性”对话框如图 18.15 所示。

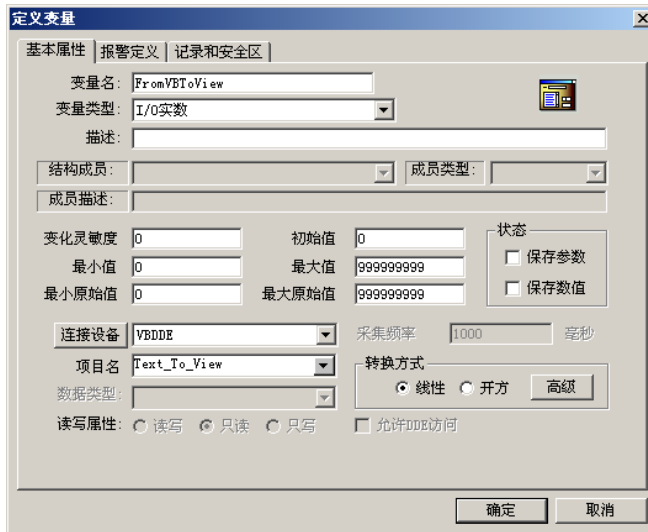


图 18.15 组态王 7.0 中定义 IO 变量

5、创建“组态王 7.0”画面

新建组态王 7.0 画面名为 test，如图 18.16 所示。

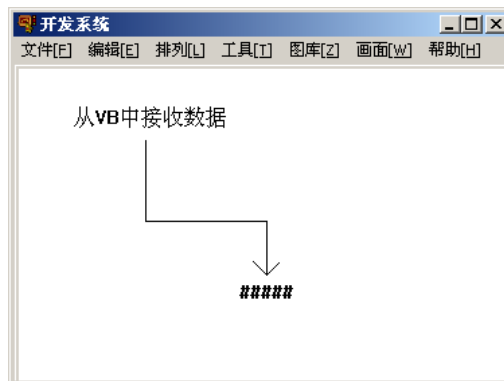


图 18.16 组态王 7.0 中输出来自 VB 的数据

为对象“#####”设置“模拟值输出”的动画连接，如图 18.17 所示。



图18.17 组态王 7.0 中为变量输出建立动画连接

设置完成后，选择菜单“文件\全部存”。选择菜单“数据库\主画面配置”，将画面 test 设置为主画面。DDE 连接设置完成。

6、执行应用程序

在 VB 中选择菜单“Run\Start”，运行 vbdde.exe 程序，在文本框中输入数值。运行组态王 7.0，得到 VB 中的数值。如图 18.18 所示。

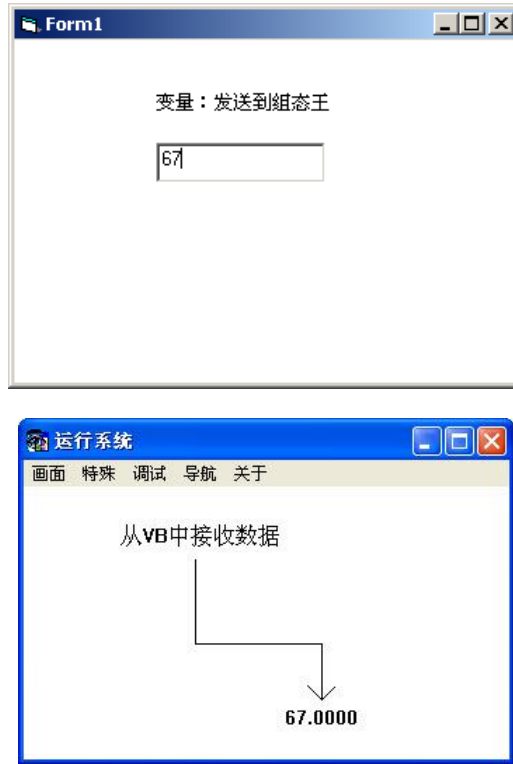


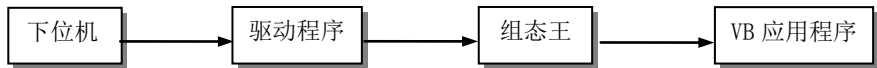
图18.18 组态王7.0 中为变量输出建立动画连接

如果画面运行异常，选择 TouchView 菜单“特殊\重新建立未成功的 DDE 连接”，连接完成后再试一试以上程序。

18.3.2 VB 访问组态王 7.0 的数据

在本例中，假设 VB 访问“组态王 7.0”的数据，VB 作为客户程序向“组态王 7.0”请求数据。“组态王 7.0”通过 OMRON 驱动程序从下位机采集数据，VB 又向“组态王 7.0”

请求数据。数据流向如下图所示。



1、在“组态王 7.0”中定义设备

在工程浏览器中，从左边的工程目录显示区中选择“设备”，然后在右边的内容显示区中双击“新建”图标，则弹出“设备配置向导”（设备的配置请参见第六章 I/O 设备管理，在这里比如建立了 OMRON 的 PLC），已配置的设备的信息总结列表框如图 18.19 所示。



图 18.19 利用设备安装向导定义设备

定义的连接对象名为 OMRON（也就是连接设备名），定义 I/O 变量时要使用此连接设备。

2、在“组态王 7.0”中定义 I/O 变量

在工程浏览器左边的工程目录显示区中，选择“数据库\数据词典”，然后在右边的目录内容显示区中用左键双击“新建”图标，弹出“变量属性”对话框，在此对话框中建立一个 I/O 实型变量。如图 18.20 所示。

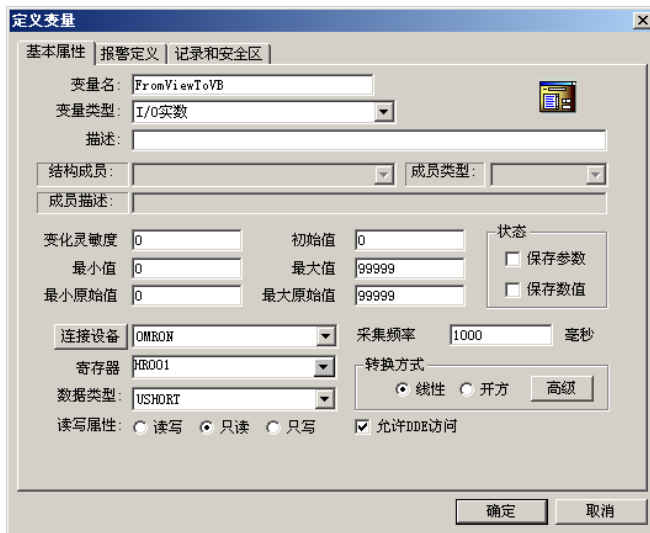


图 18.20 组态王 7.0 定义 IO 变量

变量名设为 FromViewToVB，这个名称由工程人员自己定义。项目名为 OMRON.HR001。选择“允许 DDE 访问”选项。变量名在“组态王 7.0”内部使用，项目名是供 VB 引用的，连接设备为 OMRON，用来定义服务器程序的信息，已在前面定义。

3、创建画面

在组态王 7.0 画面开发系统中建立画面 test1，如图 18.21 所示。

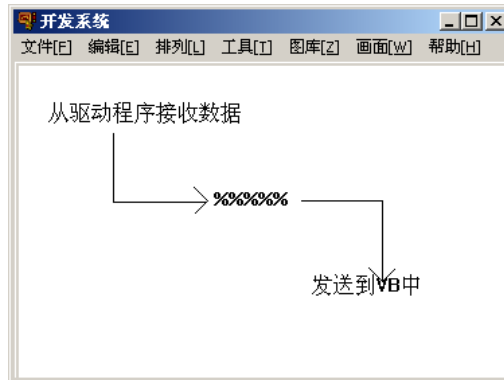


图 18.21 组态王 7.0 中输出变量

为文本对象“%%%%”设置“模拟值输出”动画连接如图 18.22 所示。



图 18.22 组态王 7.0 为变量输出建立动画连接

选择菜单“文件\全部存”，保存画面。选择菜单“数据库\主画面配置”，将画面 test1 设置为主画面。

4、运行可视化编程工具 Visual Basic

继续使用上一节的例子，设计 Form1 如图 18.23 所示。

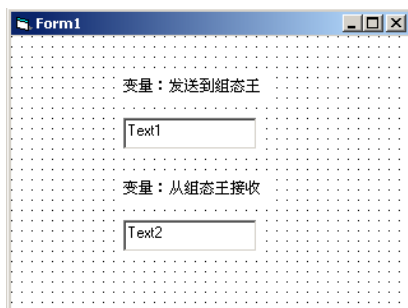


图 18.23 VB 中建立窗体和控件

5、编制 Visual Basic 程序

双击 Form1 窗体中任何没有控件的区域，弹出“Form1.frm”窗口，在窗口内书写 Form_Load 子例程，如图 18.24 所示。

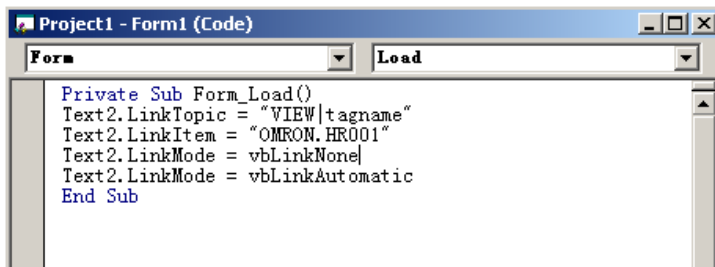


图 18.24 VB 中为控件建立与组态王 7.0 变量的连接

6、生成可执行文件

在 VB 中选择菜单“File\Save Project”保存修改结果。选择菜单“File\Make Exe File”生成 vbdde.exe 可执行文件。激活 OMRON 驱动程序和“组态王 7.0”运行系统 TouchView。在 Visual Basic 菜单中选择“Run\Start”运行 vbdde.exe 程序。窗口

Form1 的文本框 Text2 中显示出变量的值。如图 18.25 所示。

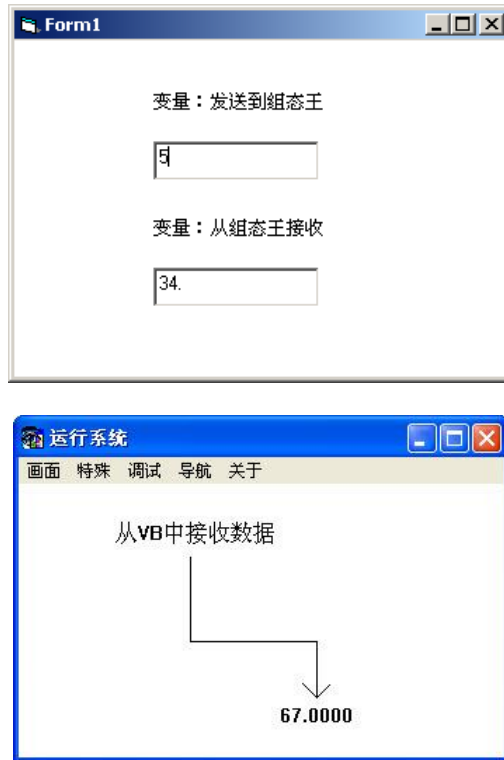


图 18.25 VB 接收组态王 7.0 的数据

18.4 如何重新建立 DDE 连接

18.4.1 重新建立 DDE 连接菜单命令

在 TouchView 启动时, 自动进行 DDE 连接, 若有未成功的连接, 则显示信息如图 18.26 所示。



图18.26 建立DDE 连接未成功信息

对话框各项意义如下:

- ◇ **继续:** 放弃此 DDE 连接的建立, 而继续建立下一个连接。
- ◇ **再试:** 再次尝试建立此 DDE 连接。通常情况下, 在发生建立 DDE 失败时, 往往是没有先启动服务程序, 此时, 可用 Alt+Tab 键切换到文件管理器, 启动服务程序后, 再选此按钮; 也可以暂时放弃建立连接, 进入 TouchView 后选择菜单“特殊\重新建立未成功的连接”。
- ◇ **退出:** 选此按钮, 则整个 TouchView 程序退出。
- ◇ **关闭信息框:** 选此按钮, 关闭“建立 DDE 失败”对话框, 但是当前的 DDE 连接没有成功。

与 DDE 有关的菜单项:

- ◇ **特殊\重新建立 DDE 连接:**
选此菜单后, “组态王 7.0” 先中断已经建立成功的连接 (如果有的话), 然后重新建立所有 I/O 变量与服务程序的连接。
- ◇ **特殊\重新建立未成功的连接:**
TouchView 重新建立未建立成功的 DDE 连接, 已经建立成功的 DDE 连接不受影响。

18.4.2 重新建立 DDE 连接函数

与 DDE 连接有关的函数有两个: ReBuildDDE ();和 ReBuildConnectDDE ();

✧ ReBuildDDE ()

此函数用于重新建立 DDE 连接。

调用形式: ReBuildDDE ();此函数无参数。

✧ ReBuildConnectDDE ()

此函数用于重新建立未成功的 DDE 连接。

调用形式: ReBuildConnectDDE ();此函数无参数。

第十九章 组态王数据库访问 (SQL)

- ☞ 介绍组态王 SQL 访问功能
- ☞ 组态王如何与通用数据库连接

组态王 SQL 访问功能是为了实现组态王和其他 ODBC 数据库之间的数据传输。它包括组态王 SQL 访问管理器、如何配置与各种数据库的连接、组态王与数据库连接实例和 SQL 函数的使用。

组态王 SQL 访问管理器用来建立数据库列和组态王变量之间的联系。通过表格模板在数据库中创建表格，表格模板信息存储在 SQL.DEF 文件中；通过记录体建立数据库表格列和组态王之间的联系，允许组态王通过记录体直接操纵数据库中的数据。这种联系存储在 BIND.DEF 文件中。

组态王可以与其他外部数据库(支持 ODBC 访问接口)进行数据传输。首先在系统 ODBC 数据源中添加数据库，然后通过组态王 SQL 访问管理器和 SQL 函数实现各种操作。

组态王 SQL 函数可以在组态王的任何一种命令语言中调用。这些函数用来创建表格，插入、删除记录，编辑已有的表格，清空、删除表格，查询记录等操作。

19.1 组态王 SQL 访问管理器

组态王 SQL 访问管理器包括表格模板和记录体两部分功能。当组态王执行 SQLCreateTable(); 指令时, 使用的表格模板将定义创建的表格的结构; 当执行 SQLInsert();、SQLSelect();或 SQLUpdate();时, 记录体中定义的连接将使组态王中的变量和数据库表格中的变量相关联。



组态王提供集成的 SQL 访问管理。在组态王工程浏览器的左侧大纲项中, 可以看到 SQL 访问管理器。如图 19.1 所示。

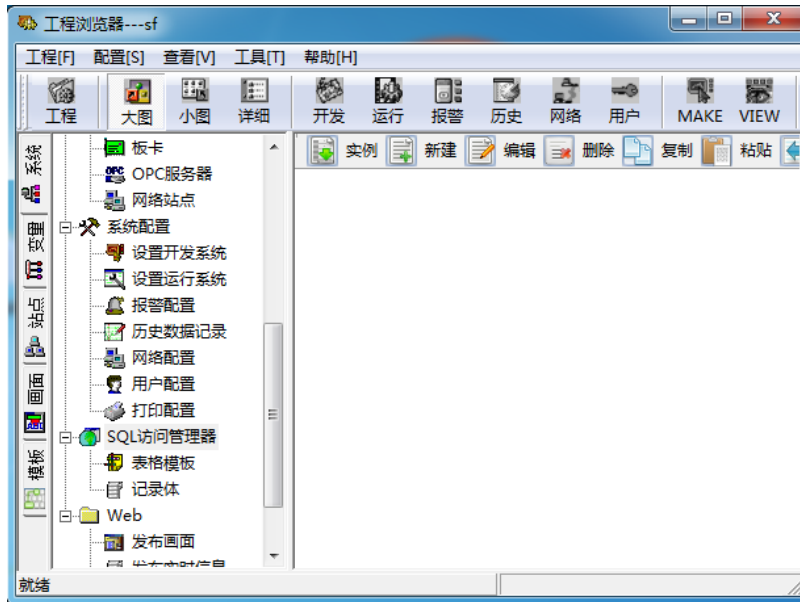


图 19.1 组态王 SQL 访问管理器

19.1.1 表格模板

选择工程浏览器左侧大纲项“SQL 访问管理器文件\表格模板”，在工程浏览器右侧用鼠标左键双击“新建”图标，弹出对话框如图 19.2 所示。该对话框用于建立新的表格模板。



图 19.2 创建表格模板

- ◇ **模板名称:** 表格模板的名称, 长度不超过 32 个字符;
- ◇ **字段名称:** 使用表格模板创建数据库表格中字段的名称, 长度不超过 32 个字符, 如果数据库中的字段名称以数字开头, 如“3Name”, 在定义表格模板时, 名称须以大括号包含, 写为 “[3Name]”;
- ◇ **变量类型:** 表格模板创建数据库表格中字段的类型。点击下拉列表框按钮, 其中有四种类型供选择, 整型、浮点型、定长字符串型、变长字符串型;
- ◇ **字段长度:** 当变量类型中选择“定长字符串型”或“变长字符串型”时, 该项文本框由“灰色”(无效)变为“黑色”(有效)。在文本框中输入字段长度数值, 该数值必须为正整数, 且不大于 127 个字符;
- ◇ **索引类型:** 点击下拉列表框按钮, 其中有三种类型供选择, 有(唯一)、有(不唯一)、没有。索引功能是数据库用于加速字段中搜索及排序的速度, 但可能会使更新

变慢。选择“是（唯一）”可以禁止该字段中出现重复值；

- ◇ **允许为空值：**选中该项，将在前面的方框中出现“✓”标志，表示数据记录到数据库的表格中该字段可以有空值。不选中该项则表示该字段的数据不能为空值；
- ◇ **增加字段：**在上面定义字段中各个项定义好后，点击“增加字段”按钮，将会把定义好的字段增加到显示框中；
- ◇ **删除字段：**从显示框中选中已有字段的字段名称，点击“删除字段”按钮，将会把定义好的字段从显示框中删除；
- ◇ **修改字段：**从显示框中选中已有字段的字段名称，此时该字段各项属性显示在定义字段各项中，对各个属性进行修改，然后点击“修改字段”按钮，将会把修改后的字段重新显示在显示框中，修改完字段后，必须点击“确认”按钮才会保存修改内容；
- ◇ **上移一行：**从显示框中选中已有字段的字段名称，点击“上移一行”按钮，将会把选中的字段向上移动一行，在数据库创建表格中将改变该字段位置；
- ◇ **下移一行：**从显示框中选中已有字段的字段名称，点击“下移一行”按钮，将会把选中的字段向下移动一行，在数据库创建表格中将改变该字段位置。



例：创建一个表格模板

创建一个表格模板:table1; 定义三个字段: salary (整型)、name (定长字符串型, 字段长度: 127)、age (整型)。如图 19.3 所示。



图 19.3 定义表格模板 table1

19.1.2 记录体

记录体用来连接表格的列和组态王数据词典中的变量。选择工程浏览器左侧大纲项“SQL 访问管理器文件\记录体”，在工程浏览器右侧用鼠标左键双击“新建”图标，弹出对话框如图 19.4 所示。该对话框用于建立新的记录体。



图 19.4 创建记录体

- ◇ **记录体名:** 记录体的名称, 长度不超过 32 个字符;
- ◇ **字段名称:** 数据库表格中的列名, 长度不超过 32 个字符;
- ◇ **变量:** 与数据库表格中指定列相关联的组态王变量名称。点击右边“?”按钮, 弹出“选择变量名”窗口, 可以从中选择变量;
- ◇ **增加字段:** 在上面定义完字段名称和组态王变量后, 点击“增加字段”按钮, 将会把定义好的字段增加到显示框中;
- ◇ **删除字段:** 从显示框中选中已有字段的字段名称, 点击“删除字段”按钮, 将会把定义好的字段从显示框中删除;
- ◇ **修改字段:** 从显示框中选中已有字段的字段名称, 此时该字段名称和对应组态王变量名称会显示在各项中, 对各项进行修改, 然后点击“修改字段”按钮, 将会把

修改后的字段重新显示在显示框中，修改完字段后，必须点击“确认”按钮才会保存修改内容；

- ◇ **上移一行**：从显示框中选中已有字段的字段名称，点击“上移一行”按钮，将会把选中的字段向上移动一行；
- ◇ **下移一行**：从显示框中选中已有字段的字段名称，点击“下移一行”按钮，将会把选中的字段向下移动一行。
- ◇ **复制记录体**：选中已定义的记录体，点右键，从弹出的右键菜单中选择“复制编辑”，最后点“确认”，即可实现对记录体的复制。



例：创建一个记录体

定义三个组态王变量，分别为：record1（内存实型）、name（内存字符串型）、age（内存整型）；

创建一个记录体: BIND1；

定义三个字段：salary（对应组态王变量 record1）、name（对应组态王变量 name）、age（对应组态王变量 age）。如图 19.5 所示。



图 19.5 创建记录体 BIND1

19.2 如何配置与数据库的连接

19.2.1 定义 ODBC 数据源

组态王 SQL 访问功能能够和其他外部数据库（支持 ODBC 访问接口）之间的数据传输。实现数据传输必须在系统 ODBC 数据源中定义相应数据库。

进入“控制面板”中的“管理工具”，用鼠标双击“数据源 (ODBC)”选项，弹出“ODBC 数据源管理器”对话框，如图 19.6 所示。



图 19.6 ODBC 数据源管理器

有些计算机的 ODBC 数据源是中文的 (如上图所示), 有些的是英文的, 视机器而定, 但是两种的使用方法相同。

“ODBC 数据源管理器”对话框中前两个属性页分别是“用户 DSN”和“系统 DSN”，二者共同点是：在它们中定义的数据源都存储了如何与指定数据提供者在连接的信息，但二者又有所区别。在“用户 DSN”中定义的数据源只对当前用户可见，而且只能用于当前机器上；在“系统 DSN”中定义的数据源对当前机器上所有用户可见，包括 NT 服务。因此用户根据数据库使用的范围进行 ODBC 数据源的建立。



例如：以 Microsoft Access 数据库为例，建立 ODBC 数据源。

- ✧ 在机器上 D 盘根目录下建立一个 Microsoft Access 数据库, 名称为: SQL 数据库. mdb;
- ✧ 双击“数据源 (ODBC)”选项, 弹出“ODBC 数据源管理器”对话框, 点击“系统

DSN”属性页，如图 19.7 所示。

- ✧ 单击右边“增加”按钮，弹出“创建新数据源”窗口，从列表中选择“Microsoft Access Driver (*.mdb)”驱动程序，如图 19.8 所示。
- ✧ 单击“完成”按钮，进入“ODBC Microsoft Access 安装”对话框，如图 19.9 所示。

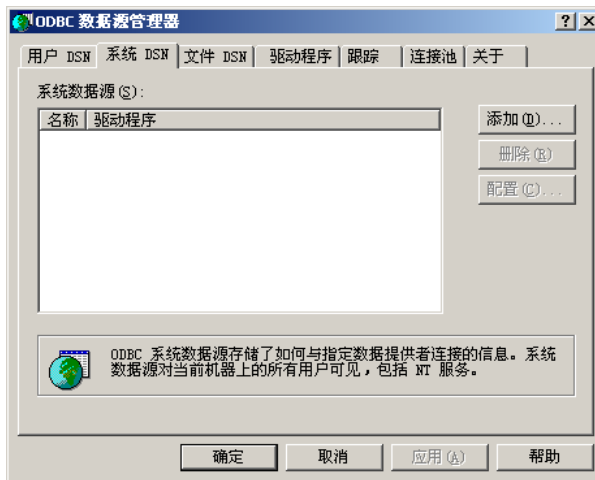


图 19.7 “系统 DSN”属性页



图 19.8 创建新数据源



图 19.9 “ODBC Microsoft Access 安装”对话框

- ◇ 在“数据源名”中输入数据源名称：mine；单击“选择”按钮，从计算机上选择数据库，选择好数据库后的对话框如图 19.10 所示。



图 19.10 “ODBC Microsoft Access 安装”对话框完成

- ✧ 点击“确定”按钮，完成数据源定义，回到“ODBC 数据源管理器”窗口，点击“确定”关闭“ODBC 数据源管理器”窗口。

完成 Microsoft Access 数据库 ODBC 数据源的定义。其它类型的数据库定义方法类似。

19.2.2 组态王支持的数据库及配置

19.2.2.1 Oracle6 数据库

为了和 Oracle6 通讯需要进行如下设置：

- ✧ 配置你的 Windows 数据库客户；
- ✧ 启动 SQL*Net TSR 和 NETINIT.EXE 程序；
- ✧ 通过在组态王命令语言中执行 SQLConnect() 函数建立和 Oracle 的连接。

启动 SQL*Net TSR 和 NETINIT.EXE

Oracle 数据库可以通过在本地机上安装 Oracle's SQL*Net 来访问。Oracle's SQL Net 包括 SQL*Net TSR 和 NETINIT.EXE 程序。为了和 Oracle 数据库连接，这两个程序都要运

行。SQL*Net TSR 必须在运行 Windows 之前在 DOS 环境中运行。NETINIT.EXE 在 Windows 中运行。

SQLConnect () 格式

本函数用来和 Oracle 数据库连接。格式如下：

```
SQLConnect (ConnectionID, “<attribute>=<value>;<attribute>=<value>; ...”);
```

下面介绍 Oracle 中使用的属性：

属性	值
DSN	微软 ODBC 管理器配置的数据源名字
UID	用户名
PWD	密码
SRVR	指明服务器和数据库



例如：

```
SQLConnect (ConnectionID, “DSN=Oracle_Data;UID=asia;PWD=abcd;SRVR=B:MKTG_SRV”);
```



注意：

SRVR 比较难于设置。根据 SQL*Net 驱动不同，设置值将不同。这一信息可以在 SQL*Net 文档中得到（或可能在 CONFIG.ORA 文件的“remote=”行中得到）。

支持的数据类型

对于 Oracle6 支持两种数据。char 类型对应着组态王中的字符串变量,缺省长度为 1。Oracle6 最多支持 255 个字符。number 类型对应着组态王中的整数和实数变量。

19.2.2.2 Oracle7.2 数据库

为了和 Oracle 通讯需要进行如下设置:

- ✧ 在组态王本机上安装 Oracle Standard Client;
- ✧ 运行 SQL_Net Easy 配置为 SQL 连接分配字符串;
- ✧ 创建一个数据源名;
- ✧ 使用 SQLConnect() 连接。

配置 SQL_Net

1. 启动 Oracle 的 SQL_Net Easy Configuration;
2. 缺省下,服务器的化名将以 wgs_ServerName_orcl 开始。数据库的化名在组态王 SQLConnect() 函数中使用;
3. 修改化名,单击 OK;
4. 单击 Modify Database Alias Select Network protocol。命名管道是 Oracle 服务器的计算机名。

配置数据源名

1. 启动控制面板中的 ODBC。单击“System DSN”属性页。单击“Add”按钮。弹出“Create New Data Source”对话框;
2. 选择 Oracle7 ODBC 驱动,然后单击 Finish。ODBC Oracle Driver Setup 将会弹出。在 Data Source Name 框中,键入你的 Oracle 服务器名;

- 单击 Advanced。使用 ODBC Oracle Advanced Driver Setup 对话框。单击 Close。ODBC Data Source Administrator 对话框将出现，如图 19.11 所示。
- 单击确定完成。

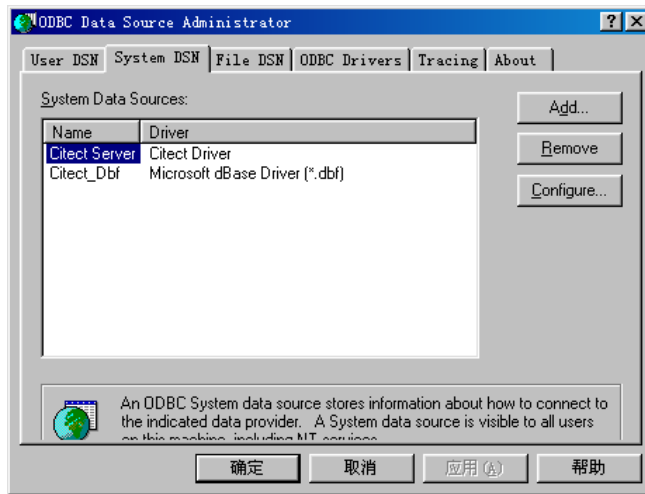


图 19.11 定义 Oracle7.2 数据源

为 Oracle 数据域登录日期和时间

为了登录日期和时间，你必须配置记录体（捆绑表）。

- 在组态王工程浏览器中，单击 SQL 访问管理器中的记录体，将弹出创建记录体对话框；
- 在字段名称栏中，输入 DATE_TIME delim() 函数；
- 在组态王变量栏中，输入你想要捆绑的变量，如图 19.12 所示。
- 在组态王命令语言中，给 DATE_TIME_TAG 变量赋予当前的日期和时间值。

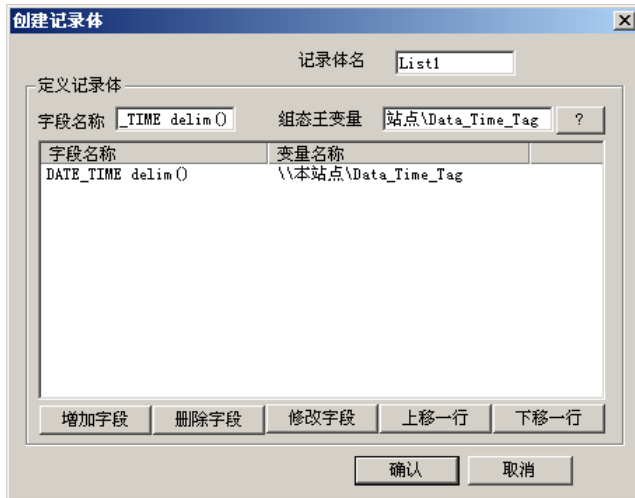


图 19.12 配置记录体

19.2.2.3 SyBase 或 MS SQLServer 数据库

为了和 SyBase 或 Microsoft SQL Server 通讯需要进行如下设置:

- ◇ 配置 Windows 的数据库用户;
- ◇ 使用 SQLConnect() 函数连接。

配置数据库

1. 打开 Windows 控制面板的 32 位 ODBC 数据源管理器。单击添加, 选择 SQL Server, 弹出 ODBC SQL Server 配置画面:
2. 在 Data Source Name 栏填写数据源名称。在 Server 栏填写数据库 Server 名称。在网络地址中, 填写 SQL Server 的访问地址。单击 Option>>, 在数据库名称栏中填写数据库名称, 如图 19.13 所示。

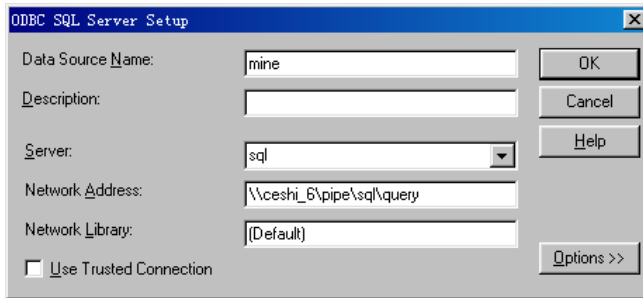


图 19.13 配置 SQL Server 数据库



注意:

SQL Server 名称必须和网络上 SQL Server 的名称一致。具体名称通过 SQL Server 管理程序“SQL Enterprise Manager”确认。

连接格式

SQLConnect (ConnectionID, ”<attribute>=<value>;<attribute>=<value>;...”);

SyBase 和 SQL Server 用到的属性如下:

属性	值
DSN	ODBC 中定义的数据源名
UID	登录 ID 号
PWD	密码, 区分大小写
SRVR	数据库所在的计算机名
DATABASE	所要访问的数据库名



例如:

SQLConnect(ConnectionID, "DSN=wang; DATABASE=kingivew;UID=user1; PWD= abcd");

组态王 7.0 目前支持 SQL Server 以下数据类型:

SQL Server 数据类型	组态王数据类型	说明
Bigint	内存实数	确切数值,备注 1
bit	内存离散	1 位二进制数据
char	内存字符串	定长字符串
datetime	内存字符串	日期数据
decimal	内存实数	以字符存贮的确切数值
float	内存实数	双精度浮点
int	内存整数	整数
money	内存整数	货币
nchar	内存字符串	定长 Unicode 字符串
ntext	内存字符串	变长 Unicode 字符串
numeric	内存实数	有符号的确切数值
nvarchar	内存字符串	Unicode 字符变长流
real	内存实数	浮点数
smalldatetime	内存字符串	yyyymmddhhmmss

smallint	内存整数	确切数值 32768
smallmoney	内存整数	货币
sql_variant	内存字符串	备注 1
text	内存字符串	变长字符
tinyint	内存整数	整数,256
varchar	内存字符串	变长流字符串

暂时还未支持的 SQL Server 数据类型:

SQL Server 数据类型	说明
binary	二进制数据流
image	变长二进制数据
varbinary	变长二进制
timestamp	二进制数据流
uniqueidentifier	GUID 全局唯一标识

备注: bigint, sql_variant 为 SQL SERVER 2000 开始支持, SQL Server 7.0 或更上版本并不支持. 组态王的对这两种类型支持也不完善, 建议尽量不要采用.



注意:

当使用 Sybase Database Server, 数据类型 (char, int, float) 区分大小写, 并且必须小写。Sybase 或 Microsoft SQL Server 在同一时刻只能支持一个激活的语句。列名和表名也区分大小写。表名必须由表的拥有者充分授权。

19.2.2.4 dBase 数据库

为了和 dBASE 连接, 必须执行 SQLConnect () 函数。

SQLConnect () 格式

SQLConnect (ConnectionID, “<attribute>=<value>; <attribute>=<value>;...”);

下面描述 dBASE 的属性。请注意遵守列出顺序。

属性	值
DSN	微软 Microsoft ODBC 配置的数据源名
CS	决定数据存储格式是以 IBM PC 符号还是以 ANSI 符号。缺省设置是 IBMPC
DB	指定 dBASE 存储的路径, 缺省为当前路径
FOC	已打开未使用的文件的最大数
LCK	决定数据库文件记录访问等级。有效值为 FILE, RECORD, (缺省), NONE
CSZ	用来缓存数据库记录的 64K 模块的个数。此值越大, 性能越

	好。缺省为 4。能够设定的最大值取决于有效的系统内存。
USF	决定驱动器何时更新数据库。当此值为 1 时，驱动器在每次 COMMIT 时都更新数据库。这将降低系统性能；缺省为 0，这意味着驱动器在文件关闭时更新数据库。但如果机器在文件关闭之前崩溃将使新增加的记录丢失
MS	本选项决定是否支持先前的 Q+E 软件。MS=0 时支持。缺省为 1
LCOMP	决定选用 dBASE 兼容的锁定还是 Q+E 兼容的锁定方式
COMP	提供对 Q+E 软件的向前兼容。COMP=DBASE 则向前兼容；COMP=ANSI 则是简装型

支持的数据类型

SQL 管理器支持 dBASE 的三种数据类型。char 类型包含定长的字符串，对应组态王中的字符串变量。数据库 dBASE 最大支持 254 个字符。numeric 类型和 float 类型对应组态王中整型或实型变量。必须设定变量长度。格式为十进制宽度。

19.2.2.5 MS Access 数据库

为了和 Microsoft Access 连接，必须执行 SQLConnect() 函数。

SQLConnect() 格式

SQLConnect (ConnectionID, “<attribute>=<value>;<attribute>=<value>;...”); 以下介绍 Access 用到的属性。

属性	值
----	---

DSN	ODBC 中设置的数据源名称
-----	----------------

支持的数据类型

SQL 访问管理器支持 Access 数据库的五种数据类型。有效的数据类型种类由你所使用的 ODBC 的版本所决定。类型 text 包括定长字符串和组态王中的字符串变量相对应，必须设定参数。Access 数据库最多支持 255 个字符。

19.3 组态王 SQL 使用简介

19.3.1 如何使组态王与数据库建立连接

使用组态王与数据库进行数据通讯，首先要建立它们之间的连接。下面通过一个实例介绍如何使组态王与数据库建立连接。



例：组态王与 Microsoft Access 数据库建立连接

- ◇ 继续使用“19.2.1 定义 ODBC 数据源”中的例子。在机器上 D 盘根目录下建立的“SQL 数据库.mdb”数据库中建立一个名为 kingview 的表格。在组态王的数据词典里定义新变量，变量名称：DeviceID，变量类型：内存整型；
- ◇ 然后在本机上的 ODBC 数据源中建立一个数据源，比如数据源名为 mine。详细配置请参照“19.2.1 定义 ODBC 数据源”一节；
- ◇ 在组态王工程浏览器中建立一个名为 BIND 的记录体，定义一个字段：name（对应组态王内存字符串变量 name），详细定义请参照“19.1.2 记录体”一节；
- ◇ 连接数据库：新建画面“数据库连接”，在画面上作一个按钮，按钮文本为：“连接

数据库”，在按钮“弹起时”动画连接中使用 `SQLConnect()` 函数和 `SQLSelect()` 函数建立与“mine”数据库进行连接：`SQLConnect(DeviceID, "dsn=mine;uid=;pwd=")`；
/*建立和数据库 mine 连接，其中 DeviceID 是用户创建的内存整型变量，用来保存 `SQLConnect()` 函数为每个数据库连接分配的一个数值*/
/*注意在执行 `SQLDisconnect`（断开和数据库的连接函数）之前，重复执行 `SQLConnect` 将会增加 DeviceID，DeviceID 最多为 255*/
`SQLSelect(DeviceID, "kingview", "BIND", "", "")`；
/* `SQLSelect()` 函数：访问一个数据库并返回一个表的信息，此信息可以被查询、删除、清空等操作*/
以上指令执行之后，使组态王与数据库建立了连接。

19.3.2 如何创建一个表格

组态王与数据库连接成功之后，可以通过操作组态王在数据库中创建表格。下面通过一个实例介绍如何创建一个表格。



例如：创建数据库表格

- ✧ 在组态王中创建一个表格模板:table1。定义三个字段：salary（整型）、name（定长字符串型，字段长度：255）、age（整型）。详细定义请参照“19.1.1 表格模板”一节；
- ✧ 将上节实例中画面上“连接数据库”按钮“弹起时”动画连接命令语言该为：
`SQLConnect(DeviceID, "dsn=mine;uid=;pwd=")`；

- ◇ 创建数据库表格：在“数据库连接”画面上新作一个按钮，按钮文本为：“创建表格”，在按钮“弹起时”动画连接中使用 `SQLCreateTable()` 函数创建表格。

```
SQLCreateTable( DeviceID, "KingTable", "table1" );
```

```
/*创建数据库表格名称为：KingTable*/
```

该命令用于以表格模板“table1”的格式在数据库中建立名为“KingTable”的表格。在自动生成的 KingTable 表格中，将生成三个字段，字段名称分别为：salary, name, age。每个字段的变量类型、变量长度及索引类型与表格模板“table1”中的定义所决定。

19.3.3 如何将数据存入数据库

创建数据库表格成功之后，可以将组态王中的数据存入到数据库表格中。下面通过一个实例介绍如何将数据存入数据库。



例：将数据存入数据库

- ◇ 在组态王中创建一个记录体: BIND1。定义三个字段：salary（整型，对应组态王变量 record1）、name（定长字符串型，字段长度：255，对应组态王变量 name）、age（整型，对应组态王变量 age）。详细定义请参照“19.1.2 记录体”一节；
- ◇ 在“数据库连接”画面上作一个按钮，按钮文本为：“插入记录”，在按钮“弹起时”动画连接中使用 `SQLInsert()` 函数：

```
SQLInsert( DeviceID, "KingTable", "BIND1" );
```

该命令使用记录体 BIND1 中定义的连接在表格 KingTable 中插入一个新的记录。

该命令执行后，组态王运行系统会将变量 salary 的当前值插入到 Access 数据库表格“KingTable”中最后一条记录的“salary”字段中，同理变量 name、age 的当前值分别赋给最后一条记录的字段：name、age 值。运行过程中可随时点击该按钮，执行插入操作，在数据库中生成多条新的记录，将变量的实时值进行保存。

19.3.4 如何进行数据查询

组态王在运行过程中还可以对已连接的数据库进行数据查询。下面通过一个实例介绍如何进行数据查询。



例：进行数据查询

- ◇ 在组态王中定义变量。这些变量用于返回数据库中的记录。“记录 salary”：内存实型；“记录 name”：内存字符串型；“记录 age”：内存整型。定义记录体 BIND2，用于定义查询时的连接。如图 19.14 所示。



图 19.14 数据查询记录体

- ◇ 在“数据库连接”画面上作一个按钮，按钮文本为：“得到选择集”，在按钮“弹起时”动画连接中使用 SQL 连接函数，得到一个指定的选择集：

```
SQLSelect( DeviceID, "KingTable", "BIND2", "", "" );
```

该命令选择表格 KingTable 中所有符合条件的记录，并以记录体 BIND2 中定义的连接返回选择集中的第一条记录。此处没有设定条件，将返回该表格中所有记录。

执行该命令后，运行系统会把得到的选择集的第一条记录的“salary”字段的值赋给记录体 BIND2 中定义的与其连接的组态王变量“记录 salary”，同样“KingTable”表格中的 name、age 字段的值分别赋给组态王变量记录 name、记录 age。

- ◇ 画面中查询返回值的显示：在画面上做三个“##”文本，分别定义值输出连接到变量“记录 salary”、“记录 name”和“记录 age”，如图 19.15 所示。



图19.15 查询返回值画面

在执行 SQLSelect() 函数后，首先返回选择集的第一条记录，在画面上“###”将显示返回值。

✧ 查询记录：在“数据库连接”画面上做四个按钮。

按钮文本：第一条记录 “弹起时”动画连接：SQLFirst(DeviceID);

按钮文本：下一条记录 “弹起时”动画连接：SQLNext(DeviceID);

按钮文本：上一条记录 “弹起时”动画连接：SQLPrev(DeviceID);

按钮文本：最后一条记录 “弹起时”动画连接：SQLLast(DeviceID);

✧ 断开连接：在“数据库连接”画面上做一个按钮，按钮文本：断开连接，“弹起时”

动画连接：SQLDisconnect(DeviceID);

该命令用于断开和数据库的连接。

最后的生成画面如图 19.16 所示：



图 19.16 组态王 SQL 数据库访问

定义复杂的查询

SQL 访问管理允许用户指定一个复杂的查询和设计自己的 SQL 语句。这些查询可以动态的建立或由外部文件包含。另外，这些查询可能包含需要在运行时转化成查询的参数。SQL 访问管理 API 允许用户执行任何一个用户的数据库支持的 SQL 语句，并且返回查询结果。同时，存储过程也可有效执行。（存储过程不能被充分支持）更详细资料，请参阅“支持存储过程”。

复杂的查询必须调用 `SQLSetStatement` 函数。

建立动态查询

为了建立动态查询,需用到两个函数: `SQLSetStatement()` 和 `SQLAppendStatement()`。
`SQLSetStatement()` 开始一个新的 SQL 语句。它可以是任何有效的 SQL 语句,包括存储过程的名字。因为组态王只支持 131 个字符, `SQLAppendStatement()` 用来附加字符串到语句中。



例如:

```
ResultCode = SQLSetStatement ( ConnectionID, "Select LotNo, LotName,  
LotDescription, LotQuantity from LotInfo, ProductionInfo" );  
  
ResultCode = SQLAppendStatement ( ConnectionID, "where LotInfo.LotNo  
= ProductionInfo.LotNo" );  
  
ResultCode = SQLAppendStatement ( ConnectionID, "order by LotNo, LotName,  
LotQuantity" );
```

这时, 语句就可以执行了。



注意:

许多数据库列名和表名区分大小写。

从文件中读 SQL 语句

用户可以在其他软件例如, Microsoft Access 和其他数据库工具建立查询, 然后使用组态王的 `SQLLoadStatement()` 函数调用这一查询。



例如:

```
ResultCode=SQLLoadStatement (ConnectionID, "c:\myappdir\lotquery. sql"  
);
```

这时, 语句就可以执行了。

修改扩展的 SQL 语句

为了提供完全的 SQL 功能, SQL Access Manager 允许用户指定一个包含组态王变量的 where 语句。为了允许运行时指定 SQL 参数, 需用到以下函数:

```
SQLPrepareStatement (), SQLSetParamType (), QLClearStatement (), SQLClearParam ();
```

为了实现 SQL 语句的替换, 在用户想要在以后再指定的参数位置替代为一个“?”。这一语句是一个替代语句, 当参数设置后, 语句即可以执行。

SQLPrepareStatement () 为一个执行准备一个语句。它并不执行语句, 它只是激活语句使用户能设置参数。



例如:

```
ResultCode = SQLSetStatement ( ConnectionID, "select LotNo, LotName,  
LotDescription, LotQuantity from LotInfo, ProductionInfo" );  
  
ResultCode = SQLAppendStatement ( ConnectionID, "where LotInfo.LotNo  
= ?");  
  
ResultCode = SQLAppendStatement ( ConnectionID, "order by LotNo, LotName,  
LotQuantity");
```

```
ResultCode = SQLPrepareStatement ( ConnectionID, SQLHandle );
```

```
ResultCode = SQLSetParameInt ( SQLHandle, 1, tagLotNumber );
```

现在，语句已经准备好，可以执行了。

一旦语句执行，并且用户不再使用准备的语句，可以调用 SQLClearStatement() 来释放语句相关的资源。

执行扩展的 SQL 语句

根据前文可知，语句可以被动态创建或从文件中读出，可以被准备和修改。现在，可以执行语句了。SQL 访问管理器使用 SQLExecute() 函数来实现此功能。SQLExecute() 将执行当前激活的语句或原来准备好的语句句柄。



例 1:

```
ResultCode = SQLLoadStatement ( ConnectionID, "c:\sql\lotquery.sql" );
```

```
ResultCode = SQLExecute ( ConnectionID, "BIND1", 0 );
```



例 2:

```
ResultCode=SQLSetStatement( DeviceID, "select name from kingviewT");
```

```
/*SQL 查询语句：从表格 kingviewT 选取 name 列*/
```

```
ResultCode=SQLAppendStatement( DeviceID, "where salary = ?");
```

```
/*使用? 以后赋值*/
```

```
ResultCode=SQLPrepareStatement( DeviceID, handle);
```

```
/*准备语句*/  
  
ResultCode=SQLSetParamInt (handle, 1, value );  
  
/*为? 设置参数 value 为内存整数*/  
  
ResultCode=SQLExecute( DeviceID, "BIND1" , handle );  
  
/*执行*/
```



例 3:

```
SQLSetStatement( ConnectID, "select agg from kingviewT" );  
  
SQLExecute( ConnectID, 0 );
```

例 3 中, SQLExecute() 函数中句柄设置为零, 语句没有必要用 SQLPrepare() 来准备。

支持存储的过程

SQLExecute 函数支持存储过程的执行。假设用户在数据库服务器上建立了一个名为“LotInfoProc”的过程, 此过程包含语句: “Select LotNo, LotName from LotInfo”。用户可以在组态王命令语言中写入以下指令实现:

当使用微软 SQL Server:

```
ResultCode = SQLSetStatement ( ConnectionID, "LotInfoProc" );  
  
ResultCode = SQLExecute( ConnectionID, "BIND1", 0 );
```

当使用 Oracle 或 Microsoft Access:

```
ResultCode = SQLSetStatement ( ConnectionID, "{CALL LotInfoProc}" );
```

```
ResultCode = SQLExecute( ConnectionID, "BIND1", 0 );
```

19.3.5 SQL 函数及 SQL 函数的参数

组态王使用 SQL 函数和数据库交换信息。这些函数是组态王标准函数的扩充，可以在组态王的任意一种命令语言中使用。这些函数允许你选择，修改，插入，或删除数据库表中的记录。

SQL 函数

以下列出所有的 SQL 函数。值得注意的是 SQL 查询功能是同步的，在 SQL 查询功能结束之前，控制权不能返回组态王（包括趋势曲线等将被挂起）。所有的 SQL 函数（除 SQLNumRows() 之外）都会返回一个结果代码。如果这个代码不为零，表明调用函数失败。结果代码可以通过 SQLErrorMsg() 函数得到。

SQL 函数的通常格式如下：

```
SQLFunction(Parameter1, Parameter2, ...)
```

需要更详细的资料，请参考“第十章命令语言程序”。

函数	功能
SQLAppendStatement	使用 SQLStatement 的内容继续一个 SQL 语句
SQLClearParam	清除特定参数的值
SQLClearStatement	释放和 SQLHandle 指定的语句相关的资源
SQLClearTable	删除表格中的记录，但保留表格

SQLCommit	定义一组 transaction 命令的结尾
SQLConnect	连接组态王到 connectstring 指定的数据库中
SQLCreatTable	使用表格模板中的参数在数据库中
SQLDelete	删除一个或多个记录
SQLDisconnect	断开和数据库的连接
SQLDropTable	破坏一个表格
SQLEndSelect	在 SQLSelect() 后使用本函数来释放资源
SQLErrorMsg	返回一个文本错误信息, 此错误信息和特定的结果代码相关
SQLExecute	执行一个 SQL 语句。如果这个语句是一个选择语句, 捆绑表中的参数所指定的名字用来捆绑数据词典中变量和数据库的列
SQLFirst	选择由 SQLSelect() 选择的表格中的首项记录
SQLGetRecord	从当前选择缓存区返回由 RecordNumber 指定的记录
SQLInsert	使用捆绑表中指定的变量中的值在表格中插入一个新记录。捆绑表中的参数定义了组态王中变量和数据库表格列的对应关系
SQLInsertEnd	释放插入语句
SQLInsertExecute	执行已经准备的语句

SQLInsertPrepare	准备一个插入语句
SQLLast	选择 SQLSelect() 指定表格中的末项
SQLLoadStatement	读包含在 FileName 中的语句，它类似与 SQLSetStatement() 创建的语句，能被 SQLAppendStatement() 挂起，或由 SQLExecute() 执行，每个文件中只能包含一个语句
SQLNext	选择表中的下一条记录
SQLNumRows	指出有多少条记录符合上一次 SQLSelect() 的指定
SQLPrepareStatement	本语句为 SQLSetParam() 准备一个 SQL 语句。一个语句可以由 SQLSetStatement(), 或 SQLLoadStatement() 创建
SQLPrev	选择表中的上一条记录
SQLRollback	
SQLSelect	访问一个数据库并返回一个表的信息，此信息可以被 SQLFirst(), SQLLast(), SQLNext() 和 SQLPrev()
SQLSetParamChar	将指定的参数设成特定的字符串。本函数可以调用多次，以设置参数值
SQLSetParamDate	将指定的日期参数设置成特定的字符串

SQLSetParamDateTi me	将指定的日期时间参数设置成特定的字符串
SQLSetParamDecima l	将指定的十进制参数设置成特定的字符串
SQLSetParamFloat	将指定的参数设置成特定的值
SQLSetParamInt	将指定的参数设置成特定的值
SQLSetParamLong	将指定的参数设置成特定的值
SQLSetParamNull	将指定的参数设置成空值
SQLSetParamTime	将指定的时间参数设置成特定的字符串
SQLSetStatement	启动一个语句缓存区，此语句缓存区由 SQLStatement() 使用
SQLTransact	定义了一组访问指令的开始。在 SQLTransact() 指令和 SQLCommit() 指令之间的一组指令称为一个访问组。一个访问组可以象单个访问一样操作。在 SQLTransact() 指令执行后，所有接下的操作都不委托给数据库，直到 SQLCommit() 指令执行
SQLUpdate	用当前的组态王变量更新数据库中的记录
SQLUpdateCurrent	更新数据库中的记录。注意：使用 SQLUpdateCurrent 时要求记录体关联的至少一个字 段为表中的不可重复字段，并且该字段类型不为

	自动编号，不为浮点数。
--	-------------

函数的参数

下面将介绍 SQL 函数用到的参数。在命令语言中，当一个参数用引号括起，参数是一个额外的字符串，如果没有引号，参数将认为是组态王的一个变量。大部分 SQL 函数调用以下参数的一个或多个：

BindList	SQL. DEF 文件中定义的捆绑列表名
ConnectionID	用户创建的内存整型变量，用来保存 SQLConnect() 函数为每个数据库连接分配的一个数值
ConnectionString	指示数据库以及任何附加登录信息的字符串
ErrorMsg	出错信息，更详细的解释，请参考第 4 节
FileName	包含信息的文件名
MaxLen	列的最大容量。这一设定决定参数是字符串型还是长字符串型。如果 MaxLen 小于数据库允许的最大长度，参数是字符串型，否则是长字符串型
OrderByExpression	定义排序的列和方向。只有列可以进行排序。格式：ColumnName[ASC DESC]。例：为名为 manager 的列进行升排序，“manager ASC”。为多列排序，格式：ColumnName[ASC DESC], ColumnName[ASC DESC]
ParameterNumber	语句中实际的参数个数
ParameterValue	设定的实际值

Precision	十进制值的精度，字符串的最大长度，日期时间的字节长度
RecordNumber	返回的实际记录个数
ResultCode	大部分 SQL 函数返回一个整数。如果函数成功，返回 0，如果失败，返回一个负值
Scale	十进制数的量程。本参数仅在参数设置为空时有用
SQLHandle	当使用高级函数时，SQL 返回 SQLHandle，供内部使用
SQLStatement	实际语句
TableName	数据库中表名
TemplateName	表格模板名
WhereExpression	定义一个条件，此条件对表格中所有的行或为真，或为假。命令只对条件为真的行起作用。格式： ColumnName comparison_operator expression 注意：如果列为字符串类型，表达式必须带单引号。 下例将选择所有 name 列为 kingview 的行： name=' kingview' 下例选择 agg 列的值介于 20 到 30 之间的行： agg>=20 and agg<30

19.3.6 如何得到 SQL 错误信息

本节介绍如何使用 SQL 函数返回的结果代码分析出错信息。

错误分析函数

所有的 SQL 函数返回一个结果代码。SQLErrorMsg() 函数能返回结果代码的错误信息。



例：

```
SQLErrorMsg( ResultCode, buf );
```

这里：ResultCode 是前一个 SQL 函数得到的整数。

Buf 为与该错误码相对应的文本解释，为一字符串变量。

结果代码错误信息

SQLErrorMsg() 函数将设定组态王字符串变量 Buf。以下是 SQL 函数可能出现的结果代码和它们的相应的信息描述：

结果代码	错误信息	描述
0	没有发现错误	指令执行成功
-5	没有更多的行	已经到了表格中的最后一条记录
-7	数据库连接失败	
-9	无效的数据库设	

	备	
-10	无效的选择集	
-12	无效的记录体	
-13	无效的表格模板	
-14	无效的句柄	
-1001	内存溢出	执行本函数内存已经不足
-1002	无效的连接	传递给函数的 ConnectionID 无效
-1003	没有发现捆绑列表	指定的捆绑列表名不存在
-1004	没有发现表格模板	指定的表格模板名不存在
-1005	内部错误	请求技术支持
-1006	字符串为空	警告：从数据库中读入的字符串为空

-1007	字符串 is truncated	警告：读入的字符串超过 131 个字符 and is truncated on a select
-1008	没有 where 语句	删除时没有 where 语句
-1009	建立连接失败	
-1010	连接指令中 “DB=” 指定的数据库不存在	
-1011	没有选定行	SQLNumRows(), SQLFirst(), SQLNext(), 或 SQLPrev() 指令在执行 SQLSelect() 指令之前被调用
-4149	你提供连接, 语句, 或查询句柄无效	列的类型可能设置不正确。例如：对于一个 dBASE 文件，如果一个表格模板的列项是 character 而不是 char，就会出错。

特定数据库的错误信息：

1. Oracle

错误信息	解决方法
------	------

ORA-03112-Host String Syntax error	如果你没有运行 NETINIT.EXE, 把它放到 Windows 的启动组中。如果你正运行 NETINIT.EXE 并想要建立多个连接, 你需要分配内存。通过在 CONFIG.ORG 文件中设定 WIN_REMOTE_SESSIONS 参数可以实现。例如要为四个连接分配内存: WIN_REMOTE_SESSIONS=4
ORA-3121-No interface driver connected	在进入 Windows 之前启动 SQL*NET TSR
ORA-6435-NetBIOS:Unable to add local name to name table	必须运行网络软件 (Novell, LAN Manager, 等.)
ORA-09301-Local kernel only supported in standard name table	在连接字符串中定义服务器名 (“SRVR= ”)
ORA-06430-Unable to make connection	检查连接字符串中属性是否正确

2.Sybase or Microsoft SQL Sever

出错信息	解决方法
You can not have more than one statement active at a time	使用 SQLEnd() 释放系统资源, 或对第二个语句使用不同的 ConnectionID
There is not enough memory available to process the command	尝试重启客户机
Invalid object name table name	你使用的数据库中不存在此表格名

3. DBASE

出错信息	解决方法
File or DLL not found	对 Windows 来说, QEDBF. DLL 必须在当前目录或 \windows\system\ 目录下。
Invalid connection	确认在你的目录下已经有 DLLS。为了 DBF 支持, 你需要 QLDBF. DLL
The connection or statement	组态王语句中可能出现索引错误

handel you provided is not valid	
-------------------------------------	--

19.3.7 如何将数据库里的数据读到组态王中来

使用组态王数据库访问功能，除了可以直观的将数据显示在画面中外，还可以将数据库中的数据显到趋势曲线上和数据表格中。组态王提供一个历史趋势曲线控件将数据库中的数据以趋势曲线的形式显示出来。具体操作请参见“历史趋势曲线控件”一节。组态王还提供一个访问开放型数据库显示控件将数据库中的数据以列表的形式显示出来。具体操作请参见“开放型数据库访问控件”一节。

第二十章 OPC 设备

- ☞ 介绍 OPC 服务器的相关知识
- ☞ 介绍组态王 7.0 与 OPC 的连接及使用
- ☞ 举例说明在 OPC 服务器中定义数据项

20.1 OPC 简介

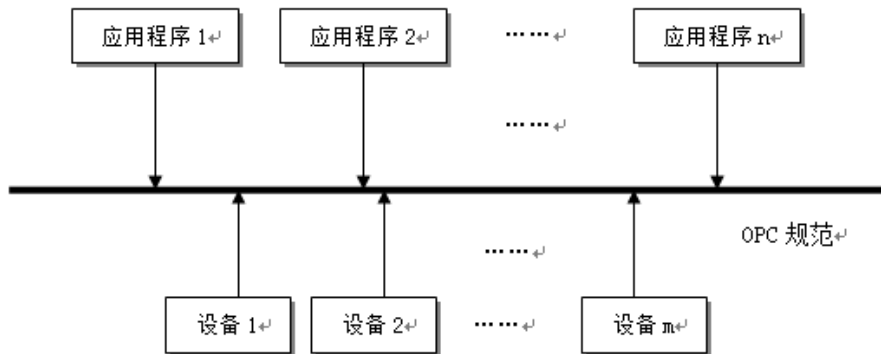
20.1.1 什么是 OPC

OPC 是 OLE for Process Control 的缩写，即把 OLE 应用于工业控制领域。OLE 原意是对象链接和嵌入，随着 OLE 2 的发行，其范围已远远超出了这个概念。现在的 OLE 包容了许多新的特征，如统一数据传输、结构化存储和自动化，已经成为独立于计算机语言、操作系统甚至硬件平台的一种规范，是面向对象程序设计概念的进一步推广。OPC 建立 OLE 规范之上，它为工业控制领域提供了一种标准的数据访问机制。

工业控制领域用到大量的现场设备，在 OPC 出现以前，软件开发商需要开发大量的驱动程序来连接这些设备。即使硬件供应商在硬件上做了一些小小改动，应用程序就可能需要重写；同时，由于不同设备甚至同一设备不同单元的驱动程序也有可能不同，软件开发

商很难同时对这些设备进行访问以优化操作。硬件供应商也在尝试解决这个问题，然而由于不同客户有着不同的需要，同时也存在着不同的数据传输协议，因此也一直没有完整的解决方案。

自 OPC 提出以后，这个问题终于得到解决。OPC 规范包括 OPC 服务器和 OPC 客户两个部分，其实质是在硬件供应商和软件开发商之间建立了一套完整的“规则”，只要遵循这套规则，数据交互对两者来说都是透明的，硬件供应商无需考虑应用程序的多种需求和传输协议，软件开发商也无需了解硬件的实质和操作过程。



OPC 的优越性是显而易见的：

- ✧ 硬件供应商只需提供一套符合 OPC Server 规范的程序组，无需考虑工程人员需求。
- ✧ 软件开发商无需重写大量的设备驱动程序。
- ✧ 工程人员在设备选型上有了更多的选择。
- ✧ OPC 扩展了设备的概念。只要符合 OPC 服务器的规范，OPC 客户都可与之进行数据交互，而无需了解设备究竟是 PLC 还是仪表，甚至在数据库系统上建立了 OPC 规范，

OPC 客户也可与之方便地实现数据交互。

20.1.2 OPC 的适用范围

OPC 设计者们最终目标是在工业领域建立一套数据传输规范，并为之制定了一系列的发展计划。现有的 OPC 规范涉及如下领域：

- ◇ 在线数据监测。实现了应用程序和工业控制设备之间高效、灵活的数据读写。
- ◇ 报警和事件处理。提供了 OPC 服务器发生异常时，以及 OPC 服务器设定事件到来时向 OPC 客户发送通知的一种机制。
- ◇ 历史数据访问。实现了读取、操作、编辑历史数据库的方法。
- ◇ 远程数据访问。借助 Microsoft 的 DCOM 技术，OPC 实现了高性能远程数据访问能力。
- ◇ OPC 近期将实现的功能还包括安全性、批处理、历史报警事件数据访问等。

OPC 的设计者在设计 OPC 时遵循如下原则：

- ◇ 易于实现。
- ◇ 灵活满足多种客户需求。
- ◇ 强大的功能。
- ◇ 高效的操作。

20.1.3 OPC 的基本概念

1、服务器、组、数据项

OPC 服务器由三类对象组成：服务器（Server）、组（Group）、数据项（Item）。

◇ 服务器对象 (Server)

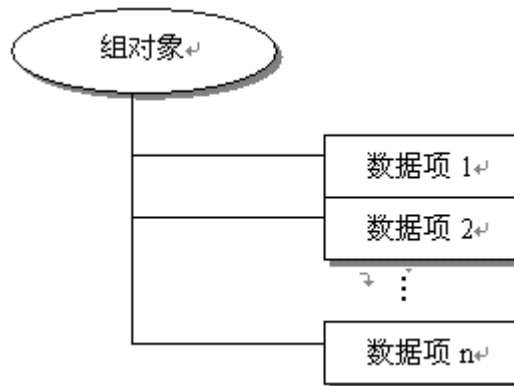
拥有服务器的所有信息，同时也是组对象 (Group) 的容器。

◇ 组对象 (Group)

拥有本组的所有信息，同时包容并逻辑组织 OPC 数据项 (Item)。OPC 组对象 (Group) 提供了客户组织数据的一种方法。客户可对之进行读写，还可设置客户端的数据更新速率。当服务器缓冲区内数据发生改变时，OPC 将向客户发出通知，客户得到通知后再进行必要的处理，而无需浪费大量的时间进行查询。OPC 规范定义了两种组对象：公共组和局部组 (私有组)。公共组由多个客户共有，局部组只隶属于一个 OPC 客户。一般说来，客户和服务器的一个连接只需定义一个组对象。

◇ 数据项 (Item)

在每个组对象中，客户可以加入多个 OPC 数据项 (Item)。如图所示。



OPC 数据项是服务器端定义的对象，通常指向设备的一个寄存器单元。OPC 客户对设备寄存器的操作都是通过其数据项来完成的，通过定义数据项，OPC 规范尽可能的隐

藏了设备的特殊信息，也使 OPC 服务器的通用性大大增强。OPC 数据项并不提供对外接口，客户不能直接对之进行操作，所有操作都是通过组对象进行的。客户操作数据项的一般步骤为：

- 通过服务器对象接口枚举服务器端定义的所有数据项，如果客户对服务器所定义的数据项非常熟悉，此步可以忽略。
- 将要操作的数据项加入客户定义的组对象中。
- 通过组对象对数据项进行读写等操作。

每个数据项的数据结构包括三个成员变量：即数据值、数据质量和时间戳。数据值是以 VARIANT 形式表示的。应当注意，数据项表示同数据源的连接而不等同于数据源，无论客户是否定义数据项，数据源都是客观存在的。可以把数据项看作数据源的地址，即数据源的引用，而不应看作数据源本身。

2、报警 (Alarm) 和事件 (Event)

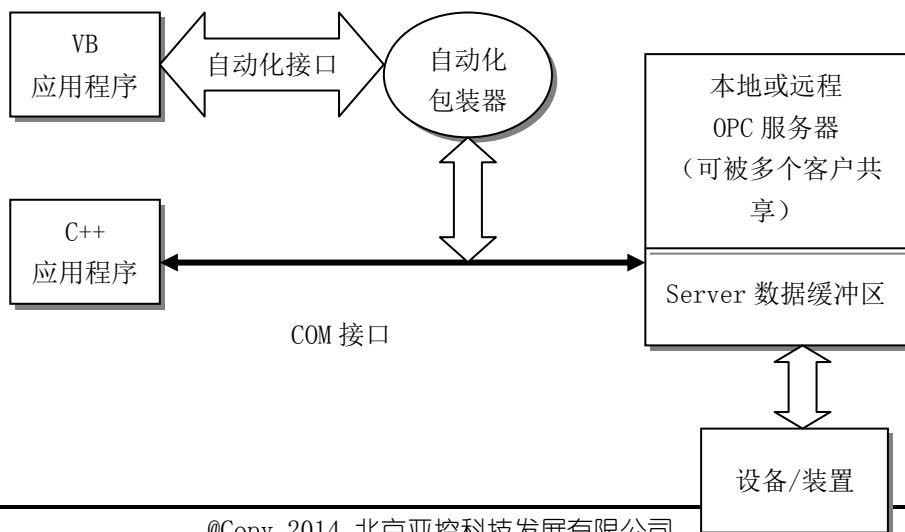
报警和事件处理机制增强了 OPC 客户处理异常的能力。服务器在工作过程中可能出现异常，此时，OPC 客户可通过报警和事件处理接口得到通知，并能通过该接口获得服务器的当前状态。在很多场合，报警 (Alarm) 和事件 (Event) 的含义并不加以区分，两者也经常互换使用。从严格意义上讲，两者含义略有差别。

◇ 依据 OPC 规范，报警是一种异常状态，是 OPC 服务器或服务器的一个对象可能出现的所有状态中的一种特殊情况。例如，服务器上标记为 FC101 的一个单元可能有如下状态：高出警戒，严重高出警戒，正常，低于警戒，严重低于警戒。除了正常状态外，其他状态都视为报警状态。

◇ 事件则是一种可以检测到的出现的情况，这种情况或来自 OPC 客户，或来自 OPC 服务器，也可能来自 OPC 服务器所代表的设备，通常都有一定的物理意义。事件可能与服务器或服务器的一个对象的状态有关，也可能毫无关系。如高出警戒和正常状态的转换事件和服务器的某个对象的状态有关，而操作设备，改变系统配置以及出现系统错误等事件和对象状态就无任何关系。

3、OPC 体系结构

OPC 规范提供了两套接口方案，即 COM 接口和自动化。COM 接口效率高，通过该接口，客户能够发挥 OPC 服务器的最佳性能，采用 C++ 语言的客户一般采用 COM 接口方案；自动化接口使解释性语言和宏语言访问 OPC 服务器成为可能，采用 VB 语言的客户一般采用自动化接口。自动化接口使解释性语言和宏语言编写客户应用程序变得简单，然而自动化客户运行时需进行类型检查，这一点则大大牺牲了程序的运行速度。OPC 服务器必须实现 COM 接口，是否实现自动化接口则取决于供应商的主观意愿。典型的 OPC 体系如图。



◇ 服务器缓冲区数据和设备数据

OPC 服务器本身就是一个可执行程序, 该程序以设定的速率不断地同物理设备进行数据交互。服务器内有一个数据缓冲区, 其中存有最新的数据值, 数据质量戳和时间戳。时间戳表明服务器最近一次从设备读取数据的时间。服务器对设备寄存器的读取是不断进行的, 时间戳也在不断更新。即使数据值和质量戳都没有发生变化, 时间戳也会进行更新。

客户既可从服务器缓冲区读取数据, 也可直接从设备读取数据, 从设备直接读取数据速度会慢一些, 一般只有在故障诊断或极特殊的情况下才会采用。

◇ 同步和异步

OPC 客户和 OPC 服务器进行数据交互可以有两种不同方式, 即同步方式和异步方式。同步方式实现较为简单, 当客户数目较少而且同服务器交互的数据量也比较少的时候可以采用这种方式; 异步方式实现较为复杂, 需要在客户程序中实现服务器回调函数。然而当有大量客户和大量数据交互时, 异步方式能提供高效的性能, 尽量避免阻塞客户数据请求, 并最大可能地节省 CPU 和网络资源。

20.1.4 组态王 7.0 与 OPC

组态王 7.0 充分利用了 OPC 服务器的强大性能, 为工程人员提供方便高效的数据访问能力。在组态王 7.0 中可以同时挂接任意多个 OPC 服务器, 每个 OPC 服务器都被作为一个外部设备, 工程人员可以定义、增加或删除它, 如同一个 PLC 或仪表设备一样。

工程人员在 OPC 服务器中定义通讯的物理参数, 定义需要采集的下位机变量 (称为数

据项，详见下文解释)；然后在组态王 7.0 中定义组态王 7.0 变量和下位机变量（数据项）的对应关系。在运行系统中，组态王 7.0 和每个 OPC 服务器建立连接，自动完成和 OPC 服务器之间的数据交换。

同时，组态王 7.0 本身也可以充当 OPC 服务器，向其他符合 OPC 规范的厂商的控制系统提供数据。

在作为 OPC 服务器的组态王 7.0 中定义相关的变量并和采集数据的硬件进行连接；然后在充当客户端的其他应用程序中与 OPC 服务器（组态王 7.0 运行系统）建立连接，并且添加数据项目。在应用程序运行时，客户端将按照指定的采集频率对组态王 7.0 的数据进行采集。组态王 7.0 的 OPC 服务器名称为“KingView.View.1”。

20.2 组态王 7.0 作为 OPC 客户端的使用方法

20.2.1 建立和删除 OPC 设备

20.2.1.1 建立 OPC 设备

组态王 7.0 中支持多 OPC 服务器。在使用 OPC 服务器之前，需要先在组态王 7.0 中建立 OPC 服务器设备。

如图 20.1 所示，在组态王 7.0 工程浏览器的“设备”项目中选中“OPC 服务器”，工程浏览器的右侧内容区显示当前工程中定义的 OPC 设备和“新建 OPC”图标。

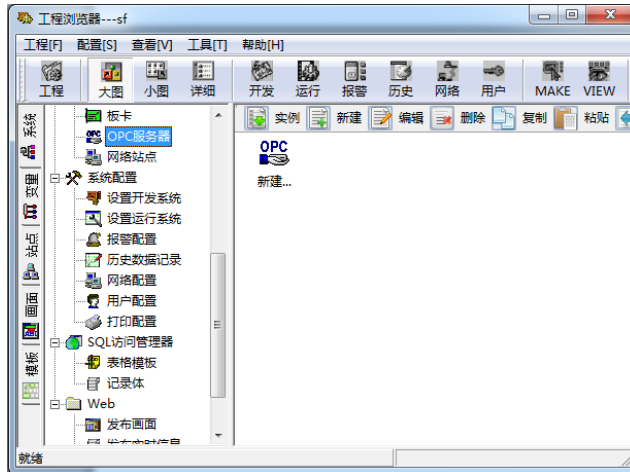


图 20.1 OPC 设备

双击“新建”图标，组态王 7.0 开始自动搜索当前的计算机系统中已经安装的所有 OPC 服务器，然后弹出“查看 OPC 服务器”对话框，如图 20.2 所示。



图20.2 OPC 设备定义

- ◇ “网络节点名”编辑框中为要查看 OPC 服务器的计算机名称，默认为“本机”。如果需要查看网络上的其它站点的 OPC 服务器，在编辑框中输入节点的 UNC 路径。如计算机名称为“数据采集站”，则输入[\\数据采集站](#)。
- ◇ 单击“查找”按钮，如果查找成功，则在右边的“OPC 服务器”列表中显示目标站点的所有已安装的 OPC 服务器名称；如果没有查找到，则提示查找失败。
- ◇ “OPC 服务器信息”文本框中显示“OPC 服务器”列表中选中的 OPC 服务器的相关说明信息。如选中“KingView.View.1”，则在信息中显示“KingView.View”。
- ◇ “读写方式”是用来定义该 OPC 设备对应的 OPC 变量在进行读写数据时采用同步或异步方式。

- ◇ “尝试恢复间隔”和“最长恢复时间”用来设置当组态王 7.0 与 OPC 服务器之间的通讯出现故障时，系统尝试恢复通讯的策略参数。
- ◇ “订阅频率”，目前组态王 7.0 异步读是按照订阅的方式实现的，可以根据不同的需要进行设置。
- ◇ “使用动态优化”是组态王 7.0 对通讯过程采取动态管理的办法。

“尝试恢复间隔”，“最长恢复时间”，“使用动态优化”的具体含义与 I/O 设备定义向导中的相同，请参看第六章“IO 设备管理”中的 6.1.3.3.1 章节。

用户可以在列表中选择所需的 OPC 服务器。单击“确定”按钮。“查看 OPC 服务器”对话框自动关闭，OPC 设备建立成功。如选择图中的“PCSoft.Sample.1”，建立的 OPC 设备如图 20.3 所示。

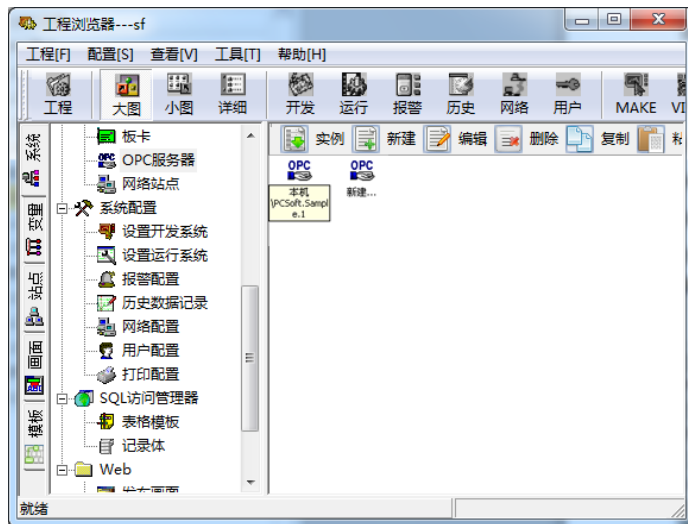


图 20.3 OPC 服务器的建立

20.2.1.2 删除 OPC 设备

对于已经建立的 OPC 设备，如果您确认不再需要，可以将它删除。如图 20.3，选中要删除的 OPC 设备，单击鼠标右键，在弹出的如图 20.4 所示的快捷菜单中选择删除，弹出如图 20.5 所示的提示信息，如果选择“是”，则将该设备从组态王 7.0 中删除。



图 20.4 快捷菜单

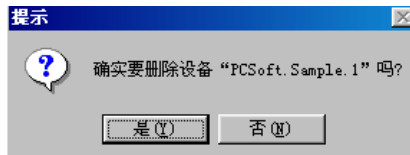


图 20.5 删除确认提示信息

20.2.2 在 OPC 服务器中定义数据项

OPC 服务器作为一个独立的应用程序，可能由硬件制造商、软件开发商或其他第三方提供，因此数据项定义的方法和界面都可能有所差异。下面以 PC Soft 公司的 Modbus Server 为例讲解 OPC Server 的使用方法。Modbus OPC Server 应用程序是一个高级的 I/O

服务器，提供友好的工程人员界面，支持 DDE，AdvanceDDE 和 FastDDE 等数据访问方式。

双击程序组 Modbus Server 图标，弹出 Modbus Server 主窗口如图 20.6 所示。

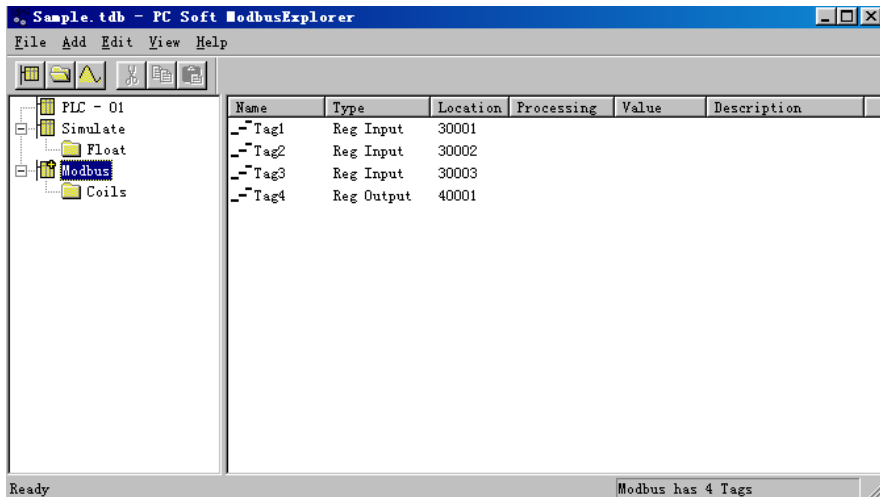



图 20.6 在 OPC 服务器中定义数据项

Modbus Server 主窗口和 Windows 资源管理器风格相似，窗口的上方是菜单和工具条，窗口的左侧列出接入的两个设备，即 Simulate 和 Modbus，每个设备都包括自己的组对象和数据项。窗口的右侧显示出所选对象中已定义的数据项。

◇ 可以向 Modbus Server 中添加、删除设备或修改设备属性。选择“Add”菜单下“New Device...”子菜单，或单击工具条按钮 ，弹出“设备属性对话框”如图 20.7 所示。

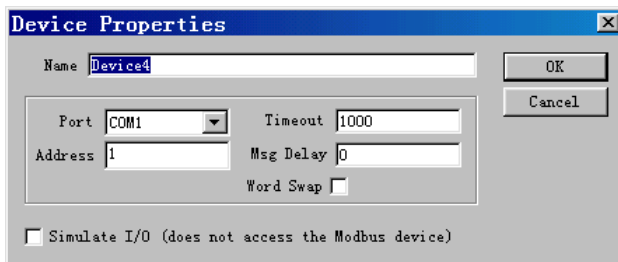


图 20.7 在 OPC 服务器中增加外围设备

对话框最下方是一个 Simulate I/O（仿真 I/O）复选框，不选该选框，指示新加入的设备连接一个真实的物理设备；选中该框指示新加入的设备并未和真实物理设备建立连接，所得到的数据只是仿真结果。


- ◇ 可以向一个设备中直接加入数据项，也可在设备中创建若干个组，将具有相似功能的数据项组织起来。组的下面也可以创建组，层次的多少原则上没有限制，然而从方便起见，层次不宜太多。设备、组、数据项的概念和 Windows 文件系统的驱动器、文件夹、文件的概念很相似。选择“Add”菜单下“New Group...”子菜单，或单击工具按键  弹出“组对话框”如图 20.8 所示。



图 20.8 在 OPC 服务器中增加组

选择“Add”菜单下“New Tag...”子菜单，或单击工具条按键弹出“数据项属性对话框”如图 20.9 所示。

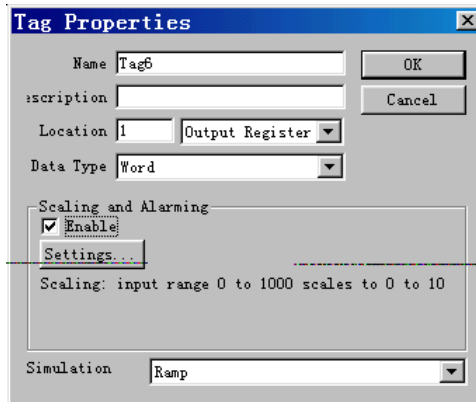


图 20.9 在 OPC 服务器中增加数据项

选中“Scaling and Alarming”组框中“Enable”复选框，单击“Setting...”按钮，弹出“数据项设置对话框”如图 20.10 所示。

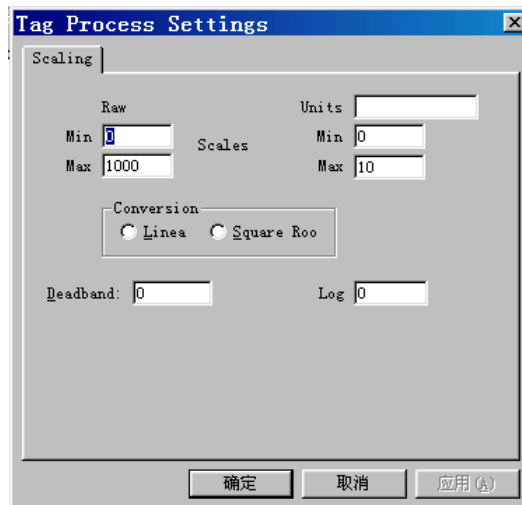


图 20.10 数据项设置对话框

填入相应数值后，按确定返回主窗口即可。

20.2.3 OPC 服务器与组态王 7.0 数据词典的连接

OPC 服务器与组态王 7.0 数据词典的连接如同 PLC 或板卡等外围设备与组态王 7.0 数据词典的连接一样。

在组态王 7.0 工程浏览器中，选中数据词典，在工程浏览器右侧双击新建图标，选择 I/O 类型变量，在连接设备处选择 OPC 服务器，如图 20.11 所示。



图 20.11 OPC 服务器与组态王 7.0 数据词典的连接



注意，

如果 OPC 服务器没有事先启动的话，此时系统会自动启动 OPC 服务器

在寄存器下拉式菜单中列出了在 OPC 服务器中定义过的所有项目名及数据项，项目名称和数据项以树型结构排列，如果某个分支下还有项目的话，鼠标双击该分支，隐藏在分支下的数据项会自动列出来。鼠标双击选择对应的数据项，则选择的数据项会自动添加到“寄存器”中。如图 20.12 所示，选择的数据项是“Modbus”下的“Tag1”。

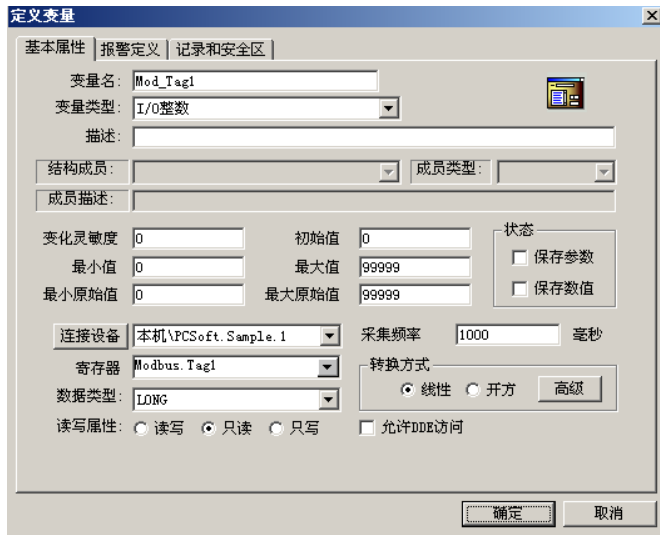


图 20.12 定义 OPC 服务器的变量

在组态王 7.0 中新建一个画面，画面上创建一个文本图素，定义图素的动画连接为“模拟值输出”，连接的变量为刚定义的“Mod_Tag1”。保存画面，切换到组态王 7.0 运行系统，打开画面，可以看到组态王 7.0 与 OPC 服务器间的数据交换，如图 20.13 所示。

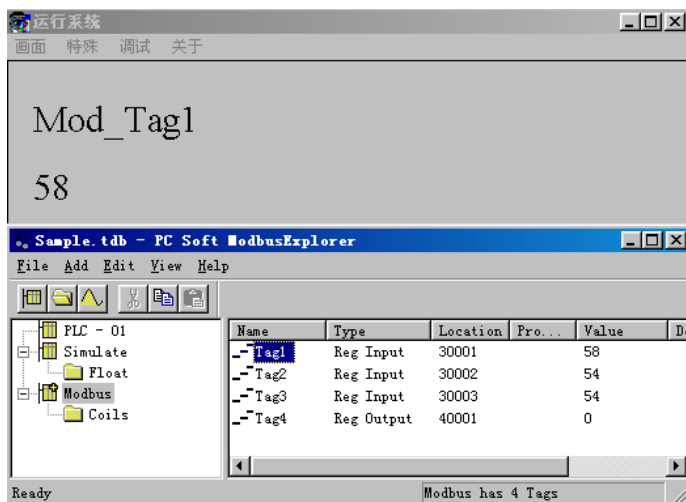


图20.13 组态王7.0 与 OPC 服务器间的数据交换

组态王 7.0 可以和各种标准 OPC 服务器之间进行数据交换。

20.3 组态王 7.0 作为 OPC 服务器的使用

组态王 7.0 在原有的 OPC 客户端的基础上添加了 OPC 服务器的功能,实现了组态王 7.0 对 OPC 的服务器和客户端的统一。通过组态王 7.0 的 OPC 服务器功能,用户可以更方便的实现其他支持 OPC 客户的应用程序与组态王 7.0 之间的数据通讯和调用。

20.3.1 组态王 7.0 OPC 服务器的功能简介

◇ OPC 通讯技术上的实现:

OPC 实现的是系统中进程间的通讯,其采用 COM 和 DCOM 技术实现了接口,调用非常方便。

◇ OPC 数据的通讯

OPC 之间的通讯是以变量为单位的，在 OPC 服务器上定义相关的变量和要采集的硬件进行连接，并生成唯一表示此变量的 ID 标识。此变量中保存着变量的数值，变量相关的信息，外部的程序能够访问的就是此变量的所有信息，即 OPC 服务器与外部的数据的传输是通过变量进行对应的。组态王 7.0 作为服务器其所有变量都可以被外部支持 OPC 的客户端进行访问，访问的对象是变量或变量的域。而且对于可读写变量的可修改的域，用户可以通过对组态王 7.0 OPC 服务器的访问得到相应的数值并能够修改相应的数值。另外，为了方便用户对组态王 7.0 OPC 功能的使用，组态王 7.0 提供了 OPC 客户端接口开发包。该接口支持 VB、VC 等编程语言，用户可以很方便的使用该动态库访问组态王 7.0 的实时数据。

20.3.2 组态王 7.0 OPC 服务器的使用

OPC 客户作为一个独立的应用程序，可能由硬件制造商、软件开发商或其他第三方提供，因此数据项定义的方法和界面都可能有所差异。下面以 FactorySoft 的 OPC 客户端为例说明组态王 7.0 OPC 服务器的使用。

- 1、启动组态王 7.0 的运行系统（组态王 7.0 的 OPC 服务器是指组态王 7.0 的运行系统）。
- 2、运行某些厂家提供的 OPC 客户端，弹出画面如图 20.14 所示。

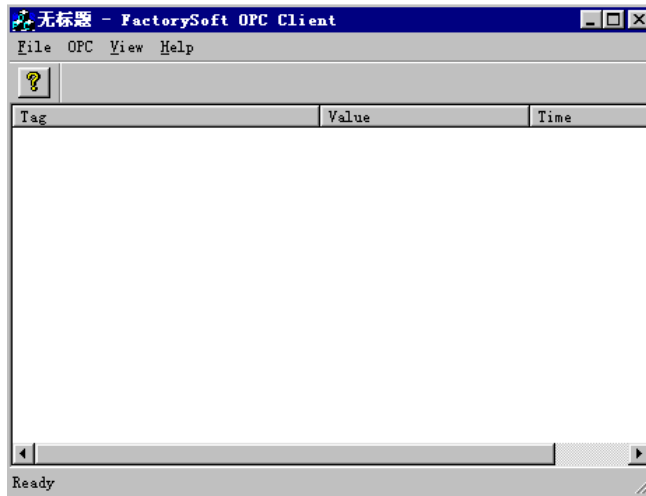


图20.14 Factory Soft OPC 客户端

3、选择界面“OPC”菜单的CONNECT选项，弹出连接服务器选项画面，如图20.15所示。

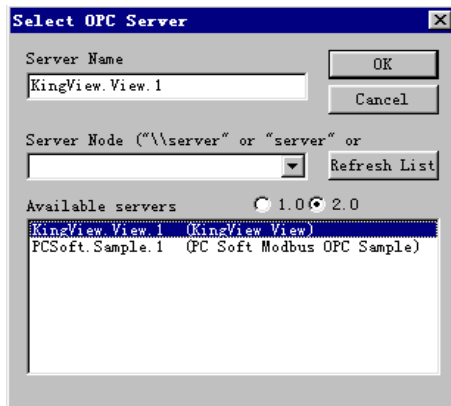


图20.15 选择OPC服务器

4、组态王7.0的OPC服务器标志是KingView.View.1(KingView.View)，用户选择此选项并点击<确定>按钮完成客户端与服务器的连接。(如果用户事先没有启动组态王

7.0 运行系统，此时将自动启动组态王 7.0。)

- 5、在客户端界面菜单中点击“OPC”菜单下的 ADDITEM 选项，弹出填加项目画面，如图 20.16 所示，在变量浏览列表中列出了组态王 7.0 的所有变量数据项。(OPC 客户端的具体使用方法因厂家不同而不同，使用时请参见厂家说明。)

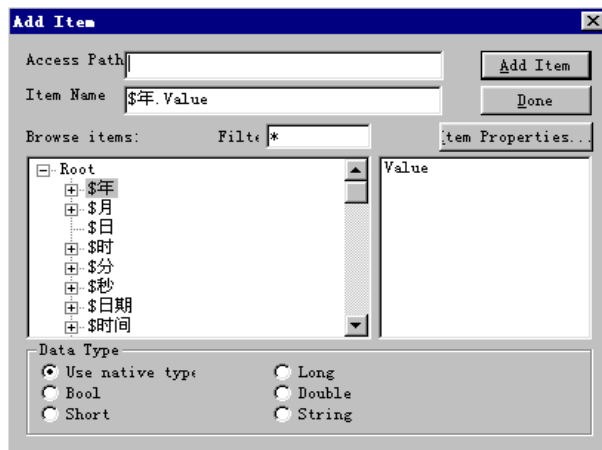
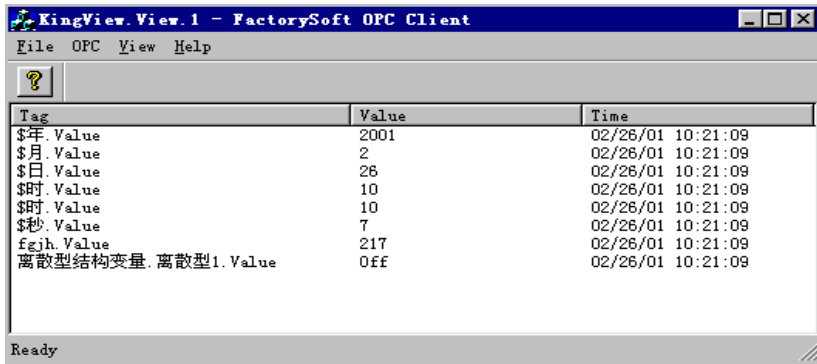


图 20.16 向客户端中加入组态王 7.0 变量

- 6、一旦在客户端中加入了组态王 7.0 的变量，客户端便按照给定的采集频率对组态王 7.0 的数据进行采集。如图 20.17 所示。




Tag	Value	Time
\$年.Value	2001	02/26/01 10:21:09
\$月.Value	2	02/26/01 10:21:09
\$日.Value	26	02/26/01 10:21:09
\$时.Value	10	02/26/01 10:21:09
\$分.Value	10	02/26/01 10:21:09
\$秒.Value	7	02/26/01 10:21:09
fgjh.Value	217	02/26/01 10:21:09
离散型结构变量.离散型1.Value	Off	02/26/01 10:21:09

图 20.17 客户端的数据采集

7、选择菜单“OPC”下的“WrightItem”项，可以对可读写变量的可读写的域进行修改。

20.3.3 组态王 7.0 为用户提供的 OPC 接口

为了方便用户使用组态王 7.0 的 OPC 服务器功能,使用户无需在无其他需求的情况下再购买其它的 OPC 客户端,组态王 7.0 提供了一整套与组态王 7.0 的 OPC 服务器连接的函数接口,这些函数可通过提供的动态库 KingviewCliend.dll 来实现。用户使用该动态库可以自行用 VB、VC 等编程语言编制组态王 7.0 的 OPC 客户端程序。

 注意:

该接口有详细的例程和说明文档,称为“SDK 开发包”,包含在亚控的工具开发包之中。请与亚控公司的技术支持或销售人员联系。

20.4 如何使用网络 OPC 通讯

组态王 7.0 支持网络 OPC 功能,组态王 7.0 与组态王 7.0 之间可以通过网络以 OPC

方式进行通讯，同样其他 OPCclient/OPCserver 也可以通过网络与组态王 7.0 之间以 OPC 方式进行通讯。

组态王 7.0 作为 OPC server 时只能在 NT/2000/XP 上使用。具体操作将在下面讲述。

20.4.1 使用网络 OPC 前的 DCOM 配置

在使用网络 OPC 模式前，需要使用 DCOM 配置工具对系统进行配置。本节主要以组态王 7.0 之间的互联来介绍，其它 OPC 程序连接方法相同。配置的方法和步骤如下。

20.4.1.1 配置充当 OPC 服务器的机器

配置充当 OPC 服务器的机器是指对 dcomcnfg 程序进行配置，win2000/winNT 以上操作系统自己带有配置程序——dcomcnfg 程序，设置过程如下：

1、运行 dcomcnfg

在 windows “开始” 菜单中选择 “运行”，在编辑框中输入 “dcomcnfg”，如下图 20.18 所示。

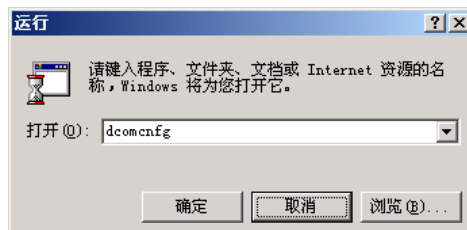


图 20.18 运行中输入命令

2、定义属性

点击“确定”后，弹出“分布式 com 配置属性”对话框，如下图 20.19 所示。

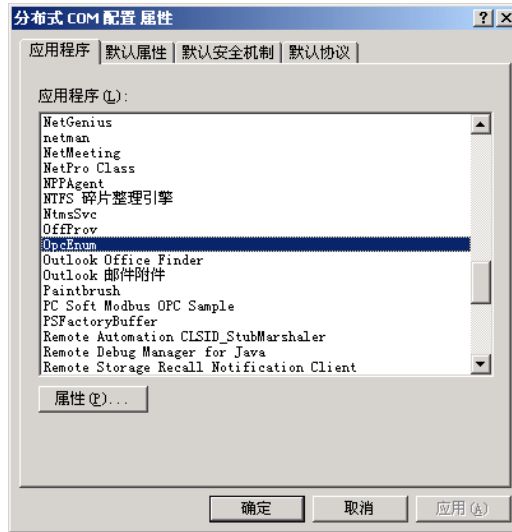


图 20.19 com 配置属性

- 点击“默认安全机制”属性页标签，进行定义，如下图 20.20 所示。

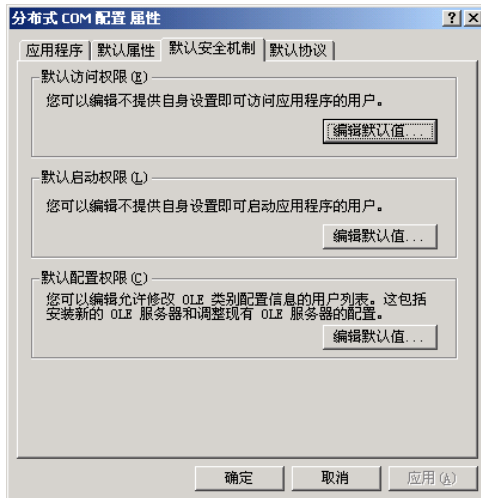


图 20.20 设置默认安全机制

- 对“默认访问权限”，“默认启动权限”和“默认配置权限”进行设置，添加“everyone”用户，并将其权限分别设置为“允许访问”，“允许调用”和“完全控制”。
- 在“应用程序”属性页的列表中选中“opcEnum”，单击“属性”按钮，弹出对话框，如下图 20.21 所示。

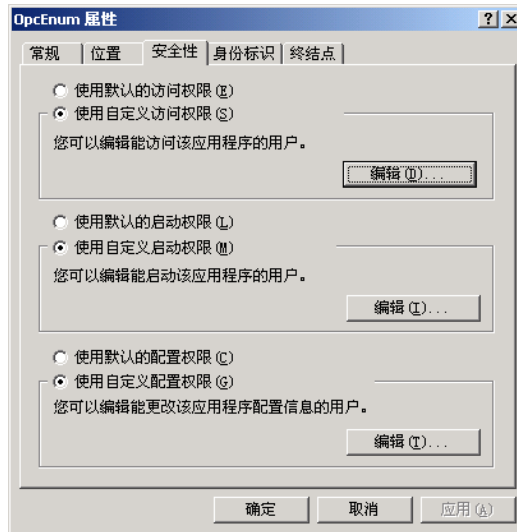


图20.21 设置opcEnum 属性

- 在“安全性”属性页中选中“使用自定义访问权限”，“使用自定义启动权限”，“和使用自定义配置权限”，并分别进行编辑，添加“everyone”用户，并将其权限分别设置为“允许访问”，“允许调用”和“完全控制”。
- 然后在“身份标识”属性页中选中“交互式用户”，如图 20.22 所示。

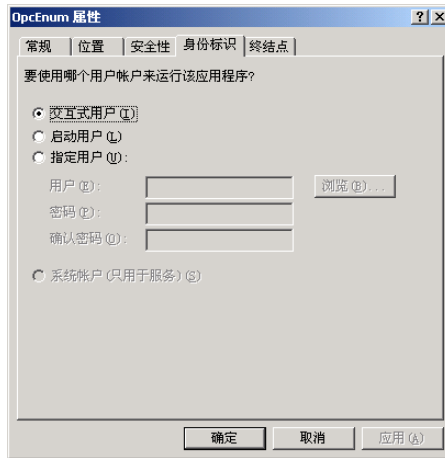


图 20.22 设置身份标识属性

3、单击“确定”按钮，回到“分布式 com 配置属性”对话框中，选中 kingview.view，如下图所示 20.23 所示。



图 20.23 分布式 com 配置属性页

单击“属性”按钮，进行属性配置。同样，在“安全性”属性页中选中“使用自定义访问权限”，“使用自定义启动权限”，“和使用自定义配置权限”，并分别进行编辑，添加“everyone”用户，并将其权限分别设置为“允许访问”，“允许调用”和“完全控制”。然后在“身份标识”属性页中选中“交互式用户”。

4、服务器端定义完成后，在“分布式 COM 配置属性”对话框上单击“确定”按钮关闭对话框。进入组态王 7.0 运行系统，此时组态王 7.0 作为 OPC 服务器。

20.4.1.2 客户端通过 OPC 连接服务器

客户端组态王 7.0 作为 OPC 客户端，可以通过网络 OPC 功能与组态王 7.0OPC 服务器连接。客户端的操作如下：

1. 定义 OPC 服务器：

在工程浏览器中，选择“OPC 服务器”，然后双击“新建”，弹出“查看 OPC 服务器”在“网络节点名”中输入服务器的机器节点名，例如运行组态王 7.0 的服务器为 test，则输入\\test，点击“查找”后，列表中会列出 test 机器上所有的 OPC 服务程序，选中 kingview.view.1，然后点“确认”，OPC 服务器就定义好了，如下图 20.24 所示。



图20.24 定义好的OPC 服务器

2. 客户端定义变量

在客户端定义变量与组态王 7.0OPC 服务器上的变量建立连接。例如定义 test，连接设备中选择刚才定义的 OPC 服务器“kingivew.view.1，在“寄存器”选项中弹出远程站点上的变量列表，选择相应变量的域。例如选择“a.value”，如下图 20.25 所示。

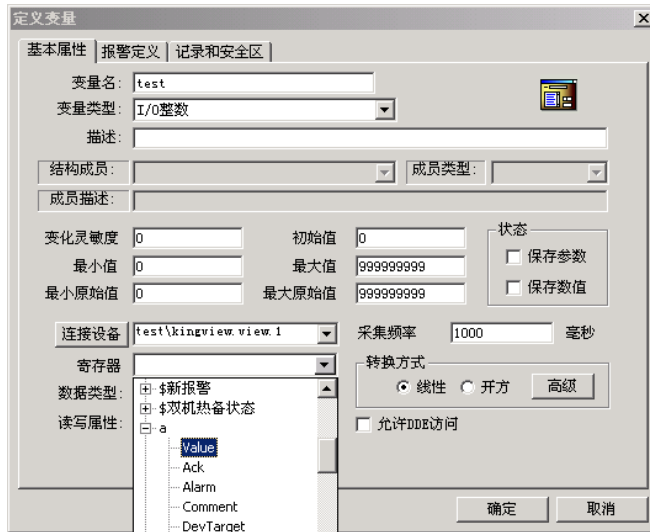



图 20.25 定义客户端变量

启动客户端运行系统，就实现了组态王 7.0 通过网络 OPC 交换数据。

 注意：

关于 DCOM 配置的具体内容用户可以参照 Windows 关于 DCOMCNFG 的帮助。

第二十一章 网络功能

- ☞ 介绍组态王 7.0 的网络功能
- ☞ 教您如何用组态王 7.0 实现“服务器”与“客户”连接
- ☞ 教您如何引用网络变量、使用分布式历史数据库、分布式报警系统和网络登录管理

21.1 组态王 7.0 网络结构概述

组态王 7.0 基于网络的概念，是一种真正的客户—服务器模式，支持分布式历史数据库和分布式报警系统，可运行在基于 TCP/IP 网络协议的网上，使用户能够实现上、下位机以及更高层次的厂级连网。

TCP/IP 网络协议提供了在不同硬件体系结构和操作系统的计算机组成的网络上进行通信的能力。一台 PC 机通过 TCP/IP 网络协议可以和多个远程计算机(即远程节点)进行通讯。

组态王 7.0 的网络结构是一种柔性结构，可以将整个应用程序分配给多个服务器，可以引用远程站点的变量到本地使用（显示、计算等），这样可以提高项目的整体容量结构并改善系统的性能。服务器的分配可以是基于项目中物理设备结构或不同的功能，用户可以根据系统需要设立专门的 IO 服务器、历史数据服务器、报警服务器、登录服务器和 WEB 服务器等。本章分五节分别介绍各个功能的配置和实现。下面先介绍一下这五种服

务器的含义：

◇ I/O 服务器：

负责进行数据采集的站点，一旦某个站点被定义为 I/O 服务器，该站点便负责数据的采集。如果某个站点虽然连接了设备，但没有定义其为 I/O 服务器，那这个站点的数
据照样进行采集，只是不向网络上发布。I/O 服务器可以按照需要设置为一个或多个。

◇ 报警服务器：

存储报警信息的站点，一旦某个站点被指定为一个或多个 I/O 服务器的报警服务器，
系统运行时，I/O 服务器上产生的报警信息将通过网络传输到指定的报警服务器上，
经报警服务器验证后，产生和记录报警信息。报警服务器可以按照需要设置为一个
或多个。报警服务器上的报警组配置应当是报警服务器和与其相关的 I/O 服务器上
报警组的合集。如果一个 I/O 服务器不作为报警服务器，系统中也没有报警服务器，
系统运行时，该 I/O 服务器的报警窗上不会看到报警信息。

◇ 历史记录服务器：

与报警服务器相同，一旦某个站点被指定为一个或多个 I/O 服务器的历史数据服务器，
系统运行时，I/O 服务器上需要记录的历史数据便被传送到历史数据服务器站点上，
保存起来。对于一个系统网络来说，建议用户只定义一个历史数据服务器，否则会
出现客户端查不到历史数据的现象。

◇ 登录服务器：

登录服务器在整个系统网络中是**唯一**的。它拥有网络中唯一的用户列表。所以用户
应该在登录服务器上建立最完整的用户列表，并保证客户端上的用户列表与登录服

务器上的用户列表保持一致。当用户在网络的任何一个站点上登录时，系统调用该用户列表，登录信息被传送到登录服务器上，经验证后，产生登录事件。然后，登录事件将被传送到该登录服务器的报警服务器上保存和显示。这样，保证了整个系统的安全性。另外，系统网络中工作站的启动、退出事件也被先传送到登录服务器上验证，然后传到该登录服务器的报警服务器上保存和显示。

◇ WEB 服务器:

WEB 服务器是运行组态王 7.0WEB 版本、保存组态王 7.0For Internet 版本发布文件的站点，传送文件所需数据，并为用户提供浏览服务的站点。详细内容请见 21.7 节。

◇ 客户:

如果某个站点被指定为客户，可以访问其指定的 IO 服务器、报警服务器、历史数据服务器上的数据。一个站点被定义为服务器的同时，也可以被指定为其它服务器的客户。

一个工作站站点可以充当多种服务器功能，如 I/O 服务器可以被同时指定为报警服务器、历史数据服务器、登录服务器等。报警服务器可同时作为历史数据服务器、登录服务器等。

除了上述几种服务器和客户机之外，组态王 7.0 为了保持网络中时钟的一致，还可以定义“校时服务器”，校时服务器按照指定的时间间隔向网络发送校时帧，以统一网络上个站点的系统时间。

 注意:

工程人员要实现“组态王 7.0”的网络功能，必须满足以下条件：

1、将“组态王 7.0”安装在网络版 Windows NT 或 Windows/XP 上，并在配置网络时绑定 TCP/IP 协议，即利用“组态王 7.0”网络功能的 PC 机必须首先是某个局域网上的站点并启动该网。

2、客户机和服务器必须安装并同时运行“组态王 7.0”(除 Internet 版本的客户端)。

网络结构示意图如下：

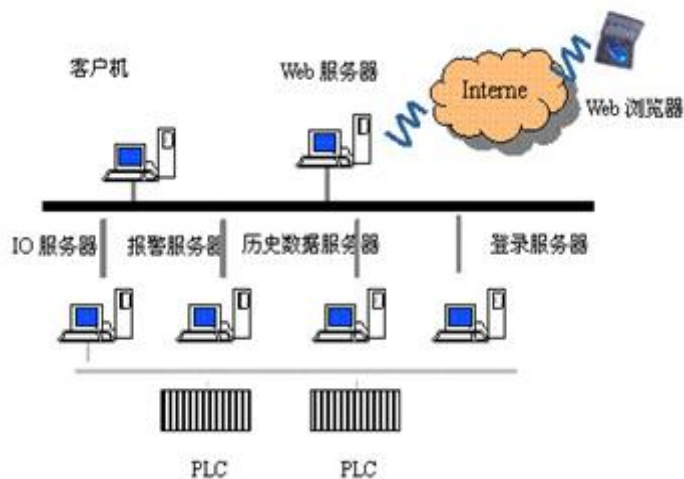


图 21.1 网络结构图

在图 21.1 所示的网络结构图中，IO 服务器只负责设备数据采集，而报警信息的验证和记录，历史数据的记录、用户登录的验证等都被分散到了报警服务器、历史数据服务器和登录服务器中，这样减轻了 IO 服务器的压力。而当 IO 服务器比较多时，这种优势显现的更为突出。报警服务器和历史数据服务器集中验证和记录来自各站点的报警信息和历史数据，IO 服务器和客户端可以集中的从几个服务器上读取到所

需实时数据、报警信息和历史数据。

21.2 网络配置

要实现“组态王 7.0”的网络功能，除了具备网络硬件设施外，还必须对组态王 7.0 的各个站点进行网络配置，设置网络参数，并且定义在网络上进行数据交换的变量，报警数据和历史数据的存储和引用等等。为了使用户了解网络配置的具体过程，下面以一个系统的具体配置来说明。

21.2.1 认识网络配置

在使用网络功能之前，要了解组态王 7.0 需要做哪些配置和工作。组态王 7.0 支持使用 TCP/IP 通信协议的网络。同一网络上每台计算机都要设置相同的通信协议。首先认识一下网络配置对话框。

在组态王 7.0 工程浏览器中，选择菜单“配置\网络设置”，或者在目录显示区中，选择大纲项系统配置下的成员网络配置，双击网络配置图标，双击此图标，弹出“网络配置”对话框，如图 21.2 所示。

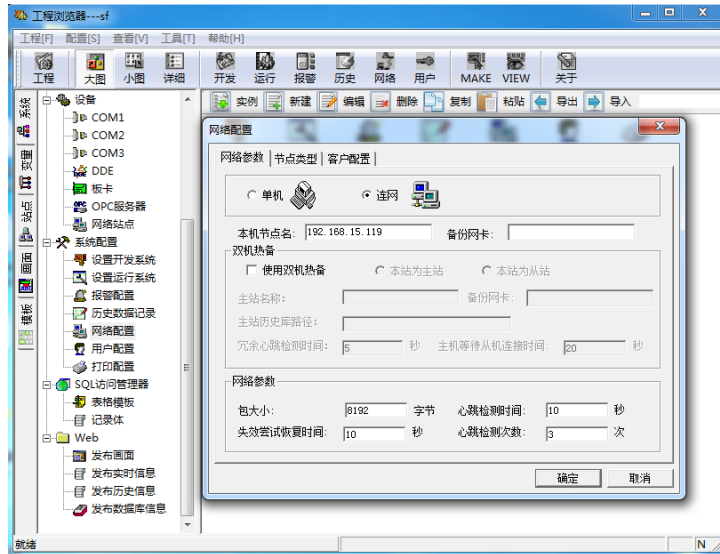


图 21.2 网络配置

网络配置有三个属性页：网络参数、节点类型和客户配置，下面分别说明其使用方法：

1、“网络参数”配置：

“组态王 7.0”运行分单机和连网两种模式，所有进入网络的计算机都要选择“连网”运行模式。如图 21.3 所示。



图 21.3 网络参数

网络参数配置页中各项的含义：

◇ 本机节点名：

就是本地计算机名称。进入网络的每一台计算机必须具有唯一的节点名，在本例中输入节点名“数据采集站”。同样，可以分别设置报警服务器、历史数据记录服务器、登录服务器和客户端的本机节点名为：“报警数据站”、“历史数据站”、“登录站”和“调度室”。本地节点名也可以使用本地主网卡的 IP 地址。

◇ 备份网卡：

当网络中使用双网络结构时，需要对每台连网的机器安装两个网卡——主网卡和从网卡，此处表示从网卡（亦称备份网卡）。在该编辑框中输入从网卡的 IP 地址。

◇ 网络参数:

“组态王 7.0”在“服务器”和“客户”之间为每一个需要传送的变量建立了对应关系。网络参数应该根据具体的网络情况来设置。

◇ 包大小:

用于控制在两个节点之间发送的数据包的长短。增大“包大小”可以增加数据吞吐量。但低速网络应该谨慎设置此项。

注意:

为保证网络传输的正常运做，同一网络系统中的包大小数值必须是一致的。

◇ 心跳检测时间:

此参数在本节点做“服务器”或“客户端”时都有效，以此时间间隔检测数据链路是否畅通。单位为秒。

◇ 心跳检测次数:

此参数在本节点做“服务器”或“客户端”时都有效，例：心跳检测次数为 5，那么当累积心跳检测失败达到 5 次后，表明数据链路中断。

◇ 失效尝试恢复时间:

此参数在本节点做“客户端”时有效，当“客户端”网络精灵检测数据链路中断后，尝试与“服务器”进行连接的时间间隔。单位为秒。

以上四个参数在使用基于 TCP/IP 协议的网络时都需要设置。

注意:

一般情况下,可以直接使用“网络参数”对话框的默认值,若需要更改,请首先咨询北京亚控科技发展有限公司的技术热线。

◇ **双机热备:**

组态王 7.0 提供双机热备功能(见第二十二章),分为“使用双机热备”,“本站为主站”和“本站为从站”三种选项。若使用双机热备功能,则选择“使用双机热备”;若使用,根据当前计算机工作状态设置为主机或从机。

◇ **主站(从站)名称:**

当选择使用双机热备功能,并且选择“本站为从站”时,此选项有效,需要在此处键入主站名称。当选择“本站为主站”时,主站名称变为从站名称,需要在此处键入从站名称。

◇ **备份网卡:**

当网络中存在双网络冗余时,从站点也需要安装两个网卡,在该编辑框中输入从站点备份网卡的 IP 地址。

◇ **主站(从站)历史库路径:**

选择使用双机热备功能后,当选择“本站为主站”时,在此处键入从站历史库记录全路径(按 UNC 格式),若选择“本站为从站”,在此处键入主站历史库记录全路径(按 UNC 格式)。

◇ **冗余机心跳检测时间:**

本节点做主机或从机时都有效,主从机双方以此时间间隔检测数据链路是否畅通。

单位为秒。

✧ 主机等待从机连接时间：

本节点做主机时有效，此参数影响主机激活的时间。主机启动后，如果在此时间间隔后发现没有从机连接，就认为从机不存在，自动激活。如果这个时间设置的短则主机激活快。但是在不同网络状态下(如双网)，从机的连接操作可能会耗费较长时间(有时候 10 秒以上)，因此为保证主、从机之间正常连接。此参数设置不要低于 10 秒。

单位为秒。

2、“节点类型”配置

该属性页主要是定义本地计算机在网络中充当的服务器功能，本地计算机可以充当一种或多种服务器的角色，同时，在网络中所有的站点充当服务器或客户都是相对而言的，即如果一台站点是服务器，也可以指定其作为别的站点的客户，反之作为客户站点，也可以指定其作为别的站点的服务器。对于报警服务器和历史数据服务器，允许指定其作为哪几台 I/O 服务器的报警或历史数据记录服务器。



例如：

在存在数据采集站 1、数据采集站 2、数据采集站 3 的情况下，可以指定一台报警服务器是三个数据采集站的服务器或单独指定其是其中一台数据采集站的服务器。节点类型对话框如图 21.4 所示。

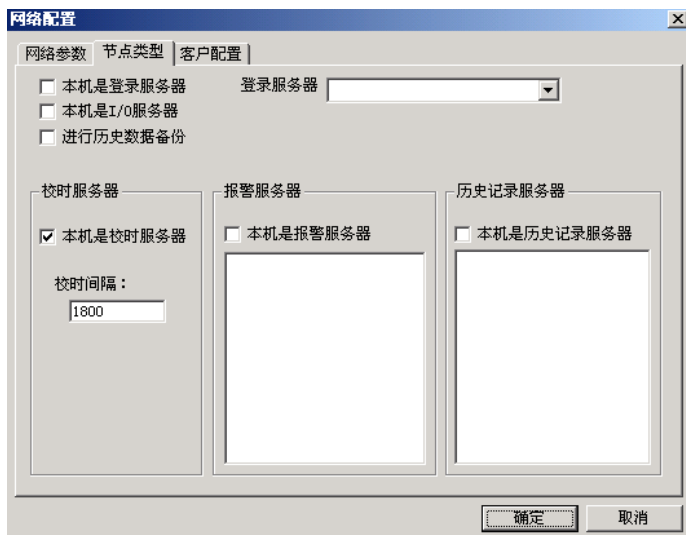


图 21.4 配置节点类型

节点类型配置页中各项的含义：

✧ **本机是登录服务器：**

对于网络工程，需要网络中有唯一的用户列表，其列表存储在登录服务器上，当访问网络中任何站点上有权限设置的信息时，都必须经过该用户列表进行验证。选中该项时，本地计算机在网络中充当登录服务器。当登录服务器没有启动时，用户的验证只能通过本机的用户列表进行，并且在操作网络变量时将以无用户状态进行。当不选“本机是登录服务器”时，必须从登录服务器列表中选择登录服务器机器名称。关于名称如何进行选择，在以后的章节中详细介绍。

✧ **本机是 IO 服务器：**

选中时，表示本地计算机连接外部设备，进行数据采集，并向网络上的其他站点提

供数据。

◇ 进行历史数据备份：

为历史数据库备份的选项，选中该项，表明本机只作为 I/O 服务器，而不作为历史数据服务器时，要暂时保存要向历史数据服务器备份的本机历史数据，将历史数据记录在本机指定的历史记录路径下。

◇ 本机是校时服务器：

组态王 7.0 运行中，尽量保持各台机器的时钟一致，选中“本机为校时服务器”时，本地计算机充当校时服务器，采取广播的方式以指定的时间间隔向网络上的各台机器发送校时帧，保持网络的始终统一。在“校时间隔”编辑框中输入校时间隔，单位为秒。

◇ 本机是报警服务器：

在分布式报警系统中，指定一台服务器作为报警服务器，在该服务器上产生所有的报警（可以指定需要生成报警的 I/O 服务器），客户机可直接浏览报警服务器中的报警信息。在报警服务器的列表框中系统会自动列出已建立连接的所有充当 I/O 服务器的远程站点的站点名。当选中“本机是报警服务器”时，可以对列表框中的 I/O 服务器进行选择，即定义本机是哪个 I/O 服务器的报警服务器。使用方法为，在“本机是报警服务器”下的站点名前的选择框中单击，选择框中出现选中标记即可。

◇ 本机是历史记录服务器：

在分布式历史数据库系统中，指定一台服务器作为历史记录服务器，在该服务器上存储所有的历史数据（可以指定需要存储历史数据的 I/O 服务器），客户机可直接浏

览历史记录服务器中的历史数据。在历史记录服务器的列表框中系统会自动列出已建立连接的所有充当 I/O 服务器的远程站点的站点名。即定义本机是哪个 I/O 服务器的历史记录服务器。

3、客户配置

该属性页主要是定义本地计算机在网络中充当的客户功能,本地计算机可以充当多台服务器的客户。如图 21.5 所示,为客户配置属性。

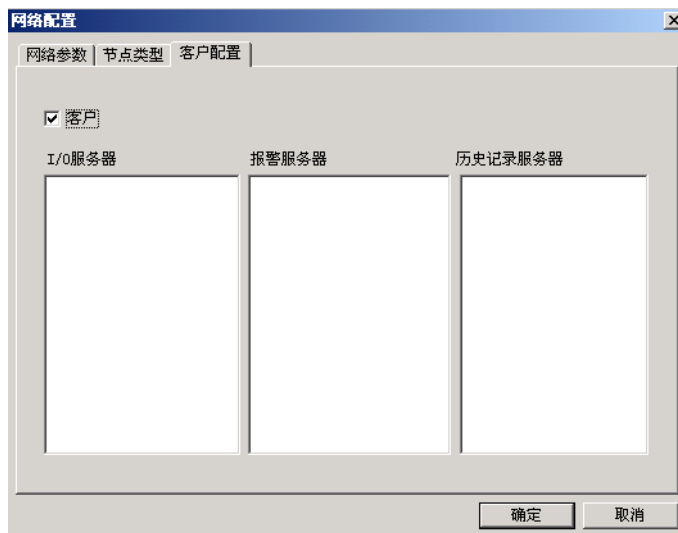


图 21.5 配置节点类型

◇ 客户：

当选定时,表明本地计算机在网络当中充当客户的角色,同时在 I/O 服务器、报警服务器、历史记录服务器中会自动列出网络中的所有 I/O 服务器、报警服务器和历史记录服务器,注意,只有配置好了网络站点后才会列出这些已定义好的服务器。

◇ I/O 服务器：

在网络当中可以存在多台 I/O 服务器，负责从外部采集数据。在该列表框中会自动列出网络上的所有充当 I/O 服务器的站点，而本地计算机可以充当一台或多台 I/O 服务器的客户端，从这些服务器端取得采集的数据。

◇ **报警服务器：**

在网络当中可以存在多台报警服务器，在其上负责验证和存储指定站点的所有数据的报警信息。在该列表框中会自动列出网络上的所有充当报警服务器的站点，而本地计算机可以充当一台或多台报警服务器的客户端，从这些报警服务器端引用报警。

◇ **历史记录服务器：**

在网络当中可以存在多台历史记录服务器，在其上存储指定站点的所有的历史数据。在该列表框中会自动列出网络上的所有充当历史记录服务器的站点，而本地计算机可以充当一台或多台历史记录服务器的客户端，从这些历史记录服务器端查询历史数据。

4、建立远程站点

要建立客户——服务器模式的网络连接，就要求个站点共享信息，互相建立连接。组态王 7.0 在工程浏览器中的左边设置了一个 TAB 按钮——“站点”，单击该按钮，进入站点管理界面。界面共分为两个部分，左边为站点名称列表区，右边为站点信息区。如图 21.6 所示。

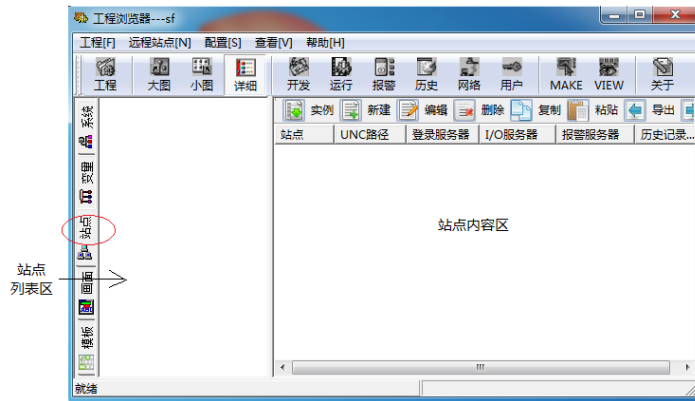


图 21.6 远程站点管理界面

在站点列表区中单击鼠标右键，弹出快捷菜单，在菜单中选择“新建远程站点”选项，弹出“远程节点”对话框，如图 21.7 所示。在对话框的“远程工程 UNC 路径”编辑框中输入网络上要连接的远程工程的路径（UNC 格式），或直接单击“读取节点配置”按钮，在弹出的文件选择对话框中选择远程工程路径。选择完成后，该远程站点的信息就会被全部读出来，自动添加到对话框中对应的剩下的各项中。如主机节点名、节点类型等，都会自动读取并添加的。也可以按照远程站点实际的网络配置，手动添加或选择对话框中的选项。定义完成后，单击“确定”按钮关闭对话框。

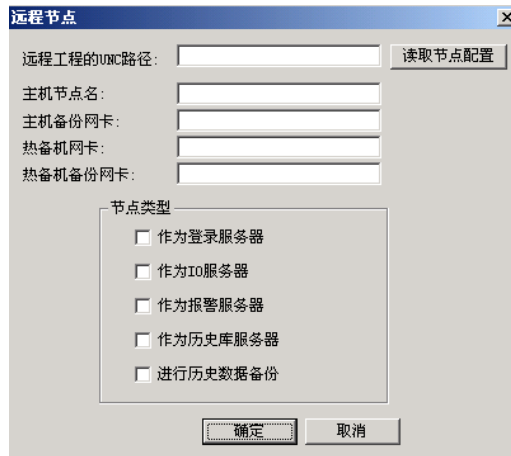



图 21.7 远程节点建立对话框

节点建立后，在工程浏览器——站点的站点列表区和站点信息区会显示出该站点的所有信息。

21.2.2 网络配置实例

在组成网络系统时，各站点上的工程路径必须完全共享给网络上的用户，以方便工程站点网络配置。下面以一个具体的网络工程实例来说明组态王 7.0 的网络配置及应用。如图 21.8，为网络结构图。

 注意：

远程站点上的工程所在的路径的文件夹必须设置为完全共享，否则会出现开发系统读取远程变量失败的现象。并且远程站点的组态王 7.0 工程的网络配置中必须设置为“连网”。否则建立系统会出现如下图提示：

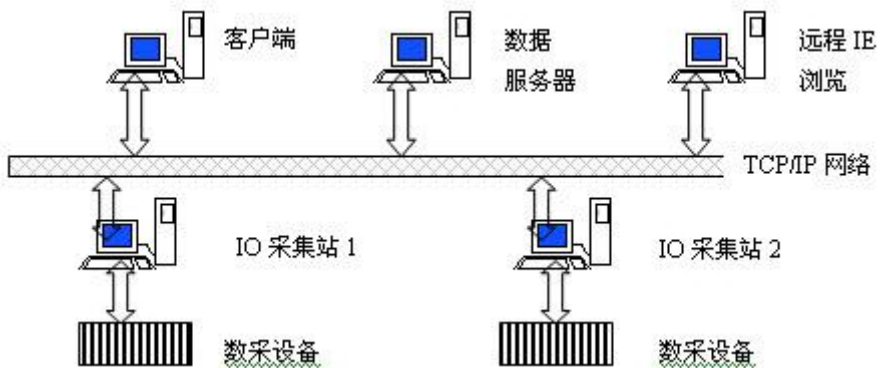


图 21.8 示例网络结构图

在该网络结构中，有以下几种站点：

- ◇ 两个 **IO 采集站**，负责 IO 数据采集和控制。要求 IO 采集站要看到报警信息和历史数据。
- ◇ 一个 **数据服务器**，承担报警服务器、登录服务器和历史记录服务器的角色。也作为中控室的调度站。

- ◇ 一个或多个**客户端**，浏览 I/O 采集站上的实时数据，查看各 I/O 站点的报警信息，查询各 I/O 站点的历史记录，可以实现对 I/O 站点连接设备的控制。

网络配置步骤：

1、首先配置数据服务器站点。

进入数据服务器站点上的工程浏览器，打开网络配置对话框，选择“连网模式”。在主机节点名中输入本机的计算机名称或 IP 地址，如在本例中计算机名为“数据服务器”。网络参数按照默认值，其它项目不用修改。如图 21.9 所示。

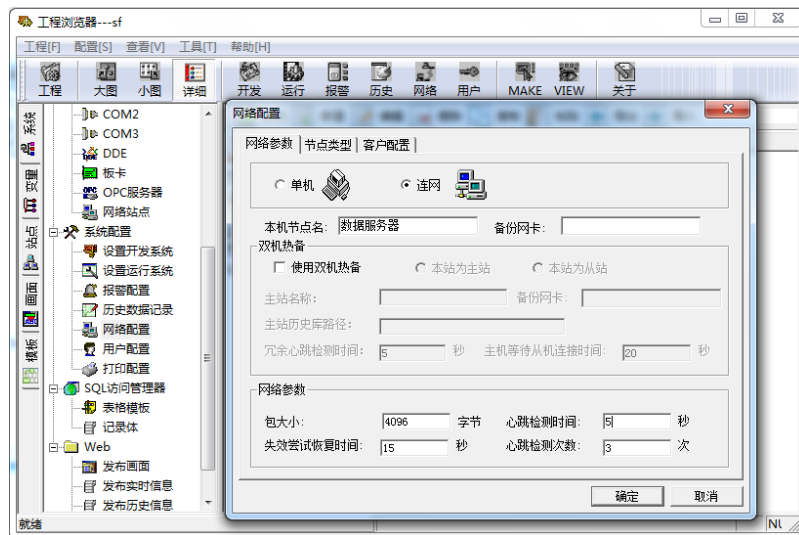


图 21.9 数据服务器网络参数配置

在**节点类型**对话框中，选择“本机是登录服务器”、“本机是报警服务器”、“本机是历史记录服务器”选项。为了保证网络时钟的一致，也可以在这里选择“本机是校时服务器”，然后输入校时间隔，或按默认值。如图 21.10 所示。配置完成后，单击“确

定”按钮，关闭对话框，暂时完成该站点的网络配置。该节点网络配置在其它站点配置完成后还需要继续进行。

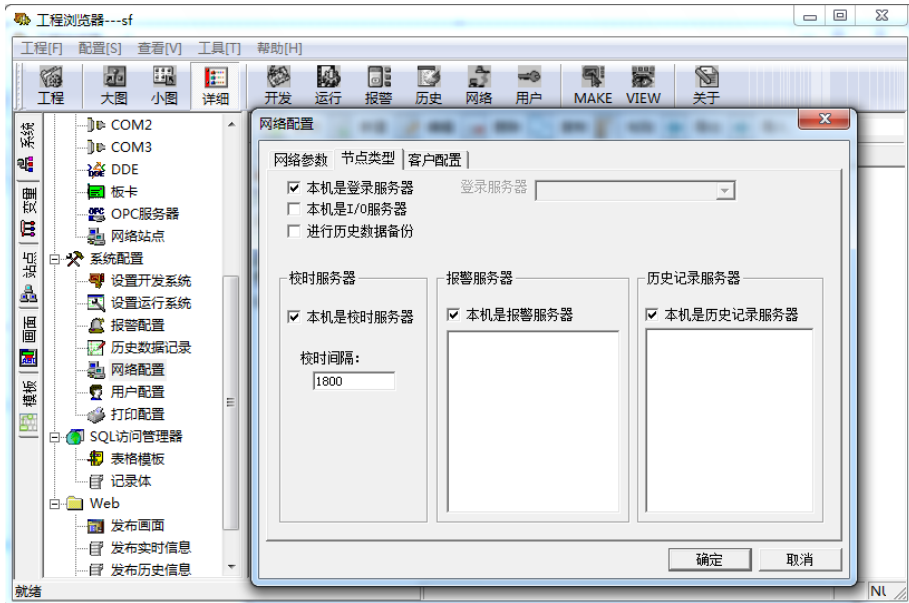


图 21.10 数据服务器节点类型配置

2、接下来配置 I/O 采集站

首先配置 I/O 采集站 1。在采集站 1 的节点名称中输入本机节点名，如本例中为“I/O 采集站 1”。其它选项不用修改。如图 21.11 所示。

⚠ 注意：

网络中个连接节点上的网络参数设置应该是一致的。

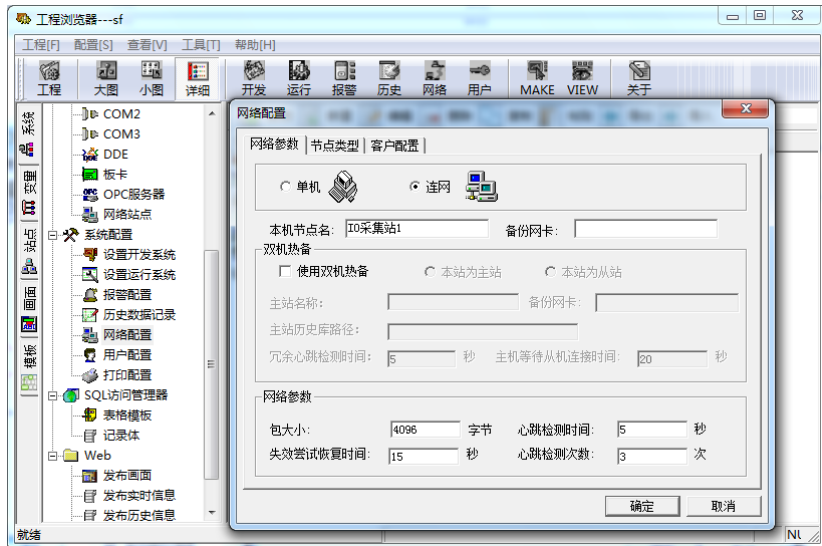


图 21.11 网络中 I0 采集站的网络参数配置

- ✧ 在“节点类型”中选择“本机是 I0 服务器”选项。（此处为了建立一个远程站点，先选择“本机是登录服务器”选项，待网络配置完成后修改。如果不选择该选项，单击确定按钮时，系统会提示“选择一个登录服务器”。）单击确定按钮，关闭对话框。
- ✧ 在 I0 采集站的工程浏览器的左边选择“站点”标签，进入站点管理界面。在左边的节点名称列表区域单击鼠标右键，在弹出的快捷菜单中选择“新建远程站点”，弹出“远程节点”对话框。单击对话框上的“读取节点配置”按钮，选择远程工程路径，如图 21.12 所示。在网络中选择“数据服务器”上共享的工程文件夹（注意：这里一定要选择到工程所在的直接文件夹），单击确定按钮，关闭对话框。则“数据

服务器”配置的工程信息被读到了“远程节点”对话框中如图 21.13 所示。确认读到的信息无误，单击“确定”按钮关闭对话框。如图 21.14 所示，在 I0 采集站 1 的“站点”界面上出现了一个“数据采集站”的信息，点击“数据词典”，可以直接看到远程数据服务器上的变量。



图 21.12 选择远程工程路径

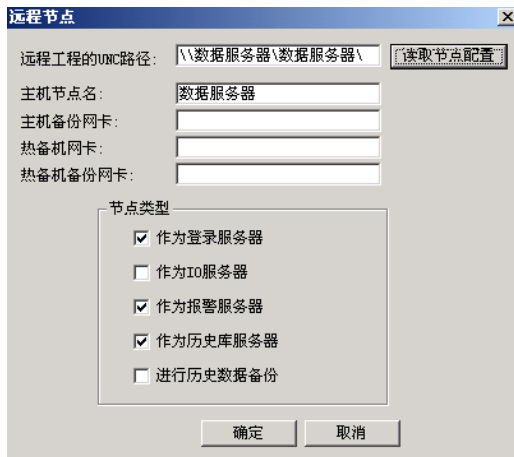


图 21.13 远程节点内容

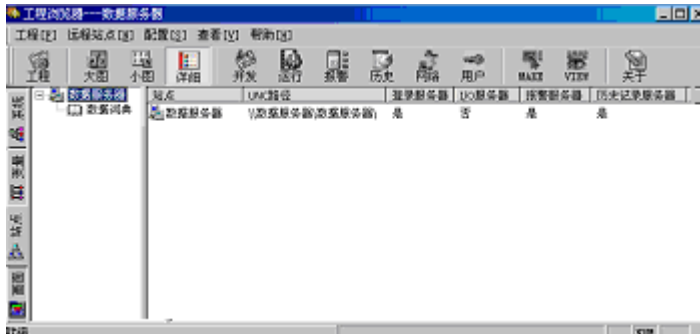


图 21.14 增加站点后的界面

- ◇ 新远程站点建立成功后，就可以进一步进行 I/O 采集站的网络配置了。选择 I/O 采集站工程管理器上的“系统”标签，双击“网络配置”项，选择“节点类型”页，去掉“本机是登录服务器”选项，在“登录服务器”列表中选择“数据服务器”作为本机的登录服务器，如图 21.15 所示。选择“客户配置”页，选中“客户”选项，此时，报警服务器和历史记录服务器列表变为有效可选，在这两个列表中列出了当前工程中添加的作为报警服务器和历史记录服务器的站点名称。选中各列表的站点名称前的复选框，如图 21.16 所示，表示当前的“I/O 采集站 1”作为“数据采集站”的客户端，看到报警和历史记录数据。配置完成后，单击“确定”按钮关闭对话框。I/O 采集站 1 的网络配置全部完成。I/O 采集站 2 的网络配置完全按照这个步骤执行。

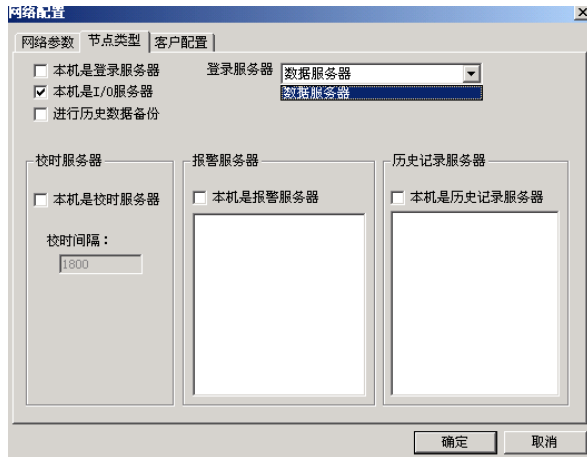


图 21.15 选择登录服务器

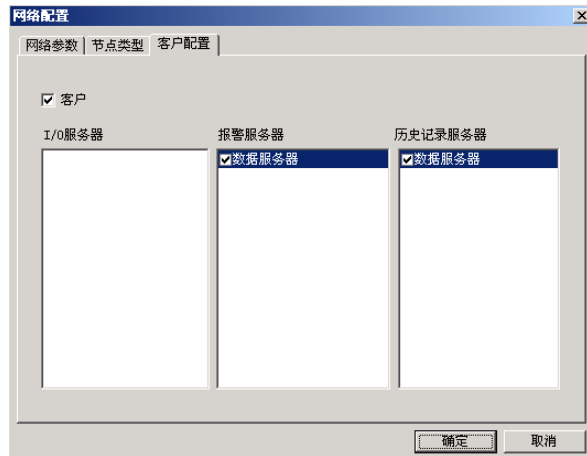


图 21.16 选择服务器

✧ 按照以上步骤，在“数据服务器”上的“站点”中新建“IO 采集站 1”、“IO 采集站 2”远程站点，完成后，打开“网络配置”对话框，进一步进行“数据服务器”的网

络配置。在“节点类型”页中，与图 21.10 相比，在图 21.17 中“本机是报警服务器”和“本机是历史记录服务器”的列表中列出了连接到本机的 IO 服务器的名称。在列表中选择 IO 服务器，表示本机在运行时作为“IO 采集站 1”和“IO 采集站 2”的报警和历史记录服务器，验证、存储来自这两个 IO 服务器的报警、历史记录数据。在“数据服务器”指定的历史记录目录下，系统会自动以 IO 采集站命名创建两个文件夹，分别保存个采集站的历史记录数据。

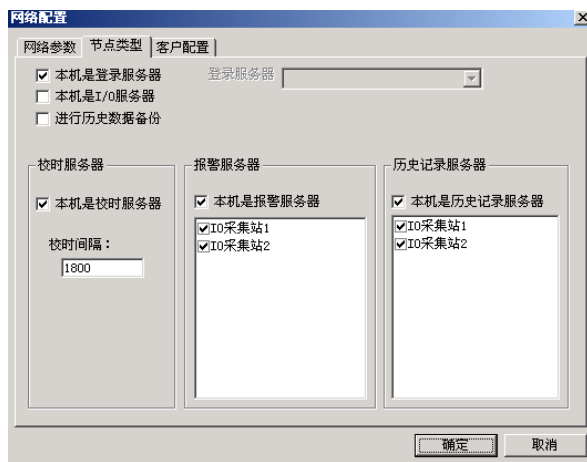


图 21.17 配置数据服务器的节点类型

- ✧ 选择“客户配置”页，选择“客户”选项，在“IO 服务器”列表中选择两个 IO 采集站的名称，表示本机作为 IO 采集站的客户端可以远程引用和访问 IO 采集站上的变量和数据。如图 21.18 所示。配置完成后，单击“确定”按钮关闭对话框。



图 21.18 数据服务器的客户配置

- ◇ 所有服务器都配置完成后，就可以来配置客户机了。启动客户端工程的工程浏览器，选择“站点”标签，新建三个远程站点——IO 采集站 1、IO 采集站 2、数据服务器。打开网络配置，选择“连网”模式，在“本机节点名”中输入本机的计算机名称。选择“节点类型”，在“登录服务器”列表中选择“数据服务器”作为本机的登录服务器。选择“客户配置”页，选中“客户”选项，在各个服务器的选项列表中进行选择，如图 21.19 所示，选择的选项表明本机作为 I/O 服务器——“IO 采集站 1”、“IO 采集站 2”——的客户端，可以远程引用和访问这两个站点上的变量和实时数据。作为报警服务器和历史记录服务器——“数据服务器”——的客户端，可以访问到该站点上保存的报警和历史记录信息和数据。配置完成后，单击“确定”按钮关闭对话框。

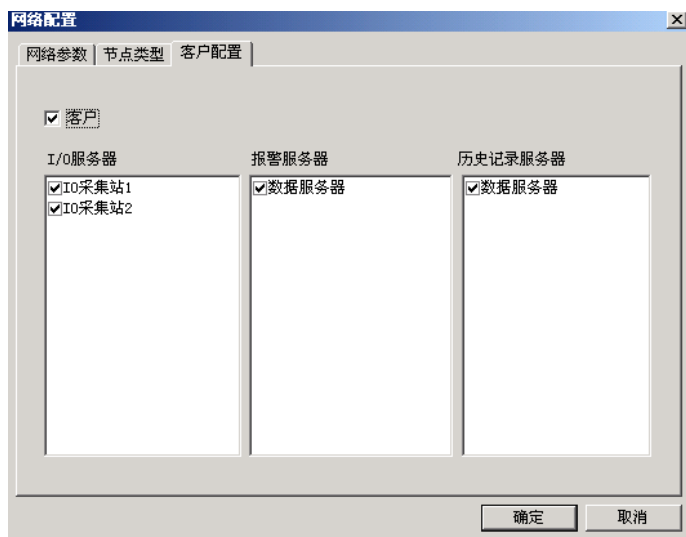


图 21.19 客户机的客户配置

至此，所有网络的配置全部完成，下一步就是进行具体的网络工程的制作了。一般因为 IO 服务器是数据源站点，所以首先制作 IO 服务器的工程，然后根据具体需要开发其它各服务器和客户端的工程。

21.3 网络变量使用

21.3.1 远程变量的引用

组态王 7.0 是一种真正的客户——服务器模式，对于网络上其它站点的变量，如果两个站点之间建立了连接（如在 21.2.2 节中的“数据服务器”上建立了“IO 采集站 1”和“IO 采集站 2”两个站点的连接），可以直接引用。例如，在站点“数据服务器”的组态王 7.0 工程中查看“IO 采集站 1”上定义的 IO 变量反应罐温度。

在画面上建立变量模拟值输出时，弹出模拟值输出连接对话框，打开变量浏览器，在变量浏览器的左边目录中，显示了可以访问到变量的站点，其实除了本站点外，其余都是本站点的“IO 服务器”。选择“IO 采集站 1”，在变量浏览器的右边变量显示区域中列出了所有的 IO 采集站 1 的变量，如图 21.20 所示。在变量列表中选择“反应罐温度”，在变量浏览器底部的状态栏中显示“\\IO 采集站 1\反应罐温度”，单击确定按钮，关闭变量浏览器。在动画连接“表达式”一栏中显示出了选择的变量名及节点名称，如图 21.21 所示。或直接在动画连接“表达式”一栏中输入远程站点的变量名程，其书写格式为\\站点名\变量名，结果也是一样的。



图 21.20 浏览远程变量列表




图 21.21 动画连接中引用远程变量

在动画连接时引用远程变量都可以照此方法操作，在命令语言中引用远程变量时，同样只需要写成\\站点名\变量名，例如：



图 21.22 命令语言中引用网络变量

 注意：

在定义数据改变命令语言和事件命令语言时，不能使用远程变量来作为触发脚本执行的条件。

在引用远程变量时，建好连接的两个站点上的组态王 7.0 工程的启动没有先后之分，即哪一个站点先启动都没关系，只有当两个站点都启动后，变量的引用关系才会发生，即客户端引用的 I/O 服务器端的数据就会与 I/O 服务器上的该数据的值一致，除非出现共享路径不存在，网络不通，计算机不存在等原因。

21.3.2 远程变量的回写

远程站点除了可以引用变量外，还可以改变变量的数值，即回写变量，使设备上的数据发生变化，可以在动画连接时或命令语言中定义回写远程变量。在权限允许的情况下，网络上的任何一个站点均可以回写变量，即远程修改变量和变量的域的值。修改变量值时，如果远程变量具有安全权限，如果是动画连接中的“值输入”和“滑动杆输入”等动画连接，则必须登录用户达到权限后才能操作，否则系统信息窗中会提示没有修改变量的权限。当有权限的变量在命令语言里被引用而改变值时，此时为了保证命令语言的正确运行，设置的权限是不起作用的，变量的值会被改变。

在连接的两个站点中，总是一个站点作为服务器端，另一个站点作为客户端，例如“数据服务器”就是“I/O 采集站 1”的客户，“I/O 采集站 1”就是“数据服务器”的 I/O 服务器。在两个站点的连接过程中，客户端为了正确地得到服务器端不断变化的数据，必须不断检测与服务器的数据链路是否畅通，参数设置详见 21.2 网络设置一节中的网络参数。

21.4 分布式报警引用

组态王 7.0 的报警系统是真正基于网络的分布式报警系统，在设定为报警服务器的站点上验证和存储所有的报警和事件信息（可以指定需要验证和存储报警的 IO 服务器），其他任何站点都可作为客户端，直接浏览报警服务器中的报警和事件信息。

21.4.1 分布式报警的概念

报警服务器端负责确定是否需要产生报警、验证报警、存储报警并将报警信息传输到报警引用端（即报警服务器的客户端，如“客户端”）。报警服务器端产生的变量报警是根据 IO 服务器端定义变量时对变量的报警定义而产生的。即当 IO 服务器变量变化时，IO 服务器会将变量的变化信息发送到报警服务器，报警服务器将根据该信息和变量的报警定义确定是否发出报警信息。在报警服务器发出报警信息后，报警服务器可以验证、确认该报警，并将该报警信息放到报警缓冲区中，进行显示和存储，同时将该信息发送到报警服务器的客户端上。也就是说，报警的验证和产生是由报警服务器完成的。这样 IO 服务器只负责发送变量变化信息，而不会产生报警信息；而报警服务器则负责产生报警信息，并对报警信息进行验证处理，通知所有客户端，并记录。这样每个站点分别担任不同角色，分工合作，提高了整个系统的效率。



注意：

一个网络中，只能有一个报警服务器。

21.4.2 分布式报警的显示和记录

按照 21.2.2 节中的示例，配置完成网络后，在客户端站点上的画面中创建一个报警窗口，双击报警窗，打开报警窗的属性对话框，定义报警窗的名称后，选择“条件属性”属性页，如图 21.23。在“报警服务器名”列表中列出了当前工程网络配置中作为本机报警服务器的机器名称，如本例中为“数据服务器”。在“报警信息源站点”列表中列出的是作为本机 I/O 服务器的机器名称，选择当前报警窗要显示属于哪个 I/O 服务器上的报警信息，如只选择“IO 采集站 1”，则运行时，只显示来自该站点的报警信息。以上两个选项在系统运行时可以在线修改。实时报警窗和历史报警窗的处理都是一样的。在运行时的报警窗口中“机器名”和“报警服务器名”两列中可以看到报警的报警信息源站点和报警服务器的名称。

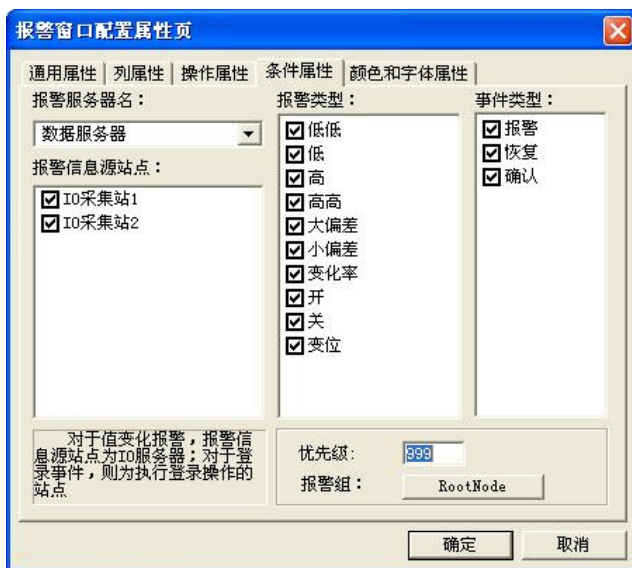


图 21.25 报警记录中记录站点名称

报警服务器将所有站点的报警信息保存到报警配置中指定的路径下的文件里。各站点的报警区分也是靠记录的“IO 服务器名”区分的。如图 21.25 所示，在报警配置中报警格式配置中选择“IO 服务器名”项和“机器名”（报警服务器）项，则报警记录时将该信息记录到文件中。其它格式含义相同。

21.5 分布式历史数据访问

组态王 7.0 的历史数据系统是真正基于网络的分布式历史数据系统，在设定为历史记录服务器的站点上存储所有的历史数据（可以指定需要存储历史数据的 IO 服务器），其他任何站点都可作为客户端，直接访问历史记录服务器中的历史数据。

历史记录服务器端负责存储历史数据并将历史数据传输到历史数据引用端（即历史记录服务器的客户端，如“调度室”）。历史记录服务器端存储的变量的历史数据是根据 IO 服务器端定义变量时对变量的记录定义而产生的。关于变量的记录定义详见第二十五章组态王 7.0 历史库。

下面以“数据采集站”的 IO 变量反应罐温度为例，详细说明分布式历史数据系统的使用。

1、网络结构：

网络中至少有三个站点：IO 服务器、历史记录服务器和客户端，在本例中由“数据采集站”、“历史数据站”和“调度室”来分别充当这三个站点，结构示意图如下：

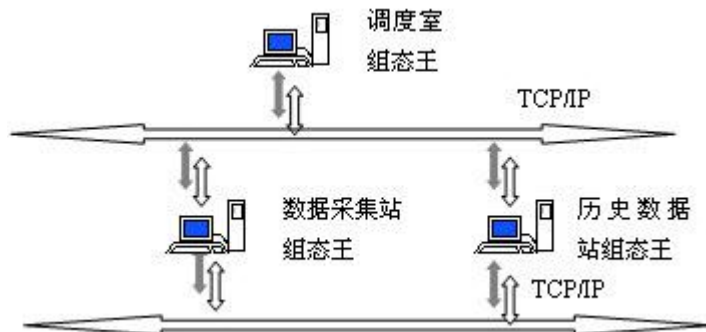


图 21.26 分布式历史数据访问结构示意图

在此网络结构中，“数据采集站”负责采集来自外部设备的数据，历史数据库是由运行在“历史数据站”上的组态王 7.0 生成的，“调度室”可以访问“历史数据站”中的历史数据库，也就是说历史数据站中的历史数据库成为共享数据库。

2、网络配置和站点配置：

三台机器均配置远程站点连接，其具体操作方法详见 21.2 网络配置一节中节点类型下的有关站点连接的说明。例如“历史数据站”中建立两个远程站点：“数据采集站”和“调度室”。然后再对每个站点进行网络配置，其具体操作方法详见 21.2 网络配置一节中网络参数和节点类型的说明。

- ◇ “数据采集站”的配置：与 21.2 中配置一样。在网络配置下的节点类型中选中“本机是 I/O 服务器”。
- ◇ “历史数据站”的配置：在网络配置下的节点类型中定义为“本机是历史记录服务器”。并且在下面的列表框中选中“数据采集站”，表示本机充当“数据采集站”这一个站点的历史记录服务器。如下图 21.27 所示。

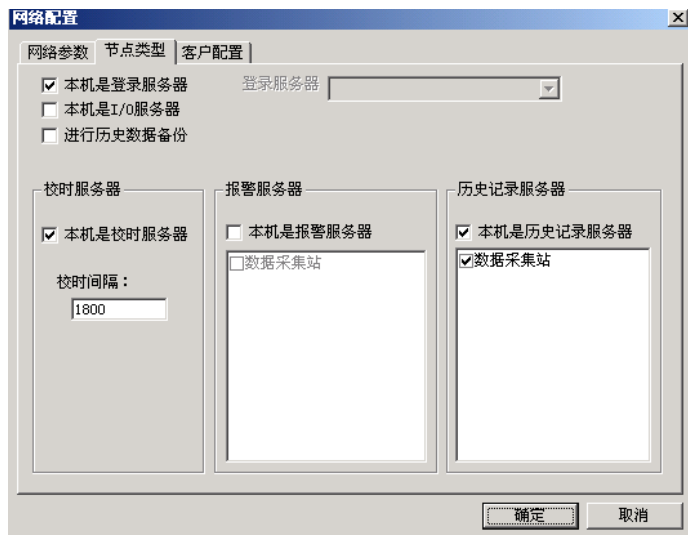



图 21.27 定义历史记录服务器

- ✧ “调度室”的配置：在网络配置下的客户类型中，选中“客户”，并且在 I/O 服务器下面的列表框中选中“数据采集站”，在历史记录服务器下面的列表框中选中“历史数据站”，表示本机是“数据采集站”和“历史数据站”的客户，如下图 21.28 所示。



图 21.28 定义调度室作为 IO 服务器和历史记录服务器的客户

 注意:

如果“调度室”只是单纯访问历史记录服务器的历史数据，那么在本设置中可以不选 IO 服务器下的“数据采集站”，即“调度室”不做“数据采集站”的客户。反之，如果选中，则表示“调度室”可以直接访问 IO 服务器“数据采集站”的变量。

3、历史数据记录定义和记录配置：

变量的数据记录定义在 IO 服务器上完成，历史记录配置在历史记录服务器上完成，在客户端只能访问历史记录服务器存储的历史数据。下面以“数据采集站”、“历史数据站”和“调度室”的设置为例来详细讲述其操作方法。

✧ 数据采集站： 在数据采集站上进行变量的记录定义，其详细操作见“第五章

变量定义和管理”中的记录定义。例如：定义 I/O 量反应罐温度，其记录定义如下图 21.29 所示。

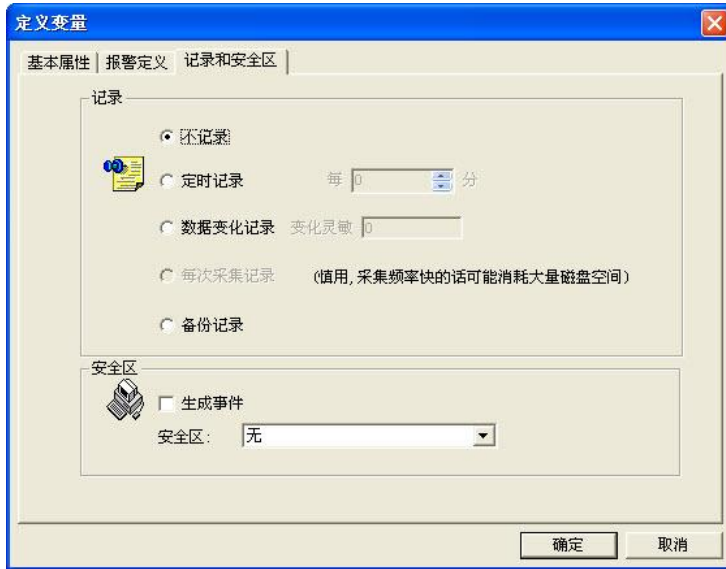
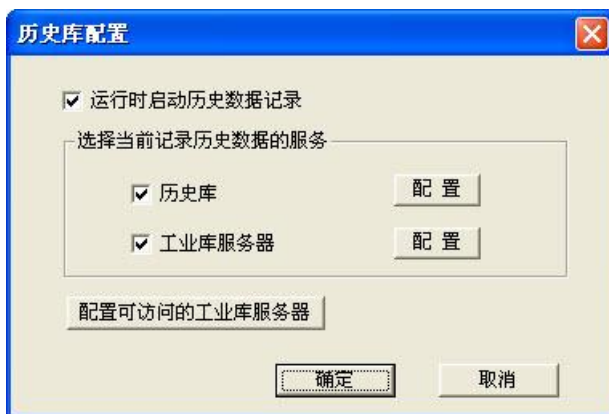


图 21.29 定义记录数据

定义了“数据变化记录”，既当 I/O 服务器的变量反应罐温度发生变化时就记录到历史数据库中。

- ✧ 历史数据站：在“历史数据站”上进行历史数据记录配置，在组态王 7.0 工程浏览器中，用鼠标左键单击菜单命令“配置/历史数据记录”，或者从左边的工程目录显示区中选择大纲项“系统配置”下的“历史数据记录”，并用左键双击，则弹出“历史库配置”对话框如下图所示。



运行时启动历史数据记录：选择该项后，当系统运行时，自动开始历史数据的记录，否则不进行数据历史记录。用户可以选择运行系统中的菜单“特殊\重启历史记录”来启动历史数据的记录。选中“运行时启动历史数据记录”和“历史库”选项，并且点击“历史库”右边的“配置”按钮，弹出对话框，如图 21.30 所示。




图 21.30 历史数据站的历史记录配置

- **数据保存天数：**历史数据记录文件保存的时间长度，最长为 8000 天，最短为 1 天。当到达规定的时间时，系统自动删除最早的历史记录文件。
- **存储路径：**有两种选择：历史数据服务器上当前工程的路径，或在该服务器上指定的路径。

4、历史数据的存储：

记录的变量的历史数据，可以在历史记录服务器上统一存储。历史记录服务器将为每台 IO 服务器站点建立相应的目录进行历史数据的存储，存储路径是历史记录服务器设置的路径，在该路径下再以 IO 服务器的站点名为目录名建立一个文件夹，各站点的历史数据分别存储在各文件夹中。

 **注意：**

如果历史记录服务器没有启动，IO 服务器可以正常地运行，但是不能进行历史数据的存储，必须重新启动历史记录服务器的组态王 7.0 工程。

5、历史数据的访问：

用户可以在网络工程的任意一个站点上引用历史记录服务器的历史数据，注意：必须先客户端（例如“调度室”）的网络配置中指定历史记录服务器名称。客户端访问的历史数据完全是由历史记录服务器传送来的。

对于历史数据的访问，实际是在客户端以历史趋势曲线窗口或报表等形式通过网络与历史记录服务器上的历史数据进行对应。当历史记录服务器先启动时，客户端可以访

问启动前在历史记录服务器上已经存储的历史数据。客户端可以通过历史趋势曲线窗口或报表查询历史数据，客户端的历史趋势曲线窗口的配置详见“第八章 趋势曲线和其他曲线”中的历史趋势曲线配置一节；报表查询历史数据详见“第十七章 报表系统”。如果历史记录服务器不启动，客户端则不能查询历史数据。

“调度室”的组态王 7.0 工程运行时历史趋势曲线窗口如下图 21.31 所示。

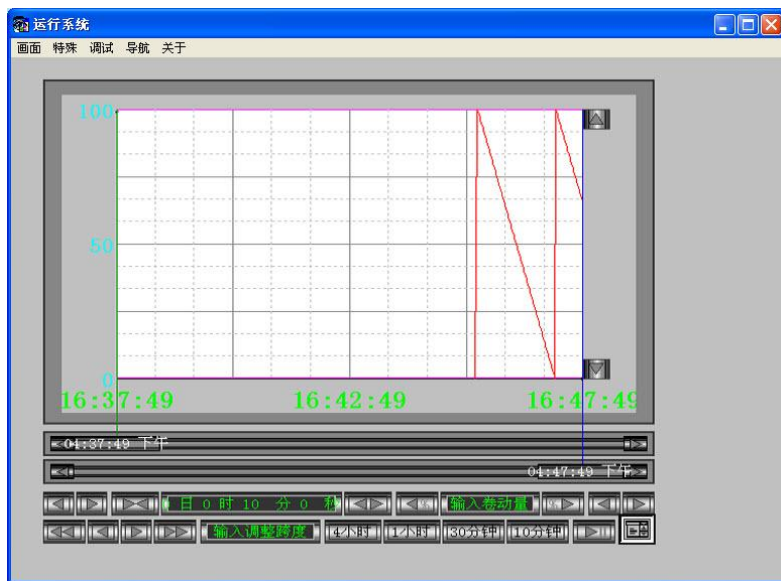



图 21.31 客户端以曲线形式浏览历史数据

21.6 网络登录管理

对于网络工程，为了增强系统的安全性、提高系统的管理能力，在整个系统网络中需要唯一的一个用户信息列表，该用户列表位于登录服务器上，是网络中最全面包含用户信息的集合。所以在配置网络时，工程人员应该在登录服务器上建立最完整的用户列表，并

保证客户端上的用户列表与登录服务器上的用户列表保持一致。当用户在某个站点登录，对系统进行操作时，该登录信息将被传送到登录服务器进行验证，产生登录事件，然后将该事件传送到该登录服务器的报警服务器上，进行登录事件报警的显示、存储。

 注意：

对于网络工程，同一个网络中，只能有一个登录服务器，不要同时配置多个登录服务器。

关于用户登录操作本机对象的使用方法请详见“第十六章 系统安全管理”，下面以“调度室”通过登录后访问“报警数据站”的报警为例，详细说明网络登录管理。

1、网络结构：

网络中由“调度室”、“报警数据站”和“登录站”来分别充当客户端、报警服务器和登录服务器，结构示意图如下图 21.32 所示。

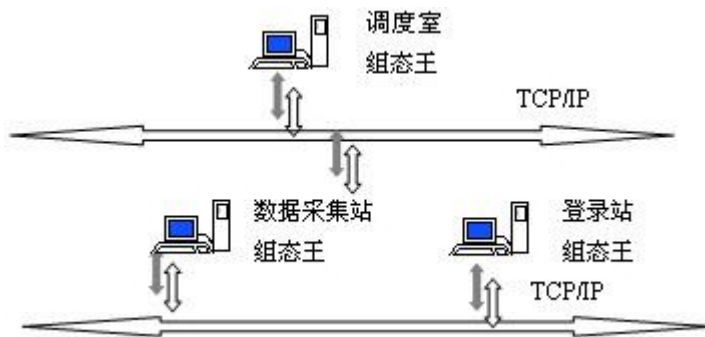


图 21.32 网络登录结构示意图

2、网络配置和站点配置：

“调度室”、“报警数据站”和“登录站”均需配置远程站点连接，其具体操作方法详见 21.2 网络配置一节中节点类型下的有关站点连接的说明。例如“登录站”中建立两个远程站点：“数据采集站”和“调度室”。然后再对每个站点进行网络配置，其具体操作方法详见 21.2 网络配置一节中网络参数和节点类型的说明。

“登录站”的配置：在网络配置下的节点类型中定义为“本机是登录服务器”。如下图 21.33 所示。

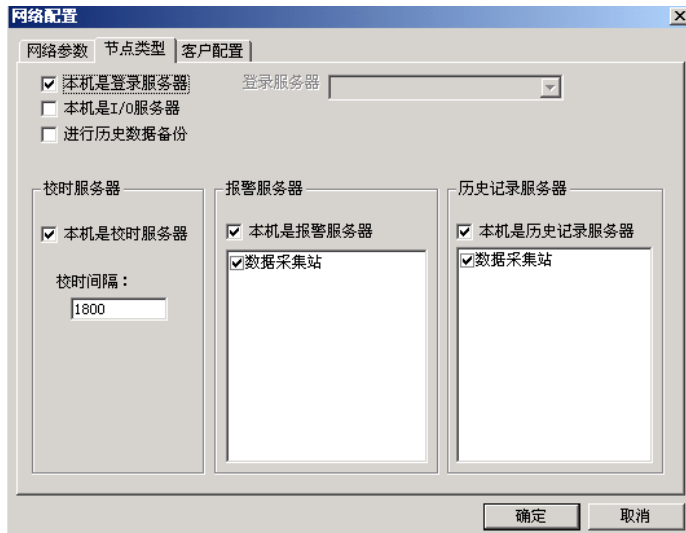


图 21.33 定义登录服务器

“调度室”的配置：在网络配置下的节点类型选中登录服务器为“登录站”。如下图 21.34 所示。

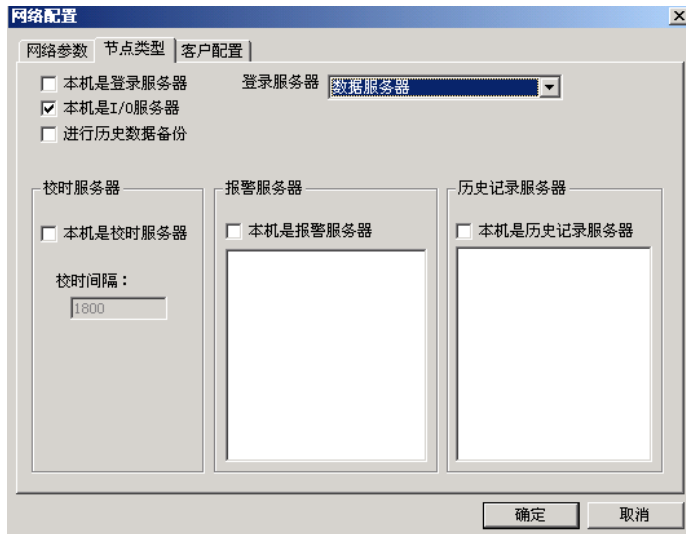


图 21.34 选择登录服务器

3、配置用户：

在登录站上配置所有的访问用户列表。关于用户配置的具体操作方法详见“第十六章 系统安全管理”。例如本例中建立的用户列表如下图 21.35 所示。



图 21.35 配置用户

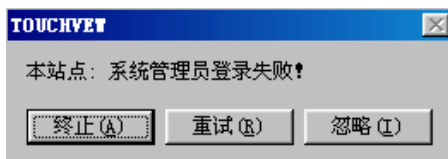
在本例中建立了一个用户组“调度室”，其下有两个成员，“操作员 A”和“操作员 B”，分别为其设置不同的优先级和安全区。

4、客户端登录：

当“调度室”端的操作员以“操作员 A”或“操作员 B”登录操作网络中的工程对象（例如，对报警进行确认或删除）时，会由登录服务器按照其上定义的用户列表进行校验。例如，在“调度室”的画面上设置报警窗的优先级为 500，如果“登录站”上的用户列表中对操作员 A 定义的优先级小于 500，那么操作员 A 通过网络登录后将不能操作该报警窗口。

5、登录对话框：

如果用户登录失败，则系统会弹出登录失败对话框，如下图所示。



终止: 终止用户的登录。

重试: 用户重新登录。


忽略: 忽略在登录服务器上的登录，登录本机。如果在本机登录，再次忽略，则终止登录。

 **注意:**

(1) 当登录服务器没有启动时，用户的验证只能通过本机的用户列表进行，并且在操作网络变量时将以无用户状态进行。

(2) 客户端的登录、注销和站点的启动和退出事件将被传输到远程的登录服务器上，并在登录服务器的报警服务器上产生站点启动和退出事件信息。

21.7 网络精灵

网络精灵是组态王 7.0 软件网络间通讯的工具，数据的收发都是通过该程序实现。在网络工程中，各站点进行通讯时，用户可以通过网络精灵来查看通讯是否正常。工程一旦定义为“连网”，启动运行系统时，网络精灵应用程序将自动启动，对网络通讯状态进行监视。网络精灵是以最小化方式启动运行的，可以双击系统托盘里的“”图标，显示网络精灵运行界面。网络精灵如下图 21.36 所示。

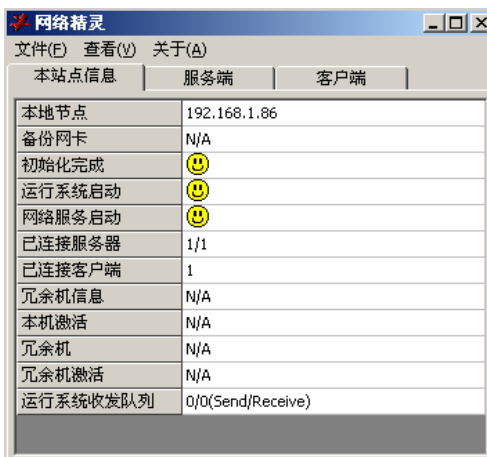


图 21.36 网络精灵——本站点信息

网络工程上的每一个站点启动后都有一个网络精灵，网络精灵中共有三个信息页，可以显示如下方面的信息：

- ◇ 显示本站点网络配置信息；
- ◇ 显示本站点作为服务器的网络信息；
- ◇ 显示本站点作为客户端的网络信息。

下面一一介绍各信息页的内容。

21.7.1 本站点信息

- ◇ **本地节点**：显示开发系统“网络配置”中定义的“本机节点名”。
- ◇ **备份网卡**：显示开发系统“网络配置”中定义的“备份网卡”，双网冗余时有效。
- ◇ **初始化完成**：网络精灵启动运行初始化完成置标志为“😊”，如果没有正确初始化则置标志为“🛑”。

- ✧ **运行系统启动:** 如果组态王 7.0 运行系统正在运行, 显示 “😊”, 如果运行系统退出, 显示 “☹”。
- ✧ **网络服务启动:** 当本节点作为服务器时, 此项有效, 显示 “😊” 时表示服务已启动可以接收客户端连接, 显示 “☹” 时表示服务断开不能接收客户端连接。
- ✧ **已连接服务器:** 当本地节点做客户端时, 需要连接的服务器数量和已经连接的服务器数量。例如显示 “2/3”, 3 表示要连接 3 个服务器节点, 2 表示目前已连接 2 个服务器节点。连接服务器的具体信息在 “客户端” 选项卡中有详细描述。
- ✧ **已连接客户端:** 当本地节点做服务器时, 已经连接的客户端数量。已连接客户端的具体信息在 “服务器” 选项卡中有详细描述。
- ✧ **冗余机信息:** 使用双机热备时有效, 如果本机定义为 “主机”, 此处则显示 “主机”, 如果本机定义为 “从机”, 此处则显示 “从机”。
- ✧ **本机激活:** 使用双机热备时有效, 显示 “😊” 表示本机处于激活状态, 显示 “☹” 表示本机处于监视状态。若没有使用双机热备则显示 “N/A”。
- ✧ **冗余机:** 使用双机热备时有效, 显示冗余机节点名称或 IP 地址。
- ✧ **冗余机激活:** 使用双机热备时有效, 显示 “😊” 表示冗余机处于激活状态, 显示 “☹” 表示冗余机处于监视状态。若没有使用双机热备则显示 “N/A”。
- ✧ **运行系统收发队列:** 网络精灵要发送给组态王 7.0 运行系统、或组态王 7.0 运行系统发送给网络精灵的数据队列长度。例如显示 100/200 (Send/Receive), 100 表示组态王 7.0 发送给网络精灵 100 包数据, 等待网络精灵发送到指定节点。200 表示网络精灵有 200 包数据等待组态王 7.0 运行系统处理。这里的一 “包” 数据可能是一

组实时数据或一组报警数据或其他可能传输数据的集合,用户可以不用关心。若 Send 参数有不断增长的趋势,则说明网络精灵发送数据的能力低于组态王 7.0 运行系统提供数据的能力,为避免出现此现象应保证本节点网络硬件设置与网络中其他节点相同(即同为 10M 或 100M 网络),并适当将“网络配置中”的“包大小”参数增大。若 Receive 参数有不断增长的趋势,则说明组态王 7.0 运行系统无法处理完网络精灵发送的数据,可能存在的原因有系统中有其他进程占用大量系统资源导致运行系统得不到资源进行数据处理,还有就是检查工程中是否有周期性且很短的时间内进行大量数据处理的地方。

21.7.2 客户端网络信息



图 21.37 网络精灵——客户端网络信息

当本地计算机作为客户端时,在网络精灵中将详细列出本机与远程站点(服务器端)的通讯信息。如图 21.37 所示,下面详细讲述各项参数的含义:

✧ **主机节点名:** 指远程站点的计算机名称或主网卡 IP 地址。

- ✧ **主机备份网卡：**远程站点存在双网络冗余时，该项会有值，指远程站点双网络冗余设置的从网卡的 IP 地址。
- ✧ **冗余机节点名：**远程站点使用双机热备时，该项会有值，指远程站点指定的备份机的计算机名称或备份机主网卡 IP 地址。
- ✧ **冗余机备份网卡：**远程站点同时存在双网络冗余和双机热备时，该项会有值，指远程站点指定的备份机的从网卡的 IP 地址。
- ✧ **本地链路：**显示本机与远程站点的通讯所使用的数据链路的 IP 地址。
- ✧ **发送字节：**显示本机向远程站点发送的数据量，单位：字节。
- ✧ **接收字节：**显示本机接收远程站点的数据量，单位：字节。
- ✧ **发送队列：**显示本机向远程节点要发送的数据队列长度。
- ✧ **本次连接时刻：**显示本机与某个远程站点最后一次正确连接的时间。

21.7.3 服务器网络信息

当远程计算机在“站点”中与本机建立了连接时，本机将作为远程站点的服务器，在网络精灵中将详细列出本机作为服务器与远程站点的通讯信息。如图 21.38 所示。

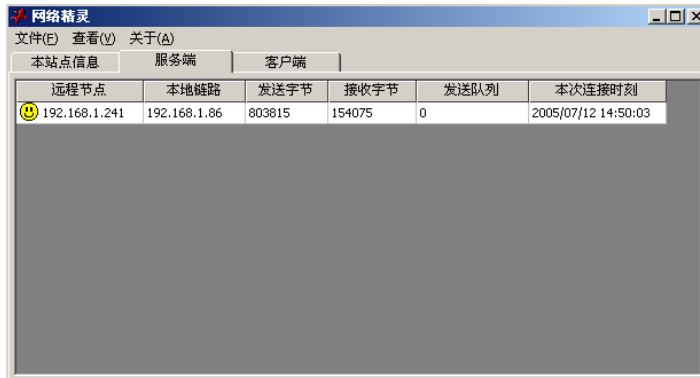


图 21.38 网络精灵——服务器网络信息

下面详细讲述各项参数的含义：

- ✧ **远程节点：**指远程节点的节点名称或 IP 地址。
- ✧ **本地链路：**显示本机与远程站点的通讯所使用的数据链路的 IP 地址。
- ✧ **发送字节：**指本机向远程站点发送的数据量，单位：字节。
- ✧ **接收字节：**指本机接收远程站点的数据量，单位：字节。
- ✧ **发送队列：**指本机向远程节点要发送的数据队列长度。
- ✧ **本次连接时刻：**指本机对某个远程节点最后一次正确连接的时间。

21.7.4 网络精灵菜单功能

- ✧ **文件—退出：**

网络精灵是一个独立进程，运行系统启动后启动网络精灵。在系统运行过程中进行此操作，精灵将提示“运行系统正在运行，是否强制退出”，如果选择退出，运行系

系统将中断与各远程站点的通讯，因此建议运行过程中不要将网络精灵退出。

✧ **查看—显示信息窗口：**

可以打开信息窗口，浏览网络精灵数据流信息。

✧ **查看—隐藏信息窗口：**

关闭已经打开的信息窗口。

✧ **查看—允许输出：**

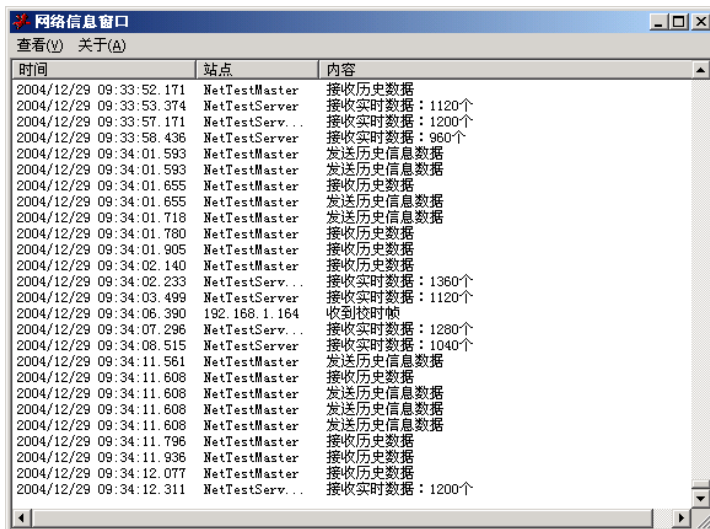
在信息窗口打开时，允许显示信息。

✧ **查看—禁止输出：**

在信息窗口打开时，禁止显示信息。

21.7.5 网络精灵信息窗口操作

网络精灵信息窗口显示本地计算机与网络中其他站点通讯情况，可以为分析通讯状态和查看数据流量提供参考。如图 21.39 所示。



时间	站点	内容
2004/12/29 09:33:52.171	NetTestMaster	接收历史数据
2004/12/29 09:33:53.374	NetTestServer	接收实时数据: 1120个
2004/12/29 09:33:57.171	NetTestServ...	接收实时数据: 1200个
2004/12/29 09:33:58.436	NetTestServer	接收实时数据: 960个
2004/12/29 09:34:01.593	NetTestMaster	发送历史信息数据
2004/12/29 09:34:01.593	NetTestMaster	发送历史信息数据
2004/12/29 09:34:01.655	NetTestMaster	接收历史数据
2004/12/29 09:34:01.655	NetTestMaster	发送历史信息数据
2004/12/29 09:34:01.718	NetTestMaster	发送历史信息数据
2004/12/29 09:34:01.780	NetTestMaster	接收历史数据
2004/12/29 09:34:01.905	NetTestMaster	接收历史数据
2004/12/29 09:34:02.140	NetTestMaster	接收历史数据
2004/12/29 09:34:02.233	NetTestServ...	接收实时数据: 1360个
2004/12/29 09:34:03.499	NetTestServer	接收实时数据: 1120个
2004/12/29 09:34:06.390	192.168.1.164	收到帧时帧
2004/12/29 09:34:07.296	NetTestServ...	接收实时数据: 1280个
2004/12/29 09:34:08.515	NetTestServer	接收实时数据: 1040个
2004/12/29 09:34:11.561	NetTestMaster	发送历史信息数据
2004/12/29 09:34:11.608	NetTestMaster	接收历史数据
2004/12/29 09:34:11.608	NetTestMaster	发送历史信息数据
2004/12/29 09:34:11.608	NetTestMaster	发送历史信息数据
2004/12/29 09:34:11.608	NetTestMaster	发送历史信息数据
2004/12/29 09:34:11.796	NetTestMaster	接收历史数据
2004/12/29 09:34:11.936	NetTestMaster	接收历史数据
2004/12/29 09:34:12.077	NetTestMaster	接收历史数据
2004/12/29 09:34:12.311	NetTestServ...	接收实时数据: 1200个

图 21.39 网络精灵——信息窗口

◆ 查看—选项:

可以选择显示哪些站点的通讯信息和定义通讯信息显示级别。如图 21.40 所示。如果站点名称前的复选框选中，则信息窗口显示与此站点的通讯信息，如果没有选中，则信息窗口不显示与此站点的通讯信息。

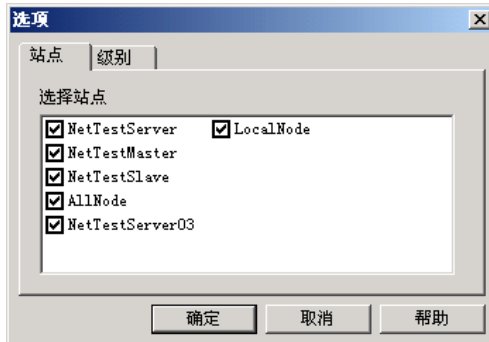


图 21.40 网络精灵信息窗口——站点选项

信息显示级别分为 1、2、3、4 级。1 级主要显示连接，断开连接，状态变化等信息；2 级主要显示初始化，请求消息，握手信号等信息；3 级主要显示错误信息；4 级显示所有信息。如图 21.41 所示。

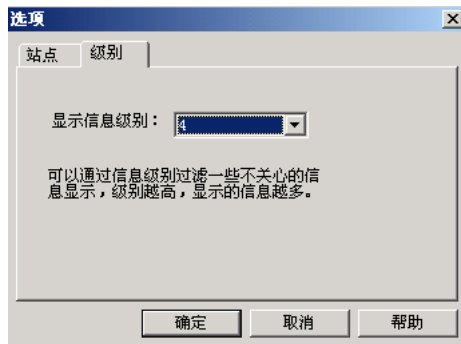


图 21.41 网络精灵信息窗口——级别选项

✧ **查看—允许滚动：**选中此项后，信息窗口将滚动显示通讯信息。

第二十二章 冗余系统

- ☞ 介绍组态王的冗余功能
- ☞ 教您如何实现双设备冗余、双机冗余和双网络冗余

组态王提供全面的冗余功能,能够有效地减少数据丢失的可能,增加了系统的可靠性,方便了系统维护。组态王提供三重意义上的冗余功能,即双设备冗余、双机热备和双网络冗余。本章将分三节分别详细讲述这三种冗余方式的特点和实现方法。

22.1 双设备冗余

22.1.1 双设备冗余概述

双设备冗余,是指设备对设备的冗余,即两台相同的设备之间的相互冗余。对于用户比较重要的数据采集系统,用户可以用两个完全一样的设备同时采集数据,并与组态王通讯。系统结构示意图如图 22.1 所示:

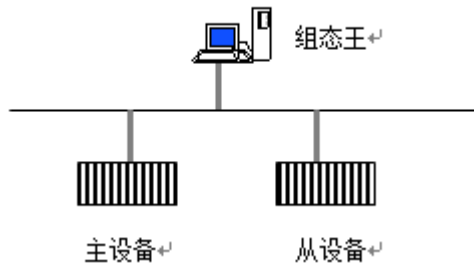


图 22.1 双设备冗余示意图

正常情况下，主设备与从设备同时采集数据，但组态王只与主设备通讯，若主设备通讯出现故障，组态王将自动断开与主设备的连接，与从设备建立连接，从设备由热备状态转入运行状态，组态王从从设备中采集数据。此后，组态王一边与从设备通讯，一边监视主设备的状态，当主设备恢复正常后，组态王自动停止与从设备的通讯，与主设备建立连接，进行通讯，从设备又转入热备状态。

这样就要求从设备与主设备应完全一样，即两台设备要完全处于热备状态。而且组态王中在定义该设备的 I/O 变量时，只能定义变量与主设备建立连接，而从设备无需定义变量，完全是对主设备的冗余。

22.1.2 双设备冗余的功能

具体地说双设备冗余主要是实现数据的不间断采集。

由于采用了设备冗余，因此一旦主设备通讯出现中断，从设备可以迅速将采集到的数据传给主设备继续与组态王进行通讯，从而保持数据的完整性。

22.1.3 双设备冗余的设置

22.1.3.1 双设备冗余设置的步骤

双设备冗余定义的步骤大致为：

- ◇ 从设备定义
- ◇ 主设备定义
- ◇ 变量定义（主设备）

双设备冗余设置一般先定义从设备，然后再定义主设备，定义主设备时可将已定义的设备定义为从设备。也可以同时将两个设备都定义，然后再指定主、从设备。

22.1.3.1.1 从设备定义

从设备的设置与一般的 I/O 设备设置方法相同，工程人员根据设备配置向导就可以完成从设备的配置，具体设置方法请参见“第六章 I/O 设备管理”。例如在 COM1 口上定义一个逻辑名为“从设备 7018”的 Nudam7018 模块设备，如图 22.2 所示。



图 22.2 定义好的从设备——从设备 7018


22.1.3.1.2 主设备定义

主设备的设置与一般的 I/O 设备设置方法也大致相同, 工程人员根据设备配置向导就可以完成主设备的配置, 具体设置方法请参见“第六章 I/O 设备管理”。



图 22.3 指定主设备的冗余设备

唯一不同的是，在“设备配置向导——逻辑名称”页中，需要指定主设备的冗余设备的逻辑名称，如上图 22.3 所示，选中“指定冗余设备”选项，“冗余设备逻辑名：”列表框就会变为可见，可从下拉列表中选出刚刚定义的设备“从设备 7018”作为该设备冗余设备。另外，对组态王来说，两个冗余的设备实际上是独立的两个设备。因此设备的地址不同。

 **注意：**

工程人员给要配置的主设备指定一个与从设备不同的逻辑名称。




例如：

在 COM2 口上定义一个逻辑名为“主设备 7018”的 Nudam7018 模块作为主设备，设备配置信息总结界面如图 22.4 所示。



图 22.4 定义好的主设备——主设备 7018

 注意:

- 1 从设备与主设备不止设备应完全一致，而且两者状态应完全一致，例如采集的数据，数据类型等等。
- 2 对于双设备冗余的设备驱动程序，需要在原有驱动程序的基础上稍加修改，如需要双设备冗余，请与亚控公司技术支持部联系。

22.1.3.1.3 变量定义

工程人员在数据词典中定义变量，必须要和主设备相连接，从设备不能定义任何变量。关于 I/O 变量的定义请参见“第五章 变量定义和管理”。

22.1.4 双设备冗余的使用

运行系统启动后，可以从组态王信息窗中看到设备初始化信息，和当主设备出现故障时，切换到从设备的提示信息。如：

```
运行系统: 打开通讯设备成功!  
运行系统: 设备初始化成功---主设备7018  
运行系统: 打开通讯设备成功!  
运行系统: 设备初始化成功---从设备7018  
运行系统: 开始记录历史数据!  
运行系统: 设备“主设备7018”通讯失败!  
运行系统: 主设备失效,启动从设备。
```

系统判断到主设备出现故障，直接启动从设备，实现了数据的不间断采集，保证了数据的完整性。

22.2 双机热备

双机热备其构造思想是主机和从机通过 TCP/IP 网络连接，正常情况下主机处于工作状态，从机处于监视状态，一旦从机发现主机异常，从机将会在很短的时间之内代替主机，完全实现主机的功能。例如，I/O 服务器的热备机将进行数据采集，报警服务器的冗余机将产生报警信息并负责将报警信息传送给客户端，历史记录服务器的冗余机将存储历史数据并负责将历史数据传送给客户端。当主机修复，重新启动后，从机检测到了主机的恢复，会自动将主机丢失的历史数据拷贝给主机，同时，将实时数据和报警缓冲区中的报警信息传递给主机，然后从机将重新处于监视状态。这样即使是发生了事故，系统也能保存一个相对完整的数据库、报警信息和历史数据等。

22.2.1 双机热备的功能

组态王的双机热备主要实现以下功能：

- ✧ 实时数据的冗余
- ✧ 历史数据的冗余
- ✧ 报警信息的冗余
- ✧ 用户登录列表的冗余

组态王提供了系统变量“\$双机热备状态”变量来表征主、从机的状态。在主机上，该变量的值为正数，在从机上，该变量的值为负数。在使用双机热备之前，应先进行双机热备的配置。



注意：

组态王实现双机热备时，主机与从机的组态王工程文件应该完全一致。

22.2.2 双机热备实现的原理

如图 22.5 所示，为双机热备的系统结构图。双机热备主要是实时数据、报警信息和变量历史记录的热备。主从机都正常工作时，主机从设备采集数据，并产生报警和事件信息。从机通过网络从主机获取实时数据和报警信息，而不会从设备读取或自己产生报警信息。主从机都各自记录变量历史数据。同时，从机通过网络监听主机，从机与主机之间的监听采取请求与应答的方式，从机以一定的时间间隔（冗余机心跳检测时间）向主机发出请求，主机应答表示工作正常，主机如果没有应答，从机将切断与主机的网络数据传输，转入活动状态，改由下位设备获取数据，并产生报警和事件信息。此后，从机还会定时监听主机状态，一旦主机恢复，就切换到热备状态。通过这种方式实现了热备。

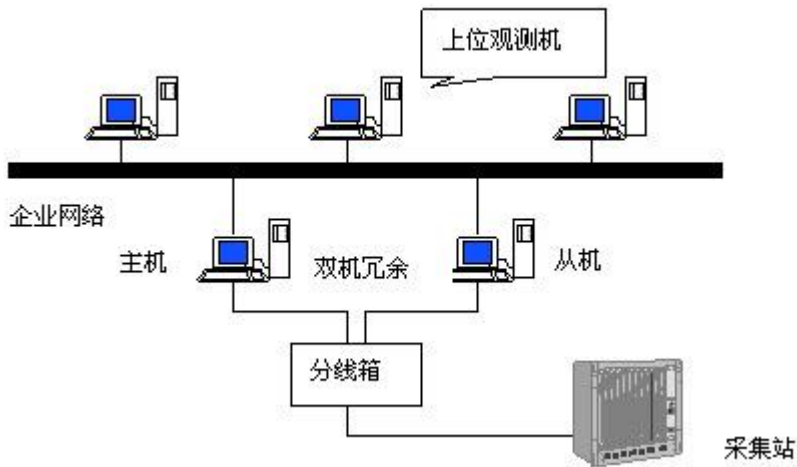


图 22.5 单机版双机热备系统结构

当主机正常运行，从机后启动时，主机先将实时数据和当前报警缓冲区中的报警和事件信息发送到从机上，完成实时数据的热备份。然后主从机同步，暂停变量历史数据记录，从机从主机上将所缺的历史记录文件通过网络拷贝到本地，完成历史数据的热备份。历史数据文件备份完成后，主从机转入正常工作状态。

当从机正常运行，主机后启动时，从机先将实时数据和当前报警缓冲区中的报警和事件信息发送到主机上，完成实时数据的热备份。然后主从机同步，暂停变量历史数据记录，主机从从机上将所缺的历史记录文件通过网络拷贝到本地，完成历史数据的热备份。历史数据文件备份完成后，主从机转入正常工作状态。

双机热备历史数据热备的结构图如图 22.6 所示。

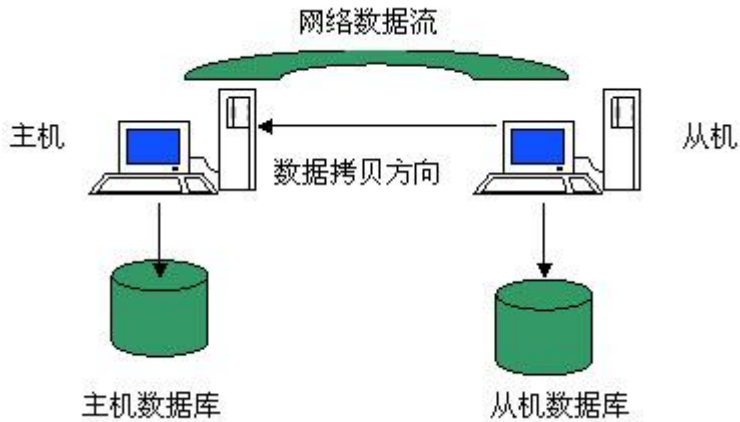


图 22.6 历史数据冗余

⚠ 注意:

- 1、主从机都正常工作时，从机上修改变量的值不会引起主机变量值的改变。
- 2、主从机都正常工作时，从机上的命令语言正常执行。
- 3、DDE 设备无法实现双机热备。

双机热备中，需要各台计算机保持时钟一致，这里就牵扯到校时服务器的概念，一般的设置是将主机和从机都设置为校时服务器，主机工作时主机采取广播的方式以一定的时间间隔向各台机器发送校时帧，保持网络时钟的统一。而当主机失效时，从机将代替主机成为网络的校时服务器。

22.2.3 网络工程的冗余

对于网络工程，即整个工程的所有功能分别由专用服务器来完成时，可以根据系统的重要性来决定对哪些服务器采取冗余，例如对于实时数据采集非常重要，而历史数据和报

警信息不是很重要的系统来说，可以只对 I/O 服务器设置冗余，如果历史数据和报警信息也同样重要的话，则需要分别设置 I/O 服务器、历史记录服务器和报警服务器的冗余机。下面将详细介绍其冗余的实现。网络结构示意图如图 22.7：

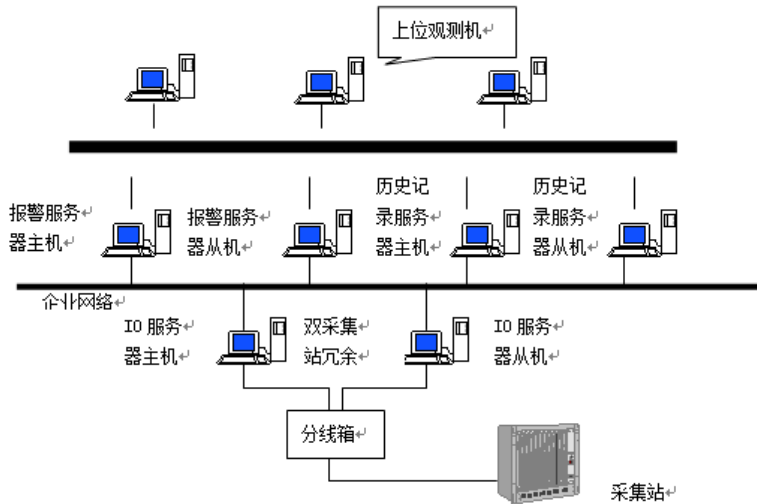


图 22.7 I/O、报警、历史记录服务器的冗余

在这种网络结构和冗余结构中，实时数据的冗余由 I/O 服务器主机和 I/O 服务器从机来完成，实时数据的冗余与单机版工程的实时数据冗余相同；历史数据的冗余由历史记录服务器主机和历史记录服务器从机来完成；报警信息的冗余由报警服务器主机和报警服务器从机来完成。

对报警服务器和历史记录服务器的冗余来说，从机没有监视状态，即报警服务器从机和历史记录服务器从机与它们的主机一样，同时从 I/O 服务器上取数据，在报警服务器从机上同样生成报警信息，并在报警服务器主机停止工作时，向客户机传送报警信息；在历史记录服务器上同样存储历史记录，并在历史记录服务器主机停止工作时，向客户机传送

历史数据。

对于客户端来说，只需要指定其 I/O 服务器、报警服务器和历史记录服务器的主机，当这些主机出现故障时，客户端会自动转为与相应的从机通讯。

22.2.4 双机热备配置

双机热备配置的三个要素

- ◇ 主机网络配置。
- ◇ 从机网络配置。
- ◇ 变量“\$双机热备状态”的使用。

22.2.4.1 主机网络配置

◇ **第一步：**

在主机上选择组态王工程浏览器中的“网络配置”项，双击该项，弹出如图 22.8 所示“网络配置”对话框，选择“连网”模式，在“本机节点名”处键入本机名称或 IP 地址。在双机热备一栏中，选择“使用双机热备”选项，后面的“本站为主站”和“本站为从站”选项变为有效。选择“本站为主站”。

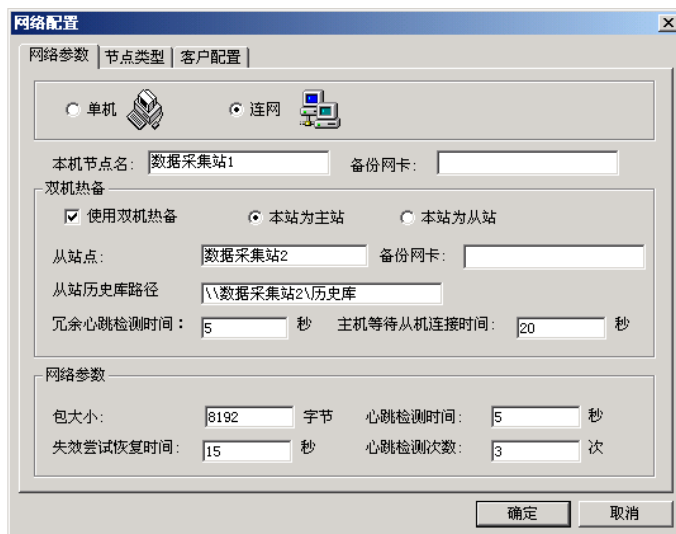


图 22.8 主机网络配置

◇ **第二步：**

在“从站点”后的编辑框中输入从站的名称或 IP 地址。在“从站历史数据库路径”一栏中键入从机保存历史数据的完整路径，该路径的定义格式采用 UNC 格式。该路径在从机上应该提供共享，这里添加的路径也是通过网络可以看到的有效路径。“网络参数”的设置请参见“第二十一章 网络功能”。

◇ **第三步：**

在“冗余机心跳检测”处输入主机检测从机的时间间隔。如图 22.8 所示，缺省为 5 秒。

◇ **第四步：**

节点类型设置，在“网络配置”中单击“节点类型”属性页，如图 22.9 所示。

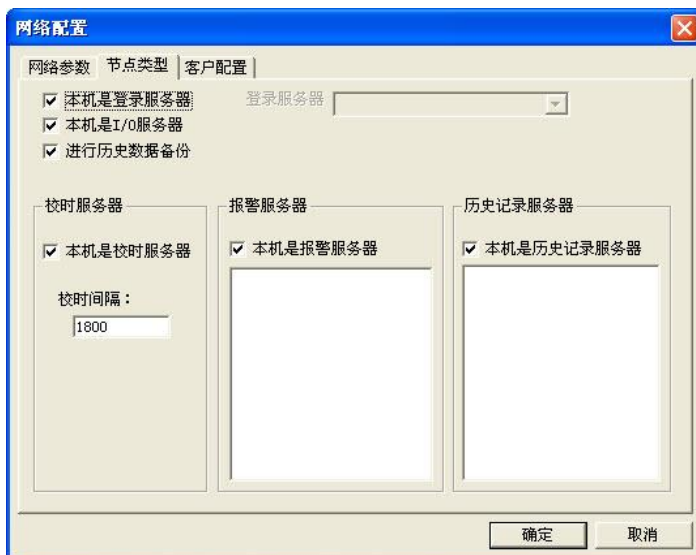


图 22.9 校时服务器设置

按照实际情况选择本站点的类型（请参见第二十一章 网络功能），一般如果是单机，各选项都要选择。选择“本机是校时服务器”，同时输入“校时间隔”，表示本机发送校时信息的时间间隔。默认为 1800 秒。

◇ 第五步

单击“确定”，完成了双机热备中主机的基本配置。

22.2.4.2 从机网络配置

◇ 第一步：

在从机上选择组态王工程浏览器中的“网络配置”项，双击该项，弹出“网络配置”对话框，如图 22.10。选择“连网”模式，在“本机节点名”处键入从机名称或 IP

地址，在双机热备一栏中，选择“使用双机热备”选项。其后面的“本机为主站”和“本机为从站”选项变为有效，选择“本站为从机”。

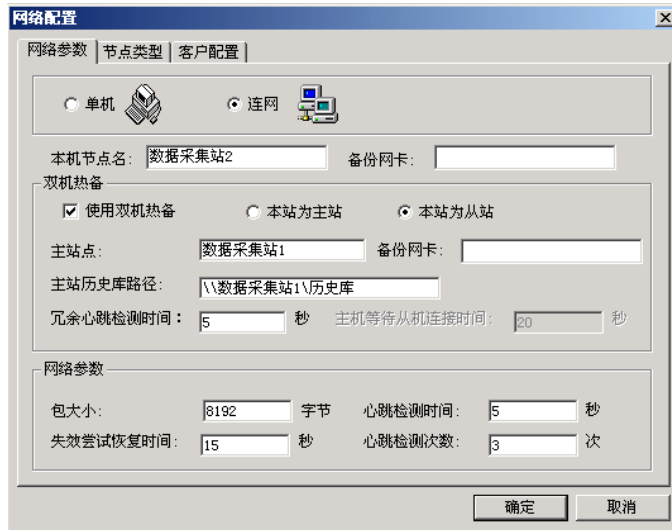


图 22.10 从机网络配置

◇ **第二步：**

在“主站名称”处键入主站名称或 IP 地址。在“主站历史数据库路径”处键入主机保存历史数据的完整路径，该路径的定义格式采用 UNC 格式。该路径在主机上应该提供共享，这里添加的路径也是通过网络可以看到的有效路径。

◇ **第三步：**

在“冗余机心跳检测”处输入从机监听主机的时间间隔。如图 22.10，缺省为 5 秒。

◇ **第四步：**

单击“确定”，完成了双机热备中从机的基本配置。

22.2.4.3 双机热备状态变量的使用

系统变量“\$双机热备状态”变量用来表征主从机的状态。在主机上，该变量的值为正数，在从机上，该变量的值为负数。

主机状态监控：


在主机的组态王工程中，可通过变量“\$双机热备状态”对主机进行监控。变量“\$双机热备状态”有以下几种状态：

- ◇ \$双机热备状态=1，此时主机状态正常。
- ◇ \$双机热备状态=2，此时主机状态异常，主机将停止工作，并不再响应从机的查询。

从机状态监控：

在从机的组态王工程中，可通过变量“\$双机热备状态”对从机进行监控。变量“\$双机热备状态”有以下几种状态：

- ◇ \$双机热备状态= -1，此时从机检测到主机状态正常。
- ◇ \$双机热备状态= -2，此时从机检测到主机状态异常，主机工作异常，从机代替主机成为主站。


 注意：

- 1、 主机与从机的双机热备状态的返回是不一样的，主机为正值，从机为负值。
- 2、 可以通过强制转换\$双机热备状态进行手动切换。但要慎重使用。

手动状态切换：

特殊情况下，可以通过强制\$双机热备状态，实现主，从机之间的手动切换。

主机切换到从机：强制主机的\$双机热备状态为 2，主机停止工作，并停止响应从机查询，从机认为主机故障，启动工作，此时主机将没有任何工作，同时主机的数据也将不再变化。当强制主机的\$双机热备状态为 1 后，又能实现从机向主机的切换。

 注意：

从机的双机热备状态不能任意强制。若主机正常，强制从机的\$双机热备状态为-2，则会出现混乱。若主机异常，强制从机的\$双机热备状态为-1，则会丢失下次监听查询前的数据。

22.3 双网络冗余

双网络冗余实现了组态王系统间两条物理网络的连接，以防单一网络系统中网络出现故障则所有站点瘫痪的弊端。对于网络的任意一个站点均安装两块网卡，并分别设置在两个网段内。当主网线路中断时，组态王网络通讯自动切换到从网，保证通讯链路不中断，为系统稳定可靠运行提供了保障。系统结构示意图如图 22.11 所示：

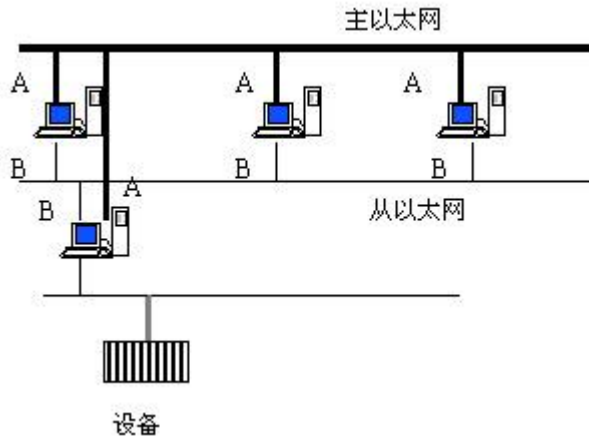



图 22.11 双网络冗余示意图

如图示，粗线表示主网，细线表示从网。A 表示主网网卡，B 表示从网网卡。网络上的任意一台机器均需要安装两个网卡，在实际使用中一般将这两块网卡分别设置在两个网段内，例如，A 网卡的 IP 地址均设置为 100.100.100.*，最后一位数字各台机器不相同，子码掩码为：255.255.255.0；B 网卡的 IP 地址均设置为 200.200.200.*，最后一位数字各台机器不相同，子码掩码为：255.255.255.0，这样就构造了两个以太网，各个站点将通过相同网段的网卡进行通讯。

 注意：

需要保证的一点是本节点两块网卡可以正确的与网络上的其他节点的任何 IP 能够 PING 通。

22.3.1 双网络冗余配置

22.3.1.1 网卡的配置

下面以“数据采集站”为例详细介绍双网络冗余的配置。由于每个站点均配置有两块网卡，因此需要设置两块网卡的 IP 地址，从计算机的“控制面板”中双击网络图标，进入“配置”，选中 TCP/IP 协议，单击属性，弹出对话框如如图 22.12 所示。

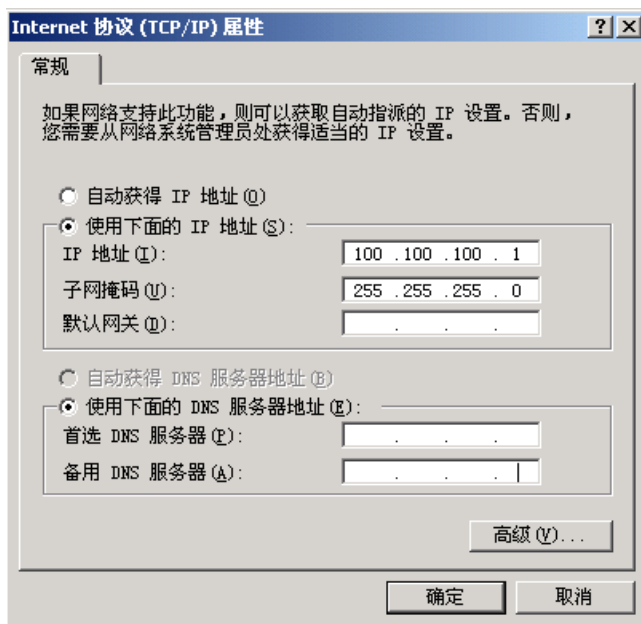


图 22.12 设置网卡的 IP 地址

在适配器下的列表框中会自动列出两个网卡，选中主网网卡，在 IP 地址栏中指定 IP 地址为：100.100.100.1，子网掩码为：255.255.255.0，默认网关可先不设置。然后从适配器下的列表框中选中从网网卡，在 IP 地址栏中指定 IP 地址为：200.200.200.1，子网

掩码为：255.255.255.0，默认网关可先不设置。

对于网络的其他每个站点都这样设置，例如“报警数据站”，网卡 1 的 IP 地址为 100.100.100.2，子网掩码为 255.255.255.0，网卡 2 的 IP 地址为：200.200.200.2，子网掩码为 255.255.255.0。

22.3.1.2 组态王网络配置

在“数据采集站”上，选择组态王工程浏览器中的“网络配置”项，双击该项，弹出如图 22.13 所示对话框，选择“连网”，在“本机节点名”处键入本机名称，例如“数据采集站”，或者是本机的主网网卡的 IP 地址，例如“100.100.100.1”，在备份网卡中输入网卡 2 的 IP 地址，例如“200.200.200.1”。

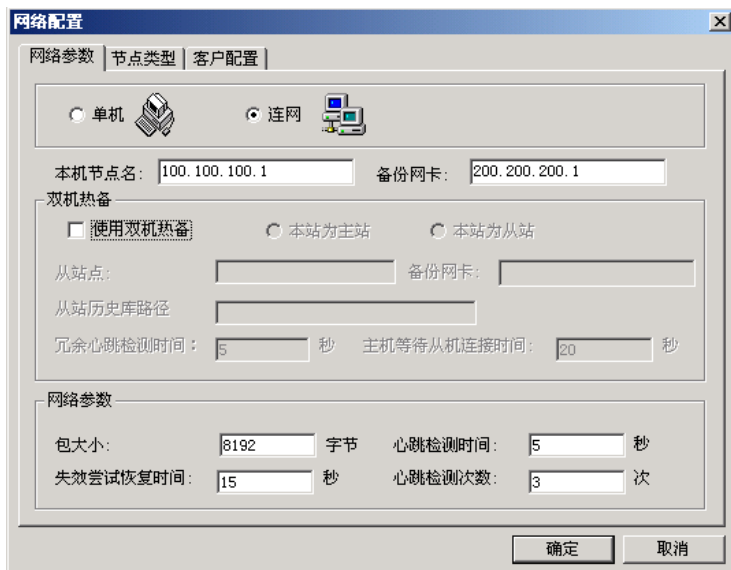


图 22.13 设置网络参数

对于网络上的其他每个站点都这样设置，例如“报警数据站”，本机节点名为“报警数据站”，或者输入 IP 地址“100.100.100.2”，备份网卡输入“200.200.200.2”。



注意：

只有在构造了局域网的条件下，“本机节点名”中输入机器名才有效，否则只能用 IP 地址，并且在有备份网卡的情况下，“本机节点名”只能是主网网卡的 IP 地址。

当主网出现故障时，将切换到从网通讯；当一个站点由于一个网卡或一段网线出现故障，而与其他站点的网络通讯出现故障时，它的备份网卡将切换到工作状态，例如“数据采集站”的 A 网卡出现故障时，它的 B 网卡将与“报警数据站”等网络上的其他站点通过从网进行通讯。

对于双设备、双机和双网络冗余这三种冗余方式，设计者可综合运用，可以同时采取三种冗余方式或采取其中的任意一种或两种。采用冗余后，系统运行时将更加稳定、可靠，对各种情况都能应付自如。

第二十三章 Web 全新版

- ☞ 介绍组态王的 Web 全新版
- ☞ 与 Web 全新版相关的工程配置
- ☞ Web 全新版的发布示例
- ☞ 如何在 IE 中浏览工程

23.1 Web 全新版简介

随着 Internet 科技日益渗透到生活、生产的各个领域，传统自动化软件的 e 趋势已发展成为整合 IT 与工业自动化的关键。组态王 7.0 提供了 Web 全新版，Web 全新版基于 ActiveX 技术，采用 B/S 结构，客户可以随时随地通过 Internet/Intranet 实现远程监控。客户端有着强大的自主功能，Internet/Intranet 网络上的任何一台 PC 机都可以通过 IE 浏览器浏览工业现场的实时画面，监控各种工业数据，实现了对客户信息服务的动态性、实时性和交互性。

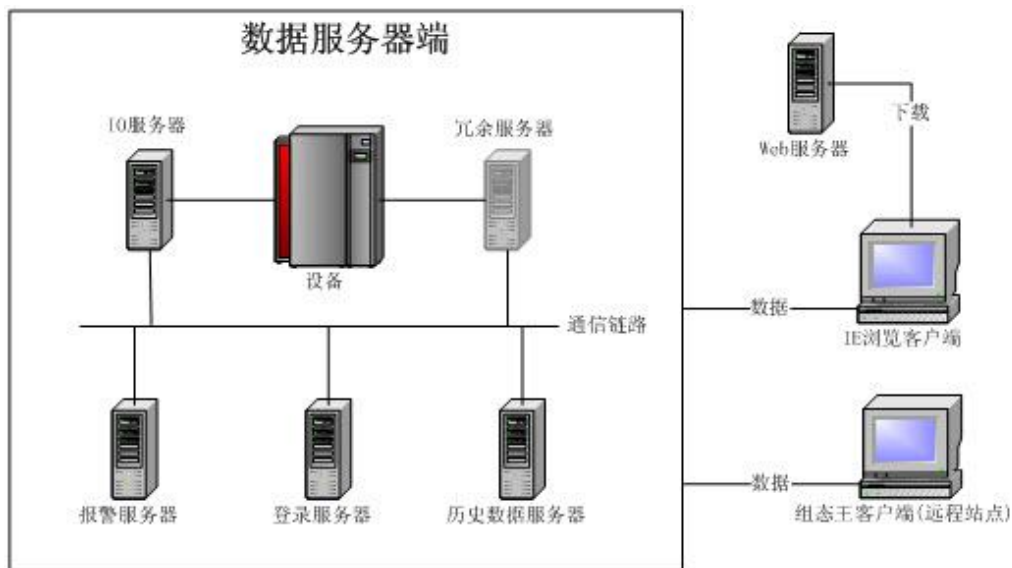
23.1.1 Web 全新版功能介绍

全新版的 Web 发布可以实现画面发布，数据发布和 OCX 控件发布，同时保留了组态王 7.0Web 的所有功能：IE 浏览客户端可以获得与组态王 7.0 运行系统相同的监控画面，IE 客户端与 Web 服务器保持高效的数据同步，通过网络您可以在任何地方获得与 Web 服务器

上相同的画面和数据显示、报表显示、报警显示等，同时可以方便快捷的向工业现场发布控制命令，实现实时控制的功能。另外，Web 全新版还具有以下几点技术特性：

- Web 全新版通过几个独立的文件实现工程发布、工程下载、画面展示和数据访问等功能；
- Web 全新版采用 ActiveX 技术，可以支持组态王 7.0 的内部控件、Windows 通用控件以及客户自定义控件的发布；
- Web 全新版实现了组态王 7.0 运行程序的大部分功能，部分 Web 全新版不支持的功能和函数请参考本章最后一节——Web 全新版支持与不支持的功能；
- Web 全新版采用组态王 7.0 远程站点的实现原理，Web 服务器能够直接引用数据服务器端的变量；
- Web 全新版需要单独建立一个组态王 7.0 工程作为 Web 服务器，并将其发布到 IIS 服务器中。

全新版的 Web 发布实现原理如下图所示：



◇ **数据服务器端：**

为客户端提供实时数据、报警数据和历史数据等。数据服务器是由 IO 服务器、冗余服务器、报警服务器、登录服务器以及历史数据服务器组成的，可以用一个组态王 7.0 工程充当所需要的所有的数据服务器，也可以由不同的组态王 7.0 工程分别实现各个服务器的功能，并且可以在一台或者多台计算机上进行组态王 7.0 工程的配置；

◇ **Web 服务器：**

组态王 7.0 的数据服务器不能直接充当 WEB 发布服务器。要求组态王 7.0 的 WEB 服务器为组态王 7.0 客户端工程(该客户端工程要求能通过远程站点的方式引用运行于数据服务器端的数据)，同时要求该客户端工程所在计算机必须安装 IIS 信息服务。

◇ **IE 浏览客户端：**

Internet/Intranet 网络中的任意一台计算机都可以下载客户端工程，这些计算机被定义为 IE 浏览客户端，用户可以在客户端上浏览工程中定义的画面，如果画面中引用了 Web 服务器中的变量，那么客户端就会从 Web 服务器上读取变量并实时更新画面。

23.1.2 Web 全新版与 Web 功能的比较

目前 Kingview 中的 Web 功能只能实现画面和数据的发布，无法实现 OCX 控件的发布，为了满足客户的需求，在 Kingview 中增加了 Web 全新版，全新版的 Web 采用 ActiveX 技术，彻底解决了长期以来 Kingview 的 Java 框架无法发布 OCX 控件的问题，使远程用户能够通过 IE 浏览访问服务器。Web 全新版与 Web 的主要区别如下：

- Web 是嵌入在组态王 7.0 内部的模块，而 Web 全新版是依赖几个独立的文件实现发布功能的；
- Web 采用 Java 机制实现发布功能；Web 全新版采用 ActiveX 技术实现发布功能；
- Web 全新版实现了组态王 7.0 运行程序的大部分功能，可以支持组态王 7.0 中大部分脚本和函数，原来的 Web 中不能实现的功能、脚本和函数可以在全新版的 Web 中实现；
- Web 全新版需要用户单独建立一个组态王 7.0 工程作为 Web 服务器，用户可以根据自己的需要对新的组态王 7.0 工程进行配置。

23.1.3 相关文件介绍

目前全新版的 Web 发布程序提供的文件包括：Touchview0cx.ocx 控件、KVWebPublish.exe、KVDownload.ocx 控件和 KV0cxDependFileList.txt 文件。下面对各个文件的作用和功能进行简要介绍。

- **Touchview0cx.ocx 控件：**

组态王 7.0 主程序的 OCX 封装，使用户能够在客户端查看发布后的 Kingview 工程；

- **KVWebPublish.exe：**

Web 发布应用程序，双击该应用程序，弹出如下图所示的对话框，在对话框中进行相应的配置后，即可实现 Kingview 工程的 Web 发布。

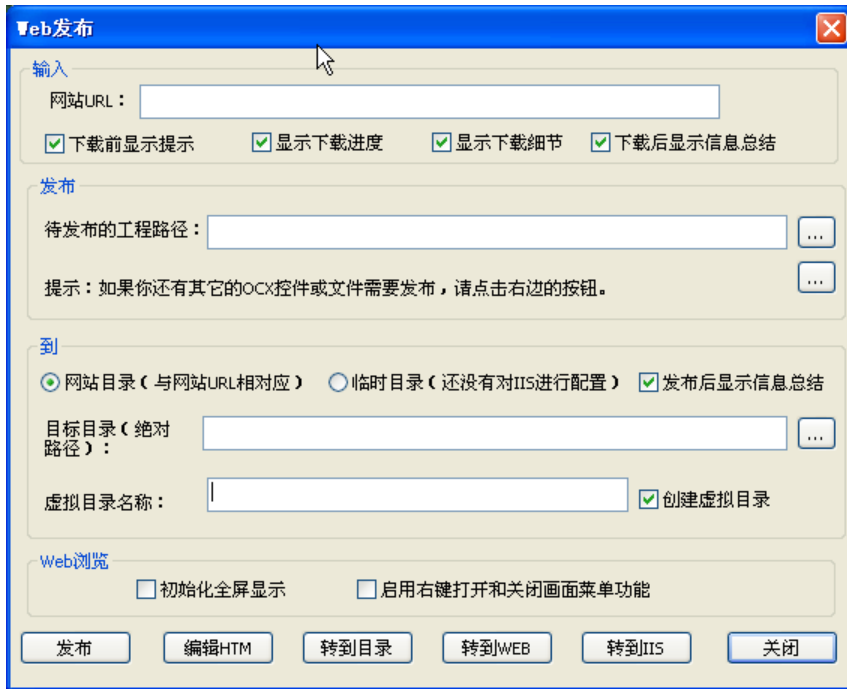


图 23.2 Web 发布设置对话框

- **KVDownload.ocx 控件:**

实现工程下载的 OCX 控件; Web 发布成功后, 该控件被加载到相应的 IIS 虚拟目录中, 登录 Web 发布网址之后, 该控件被载入到 IE 中执行下载任务。

- **KV0cxDependFileList.xml 文件:**

Web 发布列表, 文件中包含了组态王 7.0 中常见的需要发布的目标 OCX 控件及其依赖的 DLL, 不同控件之间需要换行隔开; 如果用户使用了其他 Windows 控件或者自定义控件, 可以以记事本的方式打开这个文件, 按照其中文件定义的格式添加控件即可。

Web 发布之前，读取该文件，并获得目标 OCX 及其依赖的 DLL，清单中列举的文件与实际的控制件完全一致，才可以对组态王 7.0 工程进行发布，否则，提示错误。

23.2 工程配置

根据上一节图 23.1 所示，为了实现 Web 发布，必须选择相应的组态王 7.0 工程作为数据服务器，并且需要在 Web 服务器中单独建立一个组态王 7.0 工程。对各个组态王 7.0 工程进行配置的时候，需要注意以下两点：

- Web 服务器中的登录服务器、IO 服务器、历史数据服务器都来自网络上的其他计算机，即 Web 服务器中组态王 7.0 工程的网络配置与数据服务器中组态王 7.0 工程的网络配置不同；
- Web 服务器中的组态王 7.0 工程画面可以和数据服务器端的画面一致，但是 Web 服务器上的脚本和变量中不能使用 IO 变量；

一般情况下，全新版的 Web 发布需要使用三台计算机，分别作为数据服务器、Web 服务器和 IE 浏览客户端，三台计算机的作用如下图所示：

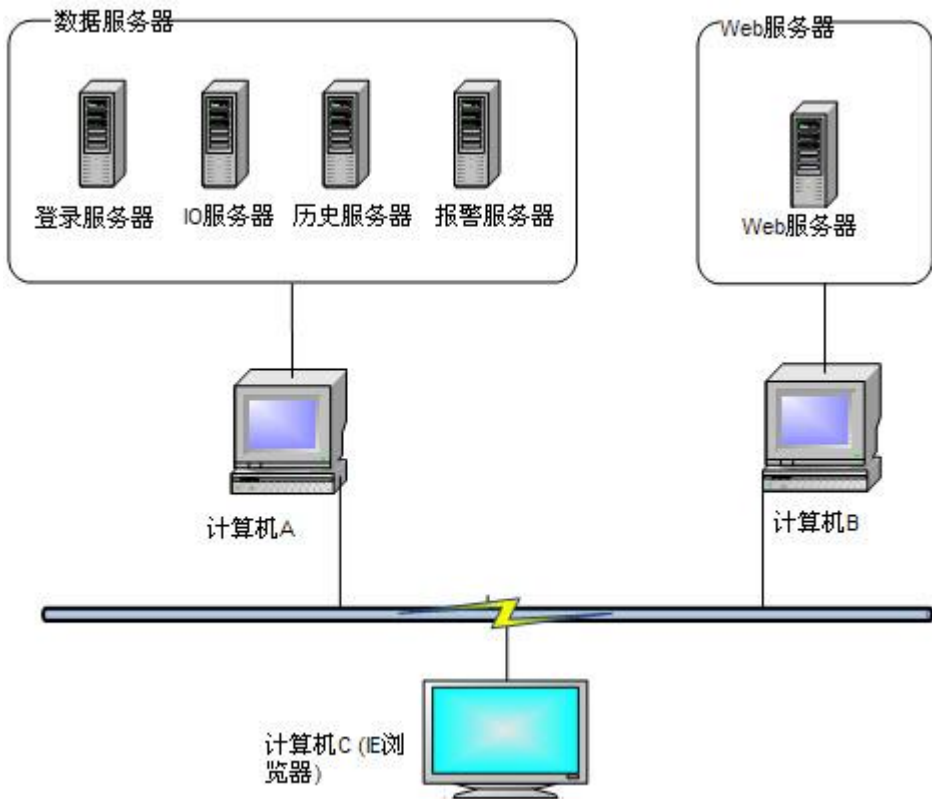


图 23.3 Web 发布实现过程

另外，为了满足不同用户的需求，全新版的 Web 发布允许用户使用一台计算机作为数据服务器和 Web 服务器，采用这种方式发布组态王 7.0 工程时，数据服务器和 Web 服务器的结构关系如下图所示：

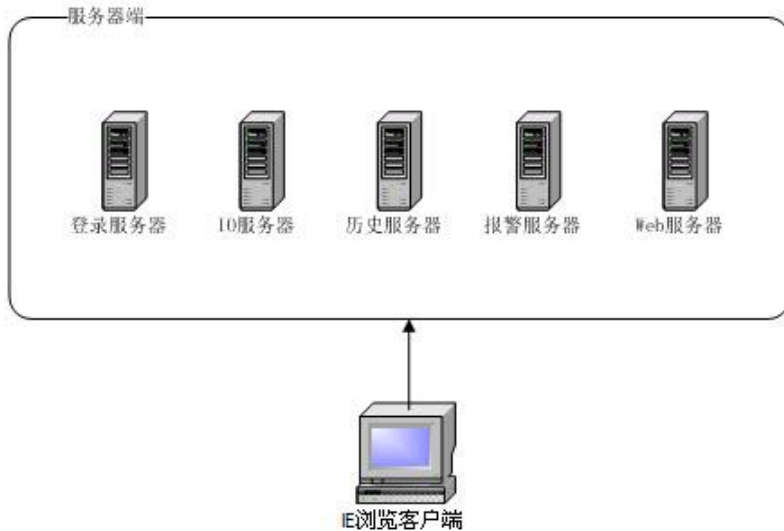


图 23.4 在一台计算机上实现 Web 发布

Web 发布之前，需要对数据服务器和 Web 服务器中的工程进行相应的配置。

⚠ 注意：

Web 服务器中的组态王 7.0 工程中不能使用 IO 变量，也不要定义 IO 设备，不能使用驱动、OPC、DDE 等相关功能。

23.2.1 数据服务器中工程的网络配置

打开 Kingview 工程浏览器，在工程浏览器中选择一个需要发布的工程的数据服务器工程，如下图所示：

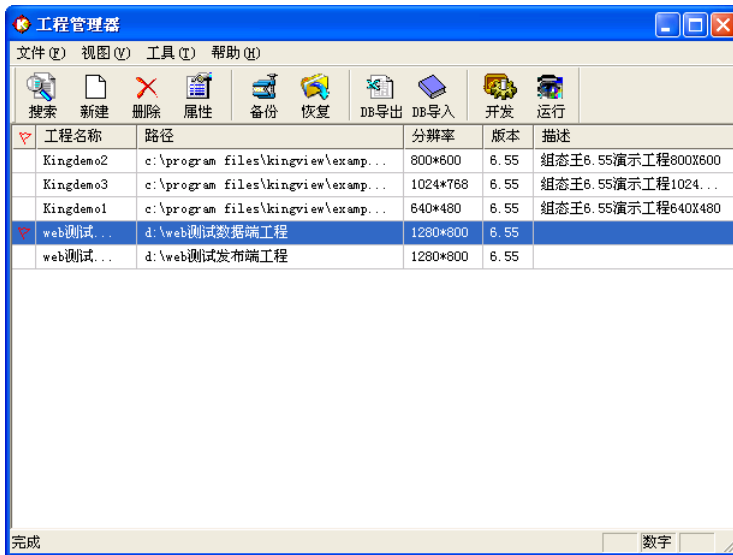


图 23.5 选择待发布的工程


双击选中的工程，或者单击快捷菜单中的图标  进入组态王 7.0 工程浏览器界面，在工程浏览器的目录显示区中双击系统配置->网络配置，或者选择菜单“配置->网络配置”，弹出“网络配置”对话框，如下图所示：



图 23.6 网络配置对话框

网络配置对话框中有三个属性页：网络参数、节点类型和客户配置，下面对各个属性页中的配置进行详细说明：

1、网络参数属性页

在网络参数属性页中选择“连网”模式，在本机节点名中输入本机所对应节点名，建议使用本地主网卡的 IP 地址作为本机节点名，如下图所示：



图 23.7 网络参数配置

2、节点类型属性页

如果一个组态王 7.0 工程充当数据服务器端所需要的所有服务器，节点类型属性页中的配置如下：

选中“本机是登录服务器”、“本机是 I/O 服务器”、“进行历史数据备份”、“本机是实时服务器”、“本机是报警服务器”和“本机是历史记录服务器”，用本机充当 IO 服务器、登录服务器、报警服务器、历史记录服务器等客户端所需要的所有服务器，如下图所示：

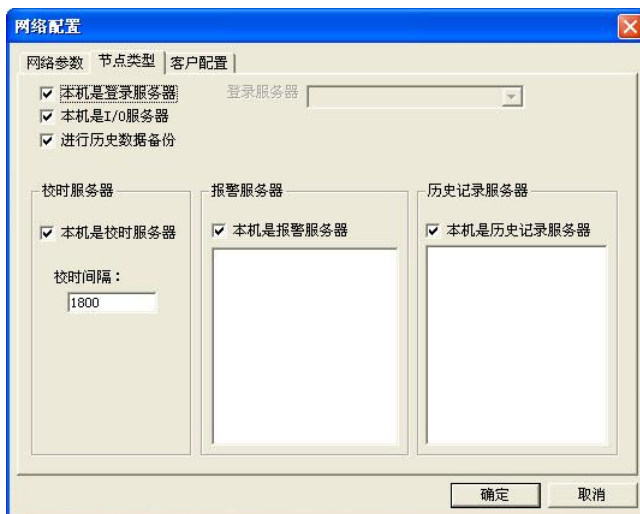



图 23.8 配置节点类型

 注意：

也可以分别采用不同的工程作为 I/O 服务器、报警服务器、历史服务器、登录服务器等，在每个工程的网络配置中分别进行配置。

3、客户配置属性页

该属性页中的配置项与组态王 7.0 工程的 Web 发布无关，因此不需要另外进行配置。网络配置中的各项配置完成之后，单击确定按钮，完成对数据服务端的网络配置。数据服务端的网络配置完成之后，需要共享网络服务工程所在的文件夹，以保证 Web 服务器能够顺利引用数据服务器中的变量。

 注意:

各个配置项的具体含义请参见 21.1 网络配置, 在此不再赘述。

23.2.2 Web 服务器中工程的网络配置

为了实现组态王 7.0 工程的 Web 发布, 首先需要在 Web 服务器上单独建立一个组态王 7.0 工程, 建立工程的方法请参见 3.2 如何新建一个工程, 在此不再赘述, 新建的组态王 7.0 工程如下所示:

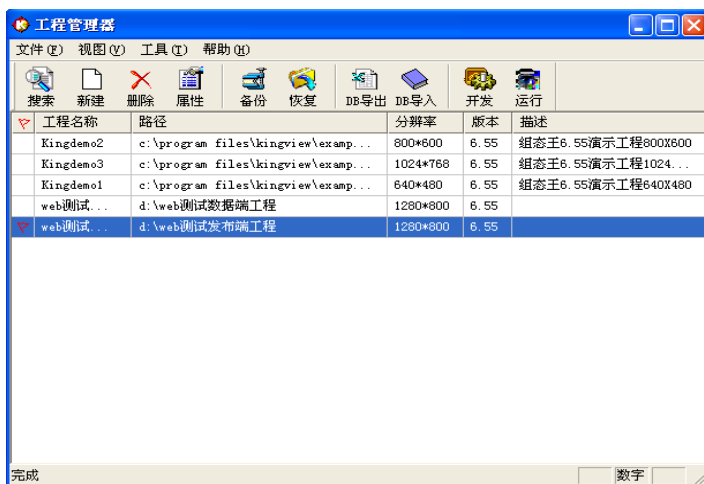



图 23.9 Web 客户端工程

双击选中的工程, 或者单击快捷菜单中的图标  进入组态王 7.0 工程浏览器界面, 在工程浏览器中建立 We 服务器与数据服务器的网络连接。

首先，在 Web 服务器中建立数据服务器的远程站点。在工程浏览器的左边选择“站点”标签，进入站点管理界面。在左侧的节点名称列表区域中单击鼠标右键，在弹出的快捷菜单中选择“新建远程站点”，弹出远程节点对话框，如下图所示：

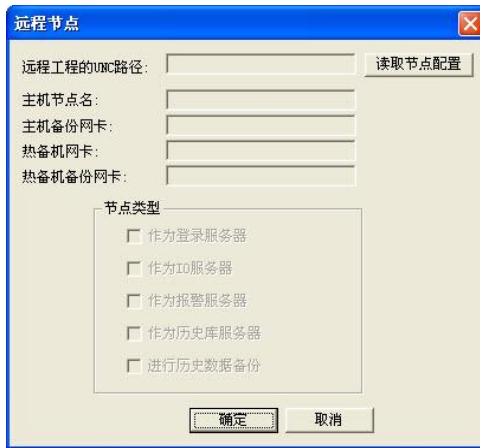


图 23.10 远程节点配置对话框

单击远程节点对话框中的“读取节点配置”按钮，在弹出的对话框中选择远程工程的路径，即数据服务器中的组态王 7.0 工程路径，如下图所示：



图 23.11 选择远程工程路径

选择完成后，该远程站点的信息就会被全部读出来，自动添加到对话框中对应的剩下的各项中，如主机节点名、节点类型等，都会自动读取并添加的。定义完成后，单击“确定”按钮关闭对话框，如下图所示：

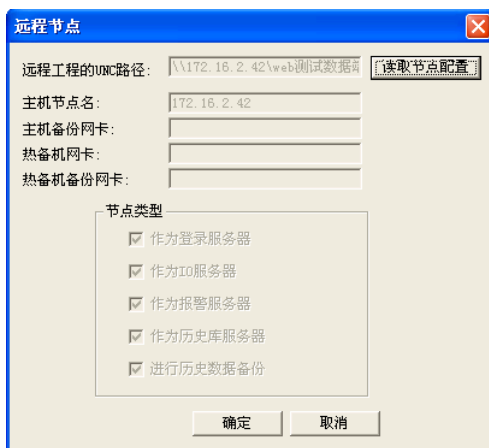


图 23.12 远程节点内容

节点建立后，在工程浏览器——站点的站点列表区和站点信息区会显示出该站点的所有信息。


 注意：

图 23.12 中选择的远程工程，即数据服务器中的组态王 7.0 工程，是用一个组态王 7.0 工程作为登录服务器、IO 服务器、报警服务器、历史库服务器的。

接下来，对 Web 服务器中的组态王 7.0 工程进行网络配置，在工程浏览器的目录显示区中双击系统配置->网络配置，或者选择菜单“配置->网络配置”，弹出“网络配置”对话框，如图 23.13 所示。

在网络配置对话框中，对三个属性页中的属性分别进行配置，配置过程如下：


1、网络参数

在网络参数属性页中选择“连网”，在“本机节点名”后面的编辑框中输入本机所对应节点名，建议使用本地主网卡的 IP 地址作为本机节点名，本例中使用一台计算机作为数据服务器和 Web 服务器，网络参数属性页的配置如下所示：



图 23.13 网络参数属性页配置

2、节点类型

在节点类型属性页中，只需要为工程设置登录服务器，单击登录服务器编辑框中的按钮，下拉菜单中列出了所有可作为登录服务器的节点名，选择数据服务端表示登录服务器的节点名即可。

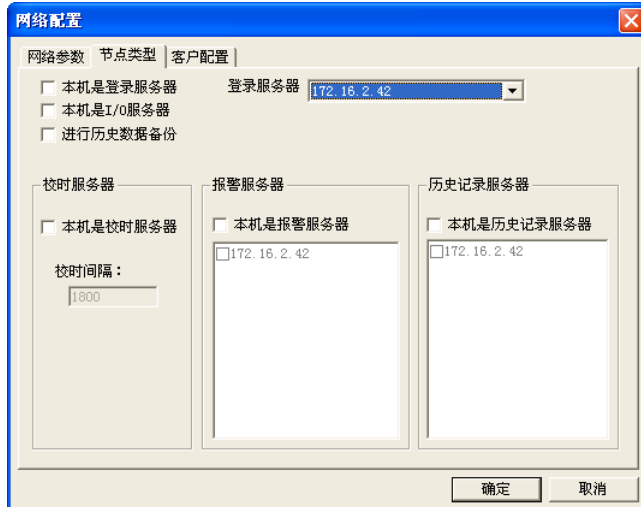


图 23.14 节点类型属性页配置

3、客户配置

在客户配置属性页中，选中“客户”复选框，各个服务器及其下面的服务器列表被激活，由于本例中采用一台计算机作为 I/O 服务器、报警服务器、历史服务器及其他所需的所有服务器，因此服务器列表都是相同的，将各个服务器下面的服务器列表全部选中，如下图所示：

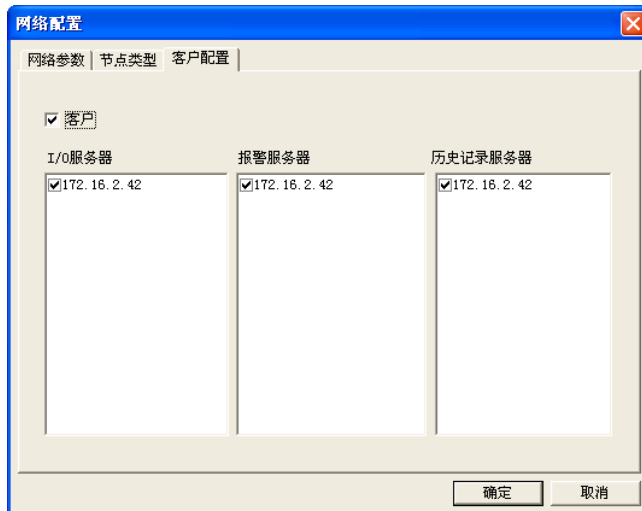


图 23.15 客户配置属性页配置

各项配置完成之后，单击确定按钮完成 Web 服务器中的组态王 7.0 工程的网络配置。此时，在工程浏览器的左边选择“站点”标签，进入站点管理界面。展开数据服务端对应的远程节点，单击“数据词典”，在右侧的内容显示区中可以看到远程数据服务器上的所有变量。

对于公网发布而且数据服务器和 web 服务器是通过路由器把部分端口映射到公网的情况，数据服务器应该映射 2001 端口(固定、不可修改)、2002 端口(固定，不可修改)；web 服务器映射 80 端口(用户可修改)。Web 发布工程除了要做以上网络配置，还要使用 WebTool 工具做配置。打开 WebTool 工具，如图 23.16 所示。

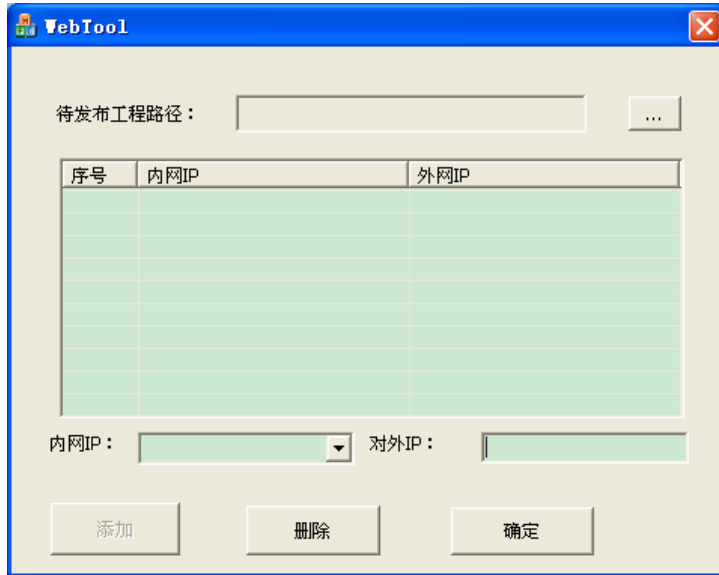


图 23.16 WebTool

单击“待发布工程路径”后的按钮选择待发布的工程，在对外 IP 后面的编辑框内填写数据服务对外映射的 IP 地址，并单击“添加”按钮如图 23.17 所示。

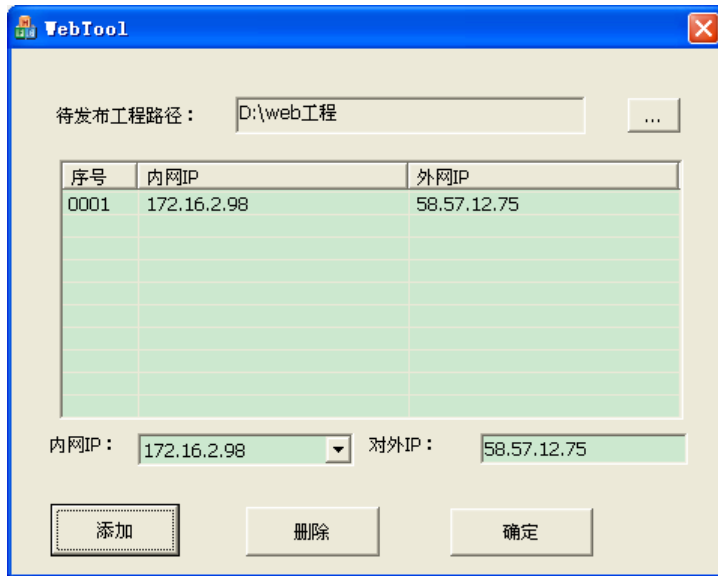


图 23.17 映射配置


数据服务器和 Web 服务器中的组态王 7.0 工程的网络配置完成之后,就建立了数据服务端和 Web 服务器之间的链接,Web 服务器的变量可以直接引用数据服务端的任意一个变量,也可以在 Web 服务器的组态王 7.0 工程中新建其他内存变量。

例如,在 Web 服务器的组态王 7.0 工程画面中,新建一个报表控件,并在报表控件中添加变量,添加过程如下所示:

1、单击报表控件的单元格,弹出报表工具箱,如下图所示:



图 23.18 报表工具箱

2、在报表工具箱的编辑框中直接输入需要进行显示的变量或者单击报表工具箱中的快捷按钮 ，弹出变量选择对话框。在变量选择对话框中，列出了本站点和远程站点中所有可供选择的变量。单击对话框左侧显示区中列出的站点名，右侧的变量显示区中就会显示相应站点中定义的所有变量，在对话框中选择需要进行显示的变量如下图所示：

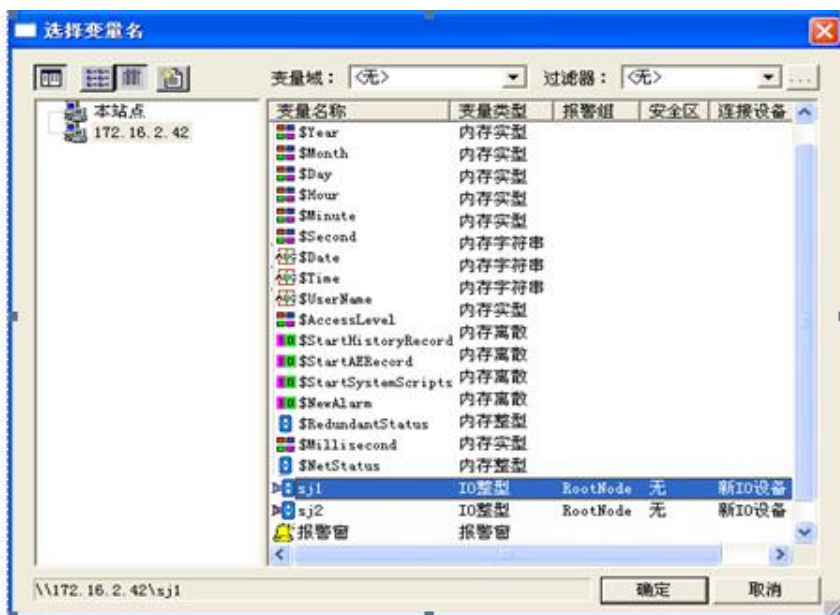

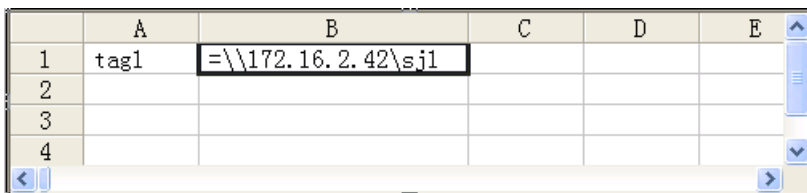


图 23.19 变量选择对话框

3、变量选择完之后，单击确定按钮，相应的变量就被添加到了报表工具箱的编辑框中。

本例中选择的是远程站点中的变量域 variable1.Name。单击报表工具箱中的按钮  将变量添加到报表控件中，如下图所示：



	A	B	C	D	E
1	tag1	=\\172.16.2.42\sj1			
2					
3					
4					

图 23.20 在报表控件中添加变量

如果需要添加其他变量，按照上述步骤重复添加即可。



注意：

Web 服务器中的组态王 7.0 工程配置完成后，把需要进行发布浏览的画面设定为主画面。

23.3 Web 全新版发布示例

23.3.1 Web 全新版发布准备工作

在进行 Web 发布之前，需要做如下准备工作：

◇ 第一步：

在 Web 服务器中，安装 IIS(即 Internet 信息服务)组件，安装过程如下(以 Windows XP 为例)：点击 Windows 开始菜单，选择“控制面板”，在打开的控制面板中选择“添加或删除程序”选项，在“添加或删除程序”界面的左侧工具栏中，选择“添加/删除 Windows 组件”，弹出对话框如图 23.21 所示：



图 23.21 安装 IIS 服务组件

在上图中选择“Internet 信息服务 (IIS)”选项，点击下一步安装配置 Windows 组件（此过程中需要 Windows 系统盘中的“I386”文件夹中的所有文件），根据提示进行安装即可。

◇ 第二步：


查找工程中是否存在用户自定义的或者 Windows 通用的控件，这些控件不是组态王 7.0 的通用控件，无法被自动添加到 Web 发布列表中，需要用户手动进行添加。

⚠ 注意： Web 发布列表存放在一个名为“KV0cxDependFileList.xml”的文件中，请参考 23.1.2 相关文件介绍查询该文件的作用。

在 Web 发布列表中添加控件的格式如下：

```
<File Name="文件名" Register="1 或者 0"/>
```


- File Name: 控件的文件名, 如果控件存放在 Kingview 的安装目录下, 直接添加控件名即可, 否则需要在文件名前面添加文件的相对路径。
- Register: 是否需要在 IE 浏览客户端对控件进行注册; 需要注册的控件 Register="1 ", 不需要注册的控件 Register=" 0"。

 例如: 需要添加一个名为 kvbmptjpg.dll 的控件, 需要在 IE 浏览客户端注册该控件。

如果控件存放在 Kingview 的安装目录下, 则将控件添加到发布列表中的格式为: <File Name="kvbmptjpg.dll" Register="1"/>; 如果控件存放的路径为 C:\Windows\System32, 那么将控件添加到发布列表中的格式为: <File Name="C:\Windows\System32\kvbmptjpg.dll" Register="1"/>

◇ 第三步:

配置 WEB 服务器所在局域网中的其他防火墙或路由器, 确保相应端口开放。为了保障组态王 7.0WEB 发布服务器在客户端可以正常浏览, 要求 WEB 发布服务器所在的网络环境中必须双向开放 80 端口(用户可修改)、2001 端口(固定、不可修改)、2002 端口(固定, 不可修改)。备注: 80 端口修改方法: 在控制面板中, 双击“管理工具\Internet 信息服务”, 弹出如下对话框:

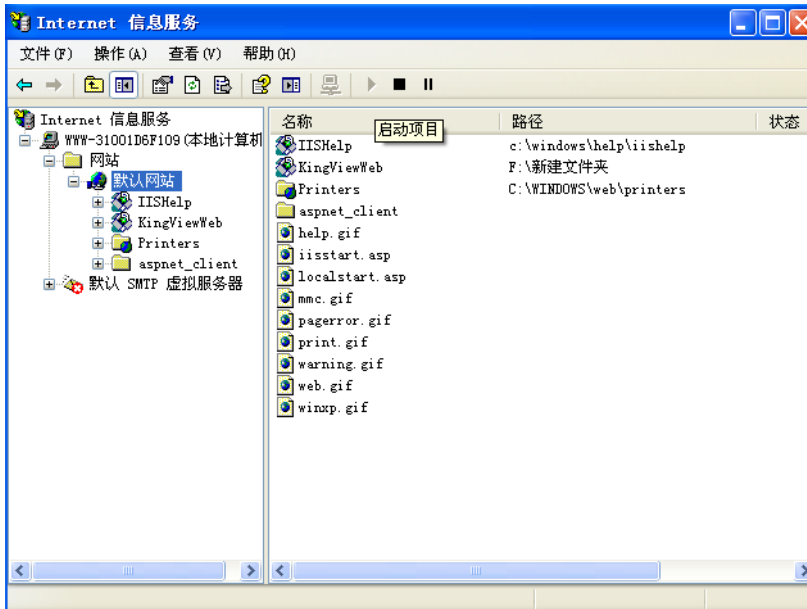


图 23.22 Internet 信息服务

用鼠标选中“默认网站”然后点击右键，在弹出的菜单中，选择“属性”菜单弹出如图 23.23 所示的对话框：

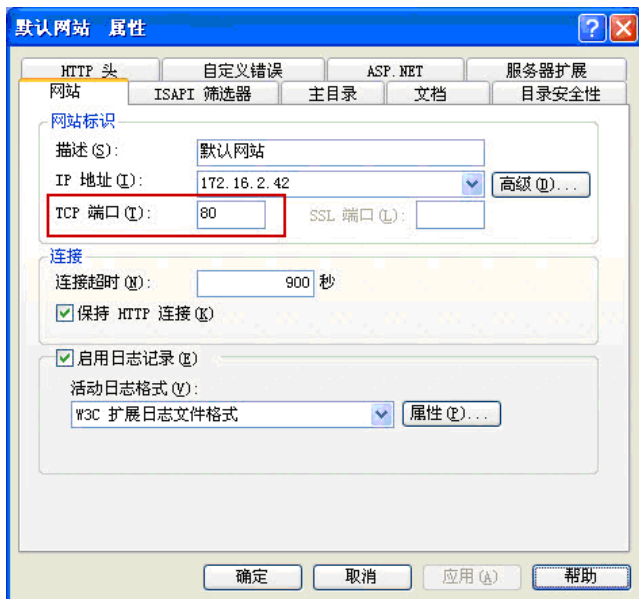


图 23.23 端口设置

当系统中 80 端口被其他应用程序占用时，用户可手动修改 80 端口为其他值。

23.3.2 Web 全新版发布步骤

准备工作完成之后，就可以进行 Web 发布了，Web 发布的主要功能就是实现组态王 7.0 工程的网络浏览，即实现 Internet/Intranet 访问。

双击 KVWebPublish.exe 应用程序，弹出对话框，如下图所示：

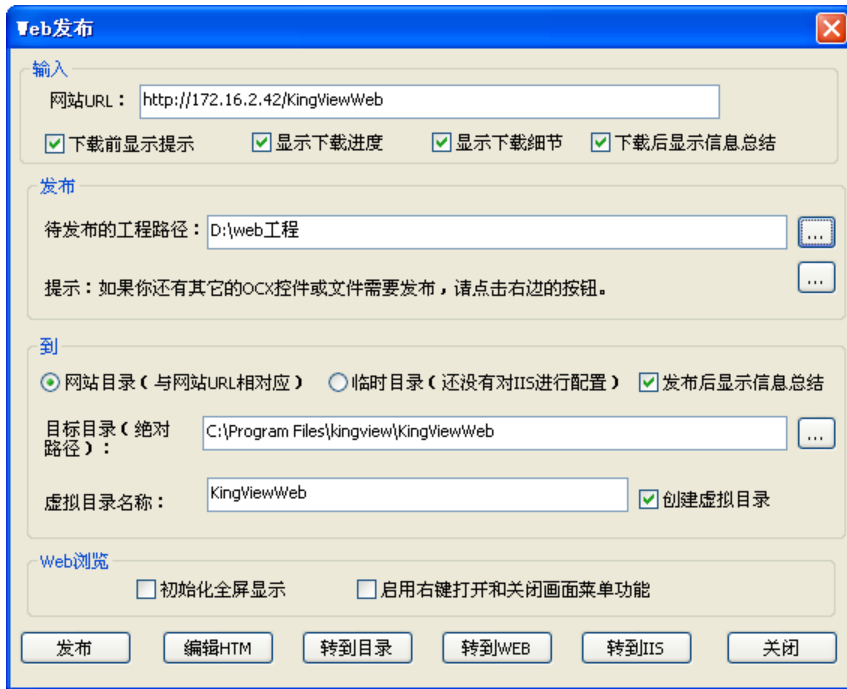



图 23.24 Web 发布对话框

◇ 网站 URL:

Web 发布后，提供给用户的客户端工程下载地址，远程用户在 IE 地址栏中输入该地址后，就可以实时访问浏览客户端工程。网站 URL 的定义格式为：
http://172.16.2.42/*****，其中 172.16.2.42 是 Web 客户端主网卡的 IP 地址，建议用户不要随意更改，*****是与虚拟目录名称相对应的网站标识符。例如：
http://172.16.2.42/KingViewWeb，其中 KingViewWeb 是虚拟目录的名称。

 注意:

发布不同工程时，必须使用不同的网站 URL。

◇ **下载前显示提示:**

选中该选项，在下载前弹出是否进行下载的提示，用户可以选中是否进行下载，否则，不弹出该提示直接进行文件下载。

◇ **显示下载进度:**

选中该选项，在 IE 浏览客户端下载组态王 7.0 工程时，弹出下载进度对话框，用户可以控制文件下载进程，否则，直接下载文件。


◇ **显示下载细节:**


使用该选项之前必须先选中“下载前显示提示”复选框；如果两个复选框同时被选中，在 IE 浏览客户端下载组态王 7.0 工程时，显示详细的下载过程，否则不显示任何下载进程。

◇ **下载后显示信息总结:**

选中该选项，在 IE 浏览客户端下载组态王 7.0 工程时，全部文件下载完成后，弹出下载信息总结对话框。

◇ **待发布的工程路径:**

Web 服务器中的组态王 7.0 工程所在的路径，可以在编辑框中直接输入，也可以点击编辑框后面的按钮 ，在弹出的对话框中选择 Web 服务器中的组态王 7.0 工程路径，选择完成后，单击确定按钮，选中的文件路径被自动添加到编辑框中。

 注意:

目前只支持本地路径工程的发布。


工程路径添加完之后, 如果还有其他的 OCX 控件或文件需要发布, 则点击提示信息后面的按钮 , 弹出“其它可能需要发布的文件”对话框, 如下图所示:




图 23.25 添加其他需要发布的 OCX 控件和文件

- **添加按钮:** 点击添加按钮, 弹出属性配置对话框如下图所示:



图 23.26 待发布文件属性配置对话框

在属性配置对话框中，对待发布的文件属性进行详细配置，具体步骤如下所示：

- **待发布的文件名：**在“待发布的文件名”后面的编辑框中直接输入待发布的文件所在的路径，或点击编辑框后面的按钮 ，在弹出的对话框中直接选择需要添加的文件后，单击确定按钮，选中的文件所在的路径被自动添加到编辑框中。
- **发布到默认路径：**该配置项不可更改。新添加的文件被发布到默认路径下，即组态王 7.0 工程发布的路径；
- **下载到默认路径：**该配置项不可更改。新添加的文件被下载到默认路径下，即组态王 7.0 工程下载的路径；
- **注册：**选择文件是否需要注册。选中则表示文件下载之后需要用户注册，不选则表示文件不需要用户注册。
- **启动运行：**选择是否自动运行文件。选中则表示下载文件之后，文件立刻自动运行，不选则文件不会自动运行。
- **仅发布不下载：**选择文件是否需要下载。选中则表示文件仅仅发布到虚拟目录，但是不需要下载到客户端，即文件不需要添加到下载列表中，不选则表示文件既需要进行发布又需要进行下载。

待发布的文件属性配置完成之后，单击确定按钮，属性配置对话框自动关闭，选中的文件被自动添加到可能需要发布的文件列表中。如下图所示：



图 23.27 添加待发布的文件

- **删除按钮：**删除选中的文件。选中需要删除的文件，单击删除按钮，则待发布的文件从列表中被删除。
- **修改按钮：**修改选中的文件。选中需要修改的文件，单击修改按钮，弹出属性配置对话框，在对话框中对文件属性进行修改。
- **上移/下移按钮：**改变文件的排列顺序。在文件列表中选择文件，单击上移/下移按钮，选中的文件在列表中上移/下移一行。配置完成后，单击确定按钮，文件列表中的文件被添加到 Web 发布列表中。

◇ **到网站目录（与网站 URL 相对应）：**

Web 服务器上配置 IIS 后，该选项有效；选中该选项之后，组态王 7.0 工程的发布信息文件被发布到目标目录和与网站对应的虚拟目录中。Web 发布成功后，远程用户能够在 IE 浏览客户端浏览工程。

◇ **到临时目录（还没有对 IIS 进行配置）：**

如果 Web 服务器上没有配置 IIS，则用户只能选择该选项。组态王 7.0 工程的发布信息文件只能被发布到目标目录中。Web 发布之前会弹出一个对话框，提示用户，Web 发布产生的信息文件只能存放在目标目录中，与网站 URL 对应的虚拟目录不存在，发布完成后，网站 URL 仍然无效，远程用户不能在 IE 浏览客户端上浏览工程。

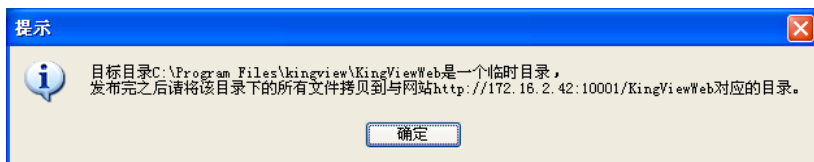



图 23.28 提示信息

◇ 目标目录：

Web 发布后，存储 Web 服务器中组态王 7.0 工程的发布信息文件的目录。Web 发布的过程中，要生成一些 HTML 格式的文件，还要对一些相关文件进行格式转换，新生成的 HTML 格式文件以及进行格式转换后的文件都在目标目录下存放。在后面的编辑框中直接输入目标目录，或点击编辑框后面的按钮 ，在弹出的对话框中直接选择目标目录，单击确定按钮，选中目标目录被自动添加到编辑框中。

◇ 虚拟目录名称/创建虚拟目录：

“虚拟目录名称”需要与“创建虚拟目录”复选项搭配使用；选中创建虚拟目录之后虚拟目录名称选项被激活，否则虚拟目录名称无效。用户可以通过以下两种方式创建虚拟目录：

- **由系统自动建立虚拟目录：**在编辑框中输入任意虚拟目录的名称后，选中“创建虚拟目录”复选框，Web 发布之后，系统会在 IIS 中自动建立一个虚拟目录。

● 手动建立虚拟目录：IIS 虚拟目录的创建过程如下所示：

- 1) 点击桌面“我的电脑”图标，弹出右键菜单，选择“管理”，弹出如图所示“计算机管理”对话框。
- 2) 选择“服务和应用程序”下的“Internet 信息服务”中的“网站”选项下的“默认网站”，点击“默认网站”，在弹出的右键菜单中选择“新建”→“虚拟目录”选项，如图 23.29 所示：



图 23.29 新建虚拟目录

- 3) 执行上述命令弹出虚拟目录创建向导对话框，如图 23.30 所示：



图 23.30 虚拟目录创建向导对话框

- 4) 点击下一步，输入虚拟目录别名（如：webpublish），如图 23.31 所示：

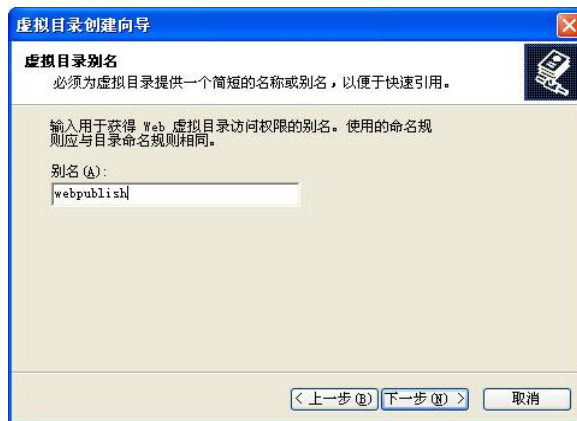


图 23.31 输入虚拟目录名称

- 5) 点击下一步，选择 Web 发布后存放信息文件的文件夹目录，即 Web 发布对话框中的目标目录，如图 23.32 所示：



图 23.32 关联 Web 发布的文件夹目录

6) 依次点击下一步，直至完成配置即可。

◇ **发布后显示信息总结：**

选择是否显示发布信息总结对话框。选中该选项，Web 发布完成后显示发布信息总结对话框，不选则不显示任何信息。

◇ **初始化全屏显示：**

选中该选项，以全屏方式进行 web 浏览。

◇ **启用右键打开和关闭画面菜单功能：**

选中该选项，在浏览器端右键菜单包含打开和关闭画面的菜单项，用户可以使用右键菜单对画面进行打开和关闭操作。否则，右键菜单对画面的打开和关闭功能不可用。

◇ **发布：**

点击发布按钮，开始对 Web 服务器中的组态王 7.0 工程进行发布。Web 发布对话框中的各项属性配置完成，并确认无误后，就可以对工程进行发布，点击发布按钮，弹出如下所示的对话框：

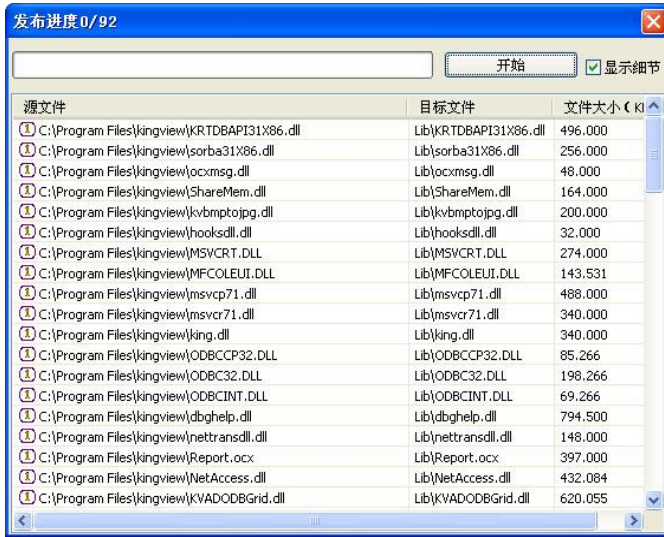


图 23.33 Web 发布进度对话框

列表中显示所有待发布的文件清单，待发布的文件由三部分组成：目标 OCX 及其依赖的 DLL 控件，用户 OCX 控件和组态王 7.0 工程文件。

- **源文件：**待发布的源文件路径。
- **目标文件：**Web 发布后的目标文件路径。
- **文件大小：**待发布文件的大小（单位 KB）。

- **显示细节：**选择是否显示待发布文件的详细信息。选中该选项，发布进度对话框如上图所示，不选则不显示待发布文件的详细信息，只显示发布进度条，如下图所示：

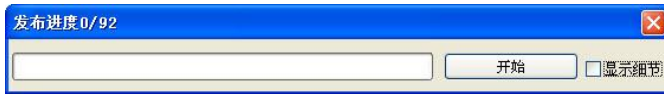


图 23.34 不显示详细信息的发布进度对话框

单击开始按钮，开始发布 Web 服务器中的组态王 7.0 工程，开始按钮被替换为另外两个按钮：暂停和取消，单击暂停按钮，Web 发布被暂停，单击取消按钮，取消 Web 发布，发布进度对话框被关闭。如果“发布后显示信息总结”复选项被选中，则 Web 发布成功后弹出信息总结对话框，如下图所示：

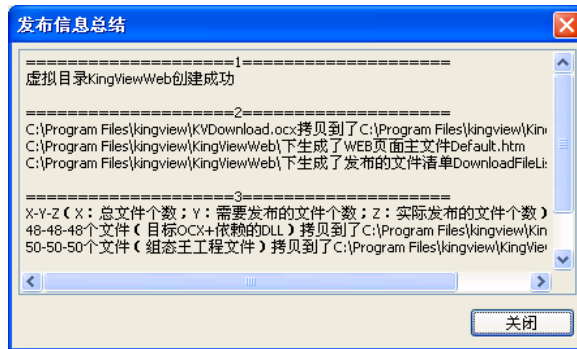


图 23.35 发布信息总结对话框

发布信息总结对话框中显示的信息包括：

- 虚拟目录是否创建成功；
- 目标目录下拷贝和生成的文件信息；

- 目标目录下的 Lib、Project 和 CustomFile 中分别需要拷贝的总文件个数、需要发布的文件个数和实际发布的文件个数。

◇ **编辑 HTML:**

单击该按钮，打开 Default.html 文本文件，该文件中记录了 Web 发布所生成的网站主页面的相关信息。

◇ **转到目录:**

单击该按钮，打开 Windows 资源管理器，并转到目标目录下。

◇ **转到 WEB:**

单击该按钮，启动 IE 浏览器，自动转到用户指定的网站 URL，并打开网站主页面文件 Default.htm。

◇ **转到 IIS:**

单击该按钮，转到 Internet 信息服务对话框，查看 Web 发布时所创建的虚拟目录下的文件。

◇ **关闭:**

单击该按钮，关闭 Web 发布对话框。



注意:

如果网络中配置了登录服务器，则要求登录服务器的用户列表必须和客户端的用户列表保持一致，否则会有各种问题；如果登录服务器的用户列表修改，必须手动刷新发布工程中远程站点的信息，再发布。也就是说，服务器端的工程进行修改

后（除了画面的修改外），一般都要求手动更新发布工程中远程站点的信息，然后再发布。


23.4 如何在 IE 浏览客户端浏览工程

Web 发布成功后，远程用户能够在 IE 浏览客户端的 IE 中浏览 Web 服务器中发布的组态王 7.0 工程，包括：实时数据、历史数据、报警数据以及用户权限等。

23.4.1 浏览前的准备工作

组态王 7.0 的 WEB 发布支持通过 IE、360、世界之窗等常见的浏览器进行浏览。下面以 IE 浏览器为例，说明浏览前的必要准备工作。

第一步：安装 IE 浏览器并进行必要的安全设置

 注意：

如果浏览器的安全级别是默认的安全级别且级别不为“高”，都可以正常下载发布内容，无需进行下文所述的浏览器的安全设置。

- 1、首先安装 IE6.0 或以上版本浏览器。
- 2、对 IE 浏览器进行安全设置。

IE 浏览器的安全设置可以由两种方法来原因：

- 1) 添加信任站点并对其进行安全设置；

添加信任站点并对其进行安全设置：双击系统控制面板上的 Internet 选项或者直接在 IE 浏览器中选择“工具/Internet 选项”菜单，打开“安全”属性页，选择“受信任的站点”图标，如下图所示：



图 23.36 Internet 选项

单击图 23.36 右下方的“站点”按钮，在打开的“可信站点”对话框中输入 Web 客户端主网卡的 IP 地址，单击“添加”按钮，即可将该站点添加到信任域中，如图 23.37 所示。

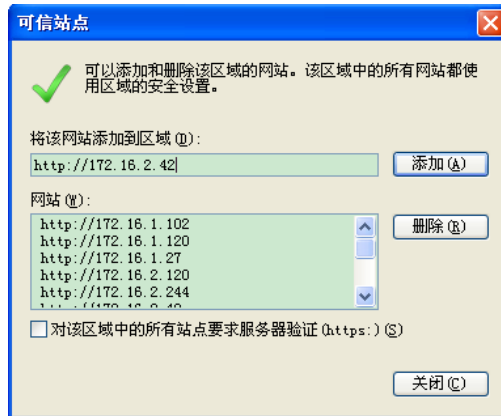


图 23.37 添加可信站点

单击“确定”按钮返回“Internet 选项”对话框。



图 23.38 受信任站点的设置

在图 23.38 中，选中受信任的站点的图标后，点击下方的“自定义级别”按钮，在弹出的安全设置对话框中，对 Active 控件和插件做如下设置：



图 23.39 受信任站点设置一



图 23.40 受信任站点设置二

单击确定按钮，完成信任站点的安全设置。**值得一提的是：**组态王 7.0 提供了一个自动完成信任站点安全设置的工具- 浏览器配置工具(保存在组态王 7.0 的安装目录下)，即在完成信任站点的添加后，通过双击运行该工具实现信任站点的安全设置

2) 直接对 IE 进行安全设置。

方法一：

直接对 IE 进行安全设置。双击系统控制面板上的 Internet 选项或者直接 在 IE 浏览器中选择“工具/Internet 选项”菜单，打开“安全”属性页，如图 23.41 所示。



图 23.41 Internet 选项

方法二：

选中 Internet 图标，然后单击下方的“自定义级别”按钮，在弹出的“安全设置”对话框中，按照图 23.39 和 23.40 所示对 Active 控件和插件做相应的设置。

注：对于上面两种 IE 安全设置的方法，我们推荐用户使用第一种方法。

第二步：配置防火墙

配置 WINDOWS 自带防火墙(如果系统防火墙被关闭,则不需要此步配置)。下面以 Windows XP 为例。

WINDOWS 防火墙是基于“例外”的,也就是默认情况下,防火墙将阻止外部“未被请求”的连接通过网络,而管理员可以在规则之外设置特定的应用程序或端口来响应外部“未被请求”的连接。

防火墙的例外可被归入两种层次的情况,一是应用程序层次,二是端口与协议层次。前者可设置特定的程序来对“未被请求”的连接进行响应,后者可设置特定的 TCP 或 UDP 端口来允许相应的通信。为了使远程用户可以浏览 Web 客户端工程,只需要在进行浏览的主机上设置相应的端口来允许相应的通信即可。

防火墙的配置过程如下:

- 1、为了给系统提供必须的保护,WINDOWS 防火墙是默认启用的如图 23.40 所示。(个人)不推荐关闭 WINDOWS 防火墙,若通信连接失败,在调试过程中可以暂时关闭防火墙以确认问题是否是由防火墙所引起。如若确定永久关闭防火墙,下面所述关于防火墙的设置均可忽略。



图 23.42 WINDOWS 防火墙

2、添加 TCP 端口。需要添加的 TCP 端口有两个：80 端口用于开放 IE 浏览器的下载功能，和 2001 端口用于开放控件与组态王 7.0 工程之间的通讯。添加端口的步骤如下所示：在“WINDOWS 防火墙”的“例外”选项卡中，点击“添加端口”按钮。在“添加端口”对话框中进行设置，如图 23.43 所示：



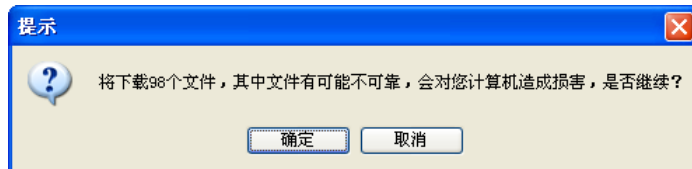
图 23.43 添加端口

接下来您就可以使用 IE 浏览器浏览 Web 客户端的工程了。

23.4.2 浏览工程

使用 IE 浏览器浏览工程时，在 IE 地址栏中输入 Web 发布过程中设置好的网站 URL，例如，<http://172.16.2.42/KingViewWeb/>。

第一次进行 Web 浏览时会弹出如下所示的对话框，提示用户将要下载文件到本地计算机：



文件下载提示对话框

单击“确定”按钮，弹出如下所示的对话框：

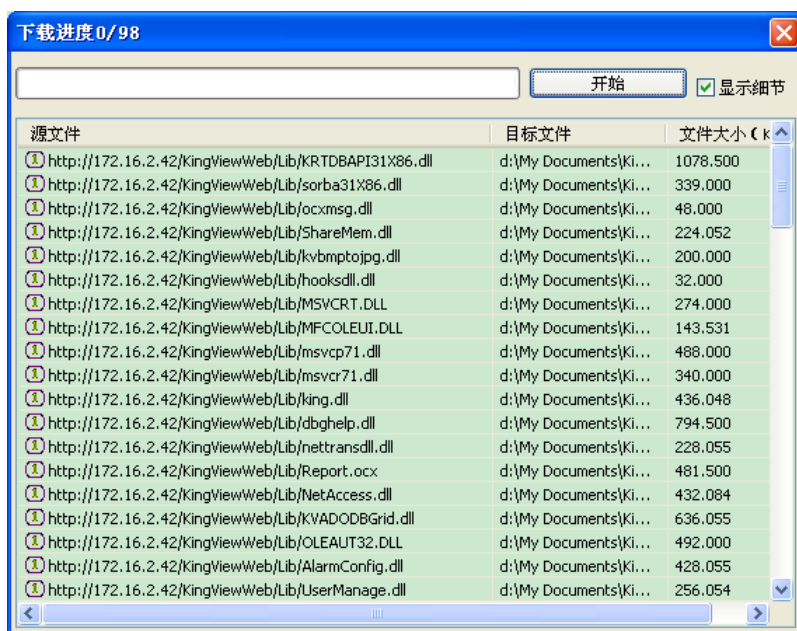


图 23.44 下载进度对话框

图 23.44 所示的对话框中，列出了将要下载到本地的文件清单，单击开始按钮，开始对文件进行下载，下载完成后，弹出下载信息总结对话框，如下图所示：

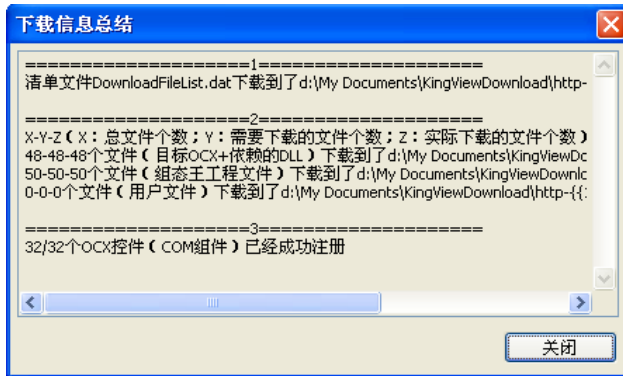



图 23.45 下载信息总结对话框

 注意:

下载进度对话框和下载信息总结对话框与发布进度对话框和发布信息总结对话框类似，在此不再赘述。

文件下载完成之后，就可以对 Web 服务器中的组态王 7.0 工程进行浏览了，浏览界面如下图所示：

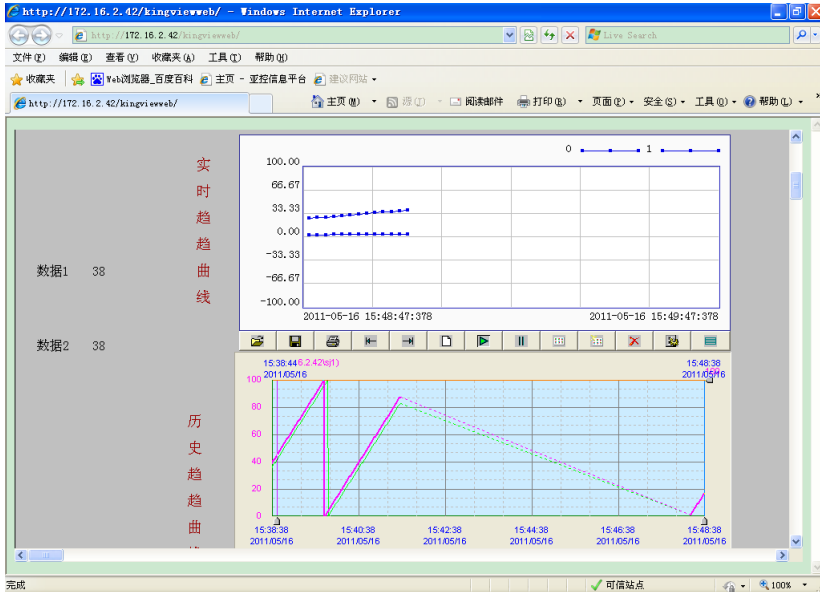
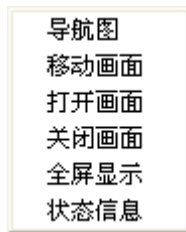


图 23.46 工程浏览页面

在上面的工程浏览页面，单击鼠标右键将会弹出快捷菜单，如下图所示。



- ◇ **导航图**：单击后，在画面的右上方会出现矩形显示小窗口，该窗口就是画面的导航图。
- ◇ **移动画面**：该选项用于画面的移动。
- ◇ **打开画面**：选择此命令后，弹出“打开画面”对话框。该对话框列出了当前

路径下所有画面的清单。用鼠标选择一个或多个窗口，单击“确定”按钮打开所选中的画面，或单击“取消”按钮撤消当前操作。（发布时如未选中“启用右键打开和关闭画面功能”选项，该右键菜单项不可用）

- ✧ **关闭画面：**关闭已打开的画面。（发布时如未选中“启用右键打开和关闭画面功能”选项，该右键菜单项不可用）
- ✧ **全屏显示：**全屏显示画面。
- ✧ **状态信息：**弹出“信息窗口”，显示画面的状态信息。



注意：

在 Win7 系统中下载和浏览工程时，可能会出现控件注册失败的错误。这时需要对用户的权限进行设置，选择“控制面板 -> 用户账户 -> 更改用户账户控制设置”。把设置更改为最低等级的从不通知。重启计算机，再使用 IE 浏览器下载发布的工程就不会出现注册失败的问题了。

23.5 数据库数据的发布

和以前的组态王 7.0 发布功能一样，Web 全新版也支持关系型数据库数据的 WEB 发布，即通过 ODBC 方式从关系型数据库获取数据。不一样的是，对于 Web 全新版的数据库数据的访问和发布，需要配置远程的 ODBC。该节将以 SQLServer 为例详细讲述 ODBC 的远程配置和 Web 全新版的数据库数据的发布。

第一步：在 web 上创建 ODBC 数据源

在 web 服务器上创建 ODBC 数据源。如果 web 服务器和数据服务器在同一台电脑上且

数据服务器已经创建过数据源，此步骤可以省略。

进入“控制面板”中的“管理工具”，用鼠标双击“数据源（ODBC）”选项，弹出“ODBC 数据源管理器”对话框，如图 23.47 所示。



图 23.47 ODBC 数据源管理器

选中“系统 DSN”标签页，单击右上方的“添加”按钮。弹出“创建新数据源”对话框，如图 23.48 所示。



图 23.48 创建新数据源

选中 SQL Server 选项，单击“完成”按钮弹出如下图所示的对话框。

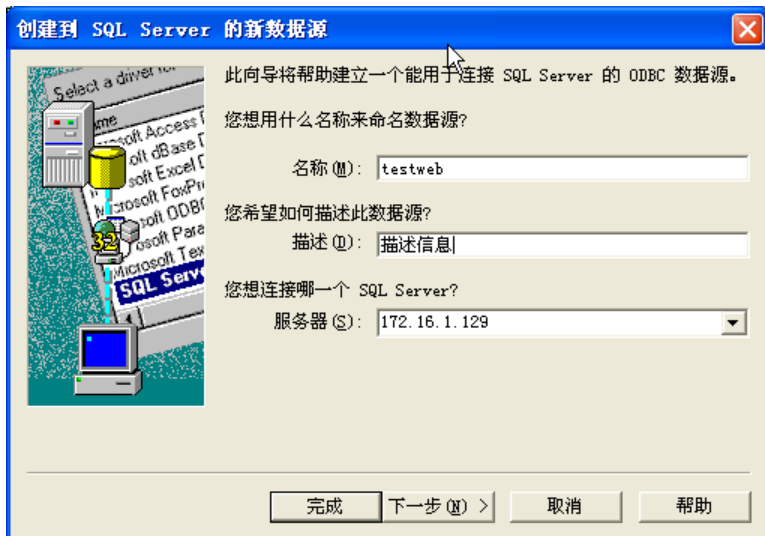


图 23.49 创建 SQL Server 数据源

输入要创建的数据源名称和描述信息，选择 SQL Server 所在的服务器名称或者 IP 地址。单击“下一步”按钮，弹出如图 23.50 所示画面。

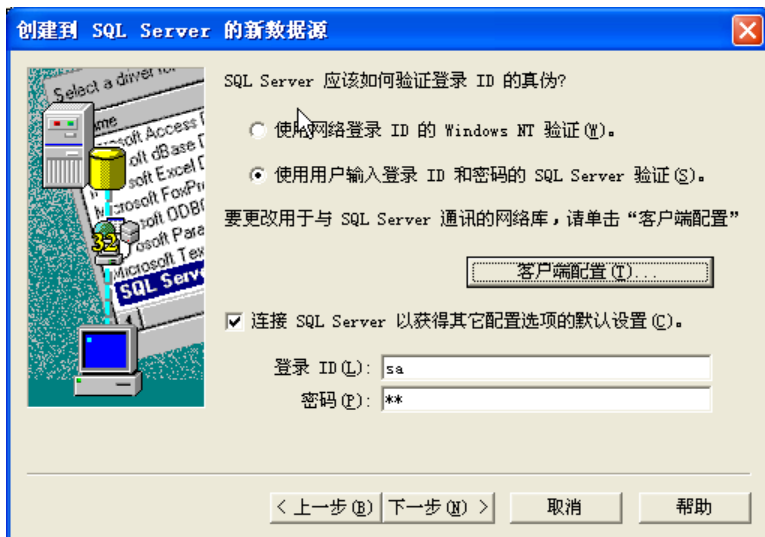


图 23.50 配置登录

选择登录 SQL Server 时的身份验证方式。并输入登录 SQL Server 时所用到的用户名和密码（这里的用户名和密码是在 SQL Server 中建立的）。单击“下一步”按钮，弹出图 23.51 所示的对话框。

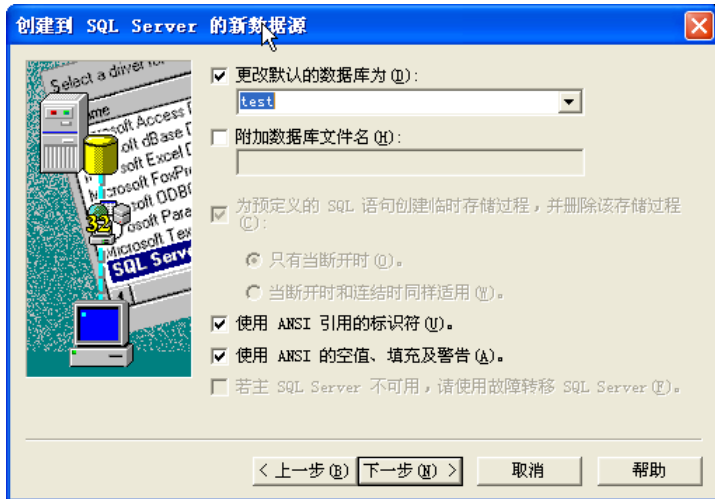


图 23.51 选择数据库

选择“更改默认的数据库为”选项，然后选择您想要连接的数据库。其余的按默认设置。然后单击“下一步”按钮。如图 23.52 所示。

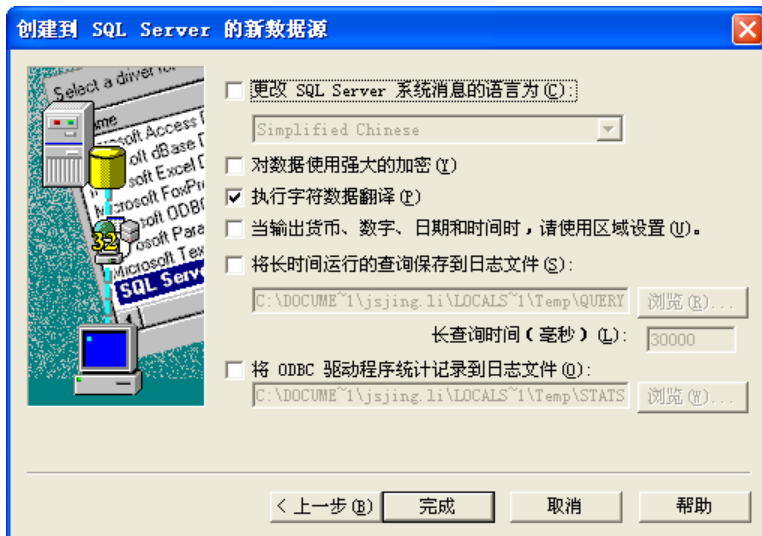


图 23.52

按照默认设置, 单击“完成”按钮, 弹出图 23.53 所示的对话框中。



图 23.53

这里给出了数据库连接的摘要信息, 如果想测试一下是否能够连接到数据库, 单击“测试数据源”按钮, 会出现测试结果, 如图 23.54 所示。



图 23.54 测试结果

单击“确定”按钮完成配置。至此，web 服务器的 ODBC 数据源的配置已完成。

第二步：创建 KVADODBGrid 控件并连接数据源

在 Web 服务器的组态王 7.0 工程画面中，新建一个 KVADODBGrid 控件（详情请参考第十四章的 14.3.4.1），并将该控件连接到上面配置 SQL Server 数据源 testweb 上，连接过程如下所示。

1. 右键单击新建的 KVADODBGrid 控件，在弹出的快捷菜单中选择“控件属性”选项，如图 23.55 所示。



图 23.55 KVADODBGrid 控件

弹出的控件属性对话框如图 23.56 所示。



图 23.56 KVADODBGrid 控件属性

2. 单击数据源标签页右上角的“浏览”按钮，会弹出“数据链接属性”对话框，在列表中显示了各种数据连接引擎，这里请选择 Microsoft OLE DB Provider for ODBC Drivers。点击“下一步”按钮，进入“连接”属性页。如下图 23.57 所示。

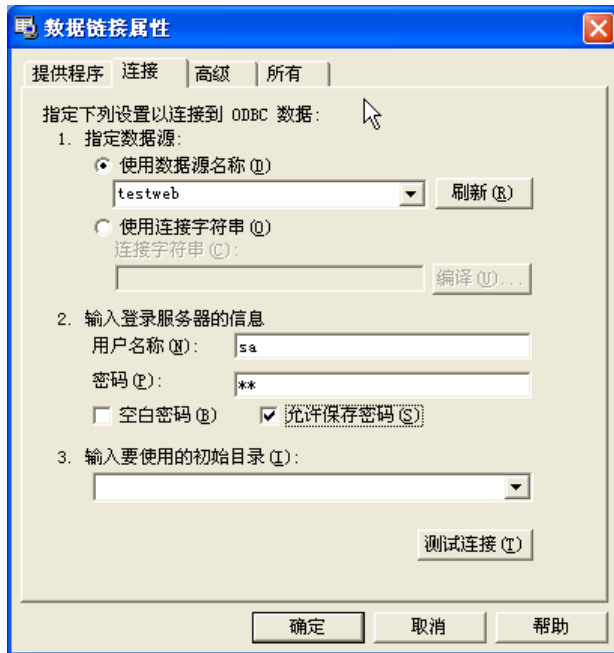


图 23.57 远程数据库的连接

在“使用数据源名称”编辑框中输入数据源名称 testweb，或单击后面的按钮在弹出的下拉列表中选择已定义的 SQL Server 数据源 testweb。然后配置登陆的用户名和密码，并选中允许保存密码。与数据源的连接测试成功后，点击“确定”按钮返回“数据源”属性页。如图 23.58 所示，选择控件连接的数据表及要显示的字段。



图 23.58 远程数据表的连接

3. 数据源连接完成后，单击“确定”按钮返回组态王 7.0 开发系统，选定的字段将会在控件标题栏中显示。然后在界面上新建两个按钮，查询数据和获得数据，并分别将它们的“动画连接\命令语言连接\弹起时”设置为 `Ctrl1.QueryDialog()` 和 `Ctrl1.FetchData()`，其中 `Ctrl1` 为控件名。如图 23.59 所示。

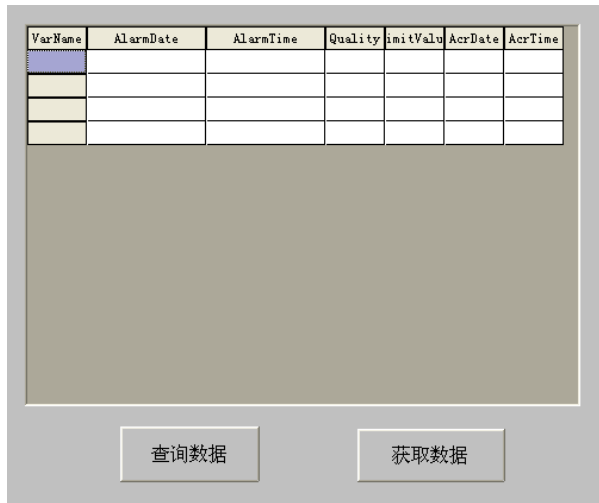


图 23.59

第三步：发布

完成数据库与 KVAD0DBGrid 的连接之后，按照该章的第 2、3、4 节完成 Web 全新版工程发布前的配置和准备工作。此外，在浏览器端电脑上需要配置跟发布服务器上同名的数据源，可以按照上述“第一步”中方法配置，也可以在发布工程中使用组态王 7.0 的函数进行自动配置。做完上面所有的步骤后，就可以在浏览器端浏览数据库的数据了，浏览界面如图 23.60 所示：

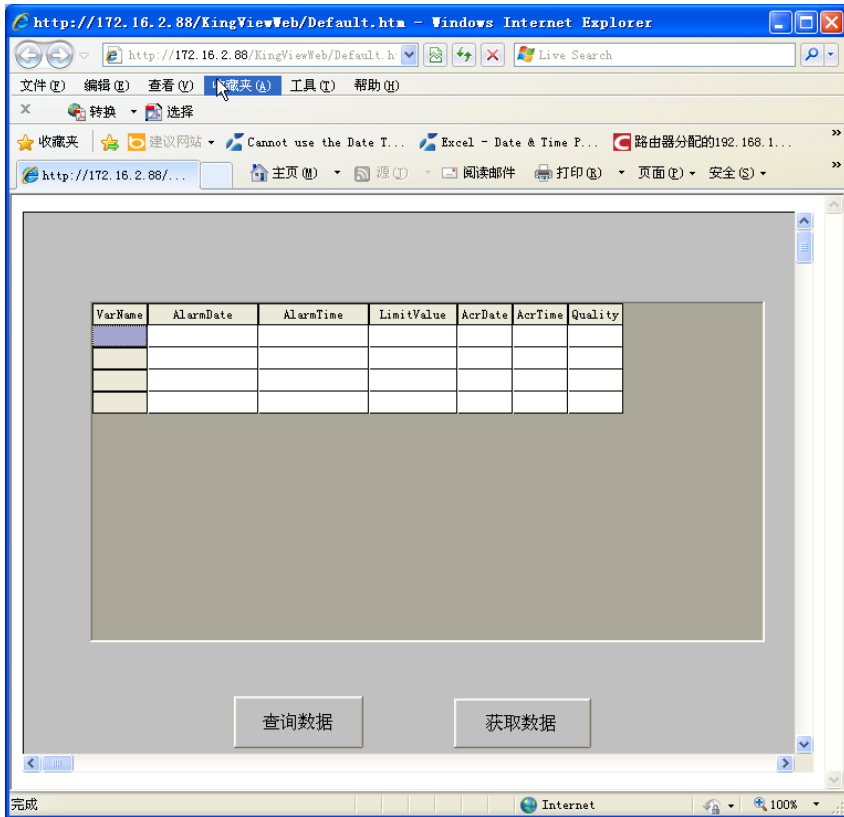


图 23.60 IE 浏览画面

单击“查询数据”按钮，弹出如图 23.61 所示的查询条件对话框。

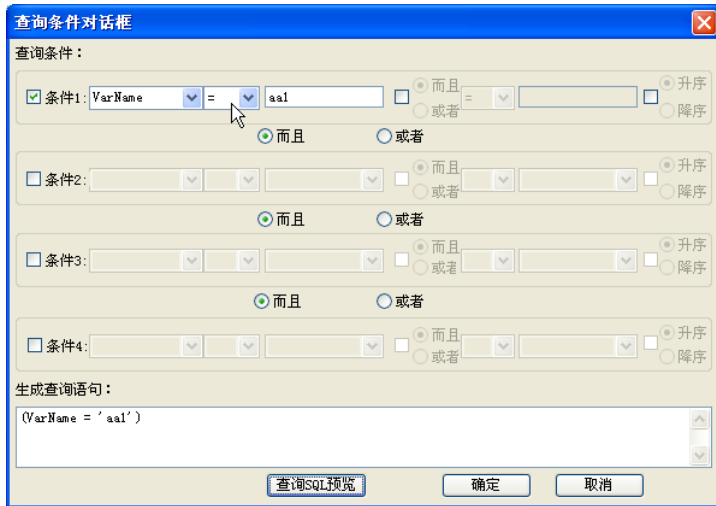


图 23.61 查询条件对话框

配置查询条件，并点击“查询 SQL 预览”按钮自动生成查询语句中的查询条件（详情请参考第十四章的 14.3.4.5）。配置好查询条件后，单击“确定”按钮返回图 23.60 所示的界面。单击“获得数据”按钮将查询到的数据集填充到控件中，如图 23.62 所示。

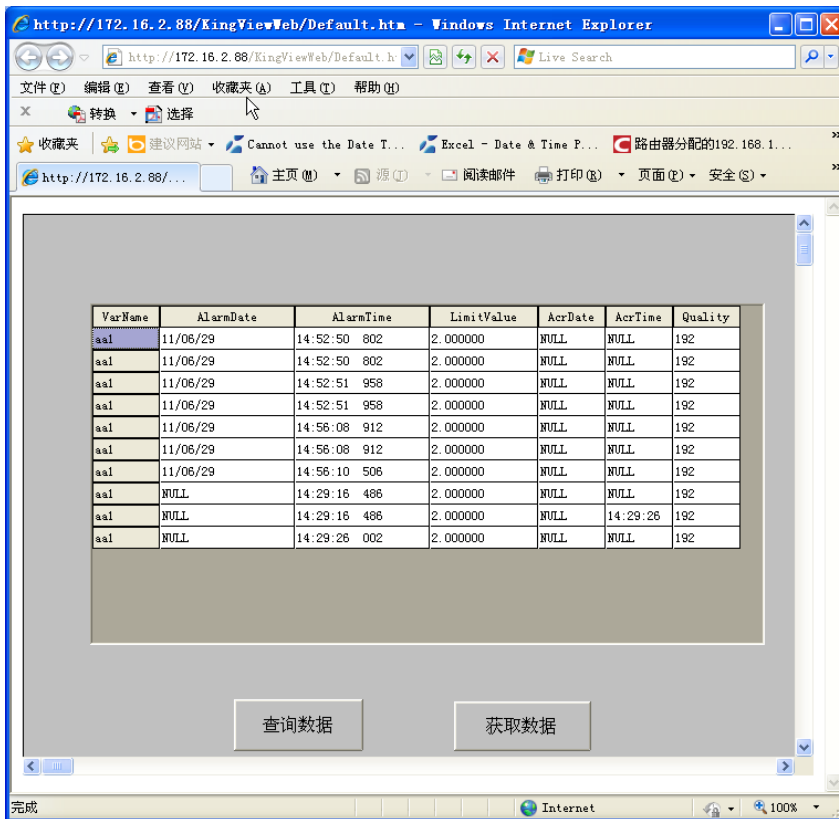



图 23.62 查询到的数据

到这里，Web 全新版的数据库数据的发布和访问已经完成，其他数据库数据的发布和访问与 Access 的类同，这里不再详述。

23.6 Web 全新版支持与不支持的功能

23.6.1 Web 全新版不支持的功能

- ◇ I/O 设备，待发布的工程中不能出现 I/O 设备，包括 OPC、DDE、虚拟设备以及各种组态王 7.0 驱动支持的板卡、仪表、PLC 等，即 Web 服务器中的工程不具备任何数据采集功能。
- ◇ 网络站点
- ◇ 配方
- ◇ 温控曲线的下载功能，因为下载功能需要设备驱动
- ◇ 与加密锁相关的所有功能
- ◇ Web 服务器中的组态王 7.0 工程不能作为 I/O 服务器、历史服务器、报警服务器、登录服务器或者校时服务器
- ◇ 不具备历史存储、报警存储及任何其他需要数据记录的功能
- ◇ 组态王 7.0 中的 Web 功能，包括发布画面、发布实时信息、发布历史信息以及发布数据库信息等功能
- ◇ 运行态的菜单，例如：文件、特殊、调试、导航、关于等
- ◇ 与 I/O 设备有关的任何内部控件，例如视频监控控件，因为这些控件不仅需要通过驱动和设备交互，还需要配置一些资源，因此在 Web 服务器中的组态王 7.0 工程中最好不要使用

 注意:


Web 全新版不支持组态王 7.0 的上述功能, 在 Web 服务器的组态王 7.0 工程中最好不要使用。

23.6.2 Web 全新版不支持的函数

- ◇ AdjustClock
- ◇ EnableDisableKeys
- ◇ EnableAllAlarm
- ◇ Exit
- ◇ GetBackupProgress
- ◇ GetStationStatus
- ◇ GetHistMinTime
- ◇ GetHistMaxTime
- ◇ GetHistAveData
- ◇ GetHistMinData
- ◇ GetHistMaxData
- ◇ GetKey
- ◇ GetRDBData
- ◇ GetRDBStatisData
- ◇ HistoryDBServerRun

- ◇ LoadDriverConfig
- ◇ LogOnEx
- ◇ ReportSetTimeEx
- ◇ ReportWebDownload
- ◇ ReadTag
- ◇ Report2
- ◇ Report1
- ◇ ReBuildUnConnectDDE
- ◇ ReBuildDDE
- ◇ SampleVarEnd
- ◇ SampleVar
- ◇ SetIoDeviceRunState
- ◇ SetPrintAlarm
- ◇ SavePicToFile 只支持保存为 BMP，去掉了转换为 JPG 的过程。

与历史库读取等操作有关的函数如 GetHistData、GetStatisData 等，只能从远程服务器上读取历史数据。

 注意：

上述函数不能在 Web 服务器的组态王 7.0 工程中使用。

23.6.3 Web 全新版支持的功能和函数

除了前面两节中列出的功能和函数之外，Web 全新版支持组态王 7.0 中其他所有的功能和函数。

第二十四章 组态王历史库

- ☞ 介绍组态王历史库的功能
- ☞ 介绍组态王网络版工程中历史库需进行的配置
- ☞ 介绍历史数据的浏览

数据存储功能对于任何一个工业系统来说都是至关重要的,随着工业自动化程度的普及和提高,工业现场对重要数据的存储和访问的要求也越来越高。一般组态软件都存在对大批量数据的存储速度慢、数据容易丢失、存储时间短、存储占用空间大、访问速度慢等不足之处,对于大规模的、高要求的系统来说,解决历史数据的存储和访问是一个刻不容缓的问题。组态王顺应这种发展趋势,提供高速历史数据库,支持毫秒级高速历史数据的存储和查询。采用最新数据压缩和搜索技术,数据库压缩比低于 20%,大大节省了磁盘空间。查询速度大大提高,一个月内的数据按照每小时间隔查询,可以在百毫秒内迅速完成。完整实现历史库数据的后期插入、合并。可以将特殊设备中存储的历史数据片段通过组态王驱动程序完整的插入到历史库中;也可以将远程站点上的组态王历史数据片段合并到历史数据记录服务器上,真正的解决了数据丢失的问题。更重要的是,从组态王 6.55 系列开始扩展了数据存储功能,允许同时向组态王的历史库和工业库 KingHistorian 中存储数据。

 注意：

由于组态王 6.5 以后的版本采用了新的历史数据记录模式，组态王软件 6.5 以前版本所存储的历史库将与新的历史库不兼容，您所遇到的问题是查询不到以前的历史数据，请在采用新版本或升级时慎重考虑！

24.1 组态王变量的历史记录属性

在组态王中，离散型、整型和实型变量支持历史记录，字符串型变量不支持历史记录。组态王的历史记录形式可以分为数据变化记录、定时记录（最小单位为 1 分钟）和备份记录。记录形式的定义通过变量属性对话框中提供的选项完成。

在工程浏览器的数据词典中找到需要定义记录的变量，双击该变量进入“定义变量”对话框，选择“记录和安全区”属性页，如图 24.1 所示。

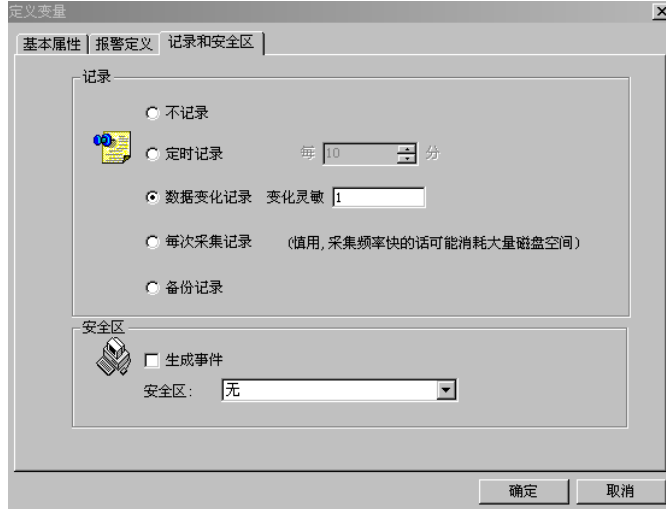


图 24.1 记录属性设置

记录属性的定义：

- ✧ **不记录：**此选项有效时，则该变量值不进行历史记录。
- ✧ **定时记录：**无论变量变化与否，系统运行时按定义的时间间隔将变量的值记录到历史库中，每隔设定的时间对变量的值进行一次记录。最小定义时间间隔单位为 1 分钟，这种方式适用于数据变化缓慢的场合。
- ✧ **数据变化记录：**系统运行时，变量的值发生变化，而且当前变量值与上次的值之间的差值大于设置的变化灵敏度时，该变量的值才会被记录到历史记录中。这种记录方式适合于数据变化较快的场合。
- ✧ **变化灵敏度：**定义变量变化记录时的阈值。当“数据变化记录”选项有效时，“变化灵敏度”选项才有效。



例如：

数据库中有一个实型变量，如果需要对该变量的值进行记录，而且规定其变化灵敏度为 1，则其记录过程如下：如果第一次记录值是 10，当第二次的变量值为 10.9 时，由于 $10.9-10=0.9<1$ ，也就是第二次变量值相对第一次记录值的变化小于设定的“变化灵敏度”，所以第二次变量值不进行记录，当第三次变量值为 12 时，由于 $12-10.9=1.1>1$ ，即变化幅度大于设定的“变化灵敏度”，所以此次变量值才被记录到历史记录中。

- ◇ **每次采集记录：**系统运行时，按照变量的采集频率进行数据记录，每到一次采集频率，记录一次数据。该功能只适用于 IO 变量，内存变量没有该记录方式。该功能应慎用，因为当数据量比较大，且采集频率比较快时，使用“每次采集记录”，存储的历史数据文件会消耗很多的磁盘空间。
- ◇ **备份记录：**选择该项，系统在平常运行时，不再直接向历史库中记录该变量的数值，而是通过其它程序调用组态王历史数据库接口，向组态王的历史记录文件中插入数据。在进行历史记录查询等时，可以查询到这些插入的数据。这种方式一般用于环境复杂的、无人职守数据采集点等场合。在这些场合使用的有些设备带有一定数量的数据存储器，可以存储一段时间的设备采集到的数据。但这些设备往往只是简单的记录数据，而不能进行历史数据的查询、浏览等操作，而且必须通过上位机的处理才可以看到。在组态王 6.55 以后的历史库中直接提供了这些处理的功能。



例如：

如图 24.2 所示。远程有若干具有历史记录功能的数据采集设备，中心控制室通过拨号网络与这些站点循环连接。因为是与每个站点间断连接的，所以如果在中心站上直接记录数据的话，会造成历史记录为间断的现象。而设备中存储的记录如果此时直接插入到历史库中，也会造成历史库混乱。在组态王 6.55 之后版本中完整的解决了这个问题。首先对应的变量的历史记录定义为“备份记录”，则无论系统是否与数据采集设备相连接，变量都不会向历史库中记录数据。当系统与某个设备连通后，系统通过驱动程序将设备中存储的历史记录读取上来，并按照约定的时间格式和变量类型等插入到组态王的历史库中。这样就保证了历史库的完整性。

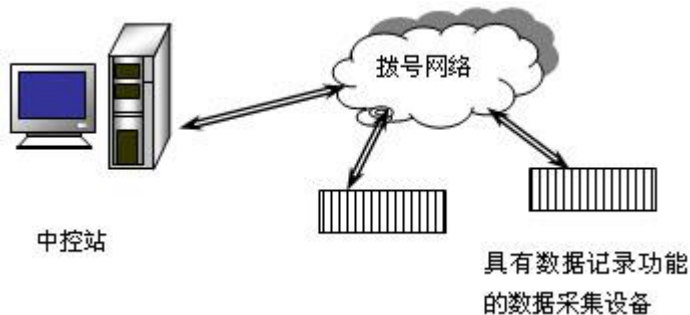


图 24.2 采集远程设备中的历史记录

24.2 历史记录存储及文件的格式

组态王以前的版本能够存储历史数据到组态王的历史库或 KingHistorian 工业库。6.55 及 7.0 版本进一步扩展了数据存储功能，即同时存储历史数据到组态王的历史库和工业库 KingHistorian 中。在组态王工程浏览器中，打开“历史库配置”属性对话框。如图 24.3 所示。

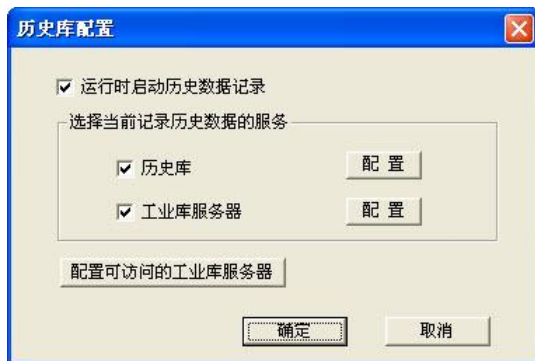


图 24.3 历史记录配置

 注意:

如果用户没有购买亚控公司的 KingHistorian 工业库服务器，不需要配置“工业库服务器”和“配置可访问的工业库服务器”两个选项，只配置“组态王历史库”就可以了。

- ◇ **运行时启动历史数据记录:** 如果选择“运行时启动…”选项，则运行系统启动时，直接启动历史记录。否则，运行时用户也可以通过系统变量“\$启动历史记录”来随时启动历史记录。或通过选择运行系统中“特殊”菜单下的“启动历史记录”命令来启动历史记录。
- ◇ **配置可访问的工业库服务器:** 当需要查询工业库服务器里的历史数据时，需要事先配置好该项。点击“配置可访问的工业库服务器”按钮，弹出“工业库配置”对话框。

- 配置工业库服务器的 IP 地址、端口号、登陆工业库的用户名、密码等选项，然后点击“添加”按钮，在列表中列出工业库服务器的信息，如图 24.4 所示。



图 24.4 工业库配置

- 选中某一工业库，在“工业库配置”栏中修改信息，再点击“修改”按钮，可修改该工业库服务器的配置参数。
 - 选中某一工业库，点击“删除”按钮，从“可访问的工业库”列表中删除该工业库的相关配置信息。
 - 选中某一工业库，点击“连接测试”按钮，可测试并提示与该工业库服务器是否能成功连接。
 - “全部测试”测试是否可与配置的所有工业库服务器成功连接。
- ◇ **选择当前记录历史数据的服务：**记录历史数据有三种选择。一般选择“组态王历史

库”选项，将历史数据直接存储到组态王历史库中；若用户购买了 KingHistorian 工业库软件，则可选择“工业库服务器”选项，将历史数据存储到已配置好的工业库服务器中；当然也可同时选择“组态王历史库”选项和“工业库服务器”选项将历史数据同时存储到组态王的历史库和工业库中。下面我们分别介绍两种选项的配置。

1) 组态王历史库：

点击“历史库”右边的“配置”按钮，弹出“历史记录配置”对话框，如图 24.5 所示。



图 24.5 历史记录配置

- **数据保存天数：**选择历史库保存的时间。最长为 8000 天，最短为 1 天。当到达规定的时间时，系统自动删除最早的历史记录文件。
- **数据存储所在磁盘空间小于：**磁盘存储空间不足时报警。当历史库文件所在的磁盘空间小于设置值时（设置范围 100—8000），系统运行后，将检测存储路径所在的硬盘空间，如果硬盘空间小于设定值，则系统给出如图 24.6 所示的提示：

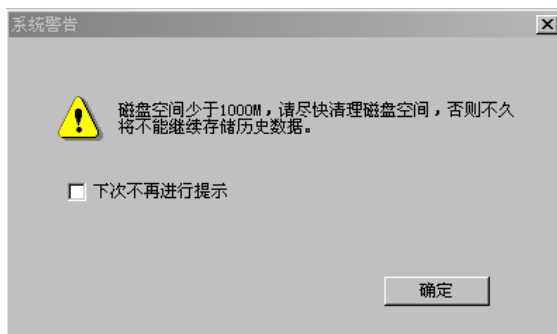


图 24.6 硬盘空间小于设定值时的提示

此时工程人员应该尽快清理磁盘空间，以保证组态王历史数据能够正常保存。如果在上图中选择了“下次不再进行提示”，则系统以后不再会出现如图 24.6 所示的提示，否则每隔 1 分钟提示一次。如果组态王运行系统检测到用于历史数据存储的硬盘空间过小（小于 50M），则会给出如图 24.7 所示的提示：

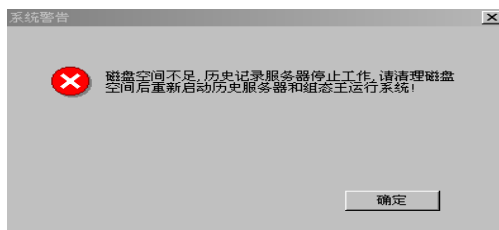


图 24.7 硬盘空间小于 50MB 时的提示

此时组态王运行系统已经停止历史记录，工程人员必须清理磁盘空间，然后再重新启动组态王历史服务程序和组态王运行系统，才能正常记录历史数据。

- **历史库存储路径的选择：**历史库的存储路径可以选择当前工程路径，也可以指定一个路径。如果工程为单机模式运行，则系统在指定目录下建立一个“本站

点”目录，存储历史记录文件。如果是网络模式，本机为“历史记录服务器”，则系统在该目录下为本机及每个与本机连接的 IO 服务器建立一个目录（本机的目录名称为本机的节点名，IO 服务器的目录名称为 IO 服务器的站点名），分别保存来自各站点的历史数据。

组态王的历史记录文件包括三种：`*.tmp`，`*.std`，`*.ev`。`*.tmp` 为临时的数据文件，`*.std` 为压缩的原始数据文件，`*.ev` 为进行了数据处理的特征值文件。

为了保证数据记录的快速和稳定，保证系统的运行效率，当被记录的变量的值发生变化或者定时记录中所设定的时间间隔到达后，组态王的历史记录首先被记录到临时文件中，该文件的文件名格式类似为 `project201008200700.tmp`（`project` 年月日时.`tmp`），每一小时生成一个，该文件不是压缩文件，而每到整点时间如 8 点，运行系统读出.`tmp` 文件中的数据进行压缩处理并生成真正的历史库记录文件——`.std` 文件，如 `project201008200700.std`。组态王将数据压缩处理完成后，生成一个新的时刻的.`tmp` 文件，上一个小时的.`tmp` 文件则被删除。组态王每一天的历史数据保存为一个.`std` 文件。

在每个整点时刻，组态王读出.`tmp` 文件中的数据进行压缩处理的同时，要对原始的数据文件进行算法过滤，存储一段时间内的特征点数据，生成.`ev` 文件，如 `project201000.ev`，每一年生成一个新的.`ev` 文件。

2) 工业库服务器:

点击“工业库服务器”右边的“配置”按钮，弹出“记录历史数据工业库配置”页面，在该页面配置工业库服务器的 IP 地址、端口号、登陆工业库的用户名、

密码等选项。



例如：

服务器：127.0.0.1（这里，工业库装在本地）

端口号：5678

用户名：sa

密码：sa

超时：0 毫秒

当连接工业库成功，用户可以继续配置“前缀名”、“后缀名”、“工程对应的组”及“组描述”。例如，设置工业库中的“前缀名”和“工程对应的组”为：khstation，即在工业库变量组中新建一个名为khstation2的子组，并将带有历史纪录的变量存储到该子组中。工程对应的组描述为：上岛电厂（2010_12_16）。工业库变量名称的命名规则为：工业库中的前缀名+“_”+组态王中的变量名+“.Value”。点击“开始”按钮，开始在工业库中添加和组态王对应的变量，如图24.8所示。

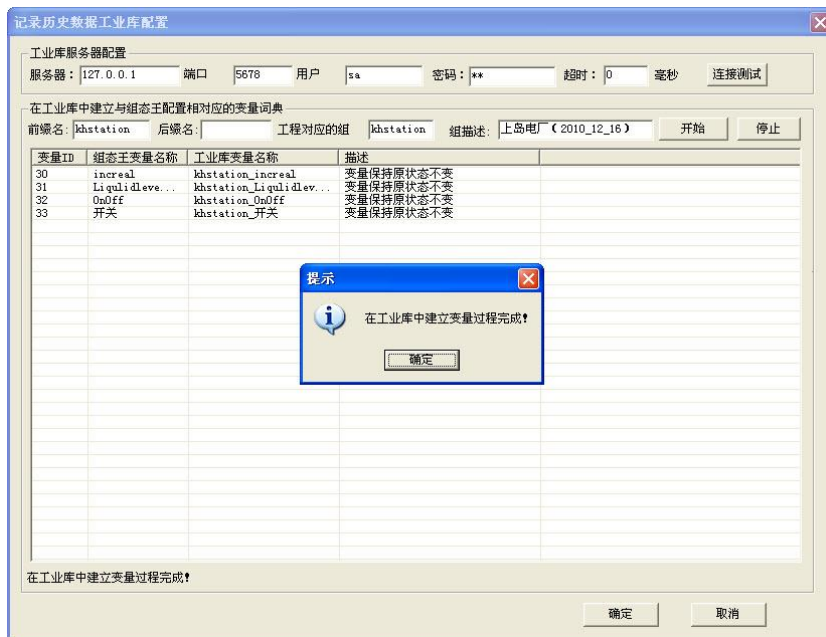


图 24.8 在工业库中添加变量词典

⚠ 注意:

用户可以即配置组态王的历史库，也配置工业库，从而同时把历史数据存储到组态王的历史库和工业库中。

24.3 历史数据的查询

在组态王运行系统中可以通过以下两种方式查询历史数据：报表、历史趋势曲线、WEB 发布中的历史数据和历史曲线的浏览。

✧ 使用报表查询历史数据。主要通过以下四个函数实现：

ReportsetHistdata() 、 ReportsetHistData2() 、 ReportsetHistData3() 和 ReportSetHistDataEx,有关这些函数的说明及使用方法请参见“第 17 章 报表系统”的“17.3.5 报表历史数据查询函数”或查看《命令语言函数速查手册》。

- ◇ 使用历史趋势曲线查询历史数据。组态王提供三种形式的历史趋势曲线：
 - 历史趋势曲线控件。历史趋势曲线控件的使用方法请参见“第 8 章 趋势曲线和其他曲线”的“8.3.3 历史趋势曲线控件”。
 - 图库中的历史趋势曲线。使用方法请参见“第 8 章 趋势曲线和其他曲线”的“8.3.2 通用历史趋势曲线”（暂不支持工业库历史数据的查询）。
 - 工具箱中的历史趋势曲线。使用方法请参见“第 8 章 趋势曲线和其他曲线”的“8.3.4 个性化历史趋势曲线”（暂不支持工业库历史数据的查询）。用户可以根据自己的需要来选择使用哪种曲线。
- ◇ 使用 WEB 历史数据发布。可以通过发布的数据视图或时间曲线查看组态王历史库或工业库中的数据,使用方法请参见“第 23 章 组态王 For Internet 应用”的“23.3.2 组态王历史数据发布”。



注意：

在历史记录配置中（如图 24.3），如果选择记录历史数据到“组态王历史库”，则可同时查询组态王历史库和工业库服务器中的历史数据。如果选择记录历史数据到“工业库服务器”，则只能查询工业库中的历史数据。另外，组态王暂不支持工业库实时数据的直接访问。

24.4 网络历史库的备份合并

在使用组态王网络功能时,有些系统中历史记录服务器与 I/O 服务器不是经常连接的,而是间断连接的,如拨号网络连接的网络系统。在这种情况下, I/O 服务器上变量的历史记录数据如果在网络不通的时候很容易丢失。

为了解决这个问题,组态王中专门提供了网络历史库存储“备份合并”的功能。在一般的网络里, I/O 服务器是不进行历史库记录的,而是将所有的数据都发送到历史记录服务器上记录。在组态王的“网络配置”中提供了一个选项“进行历史数据备份”。如图 24.9 所示。网络配置的其他选项及功能请参见“第 21 章 网络功能”。

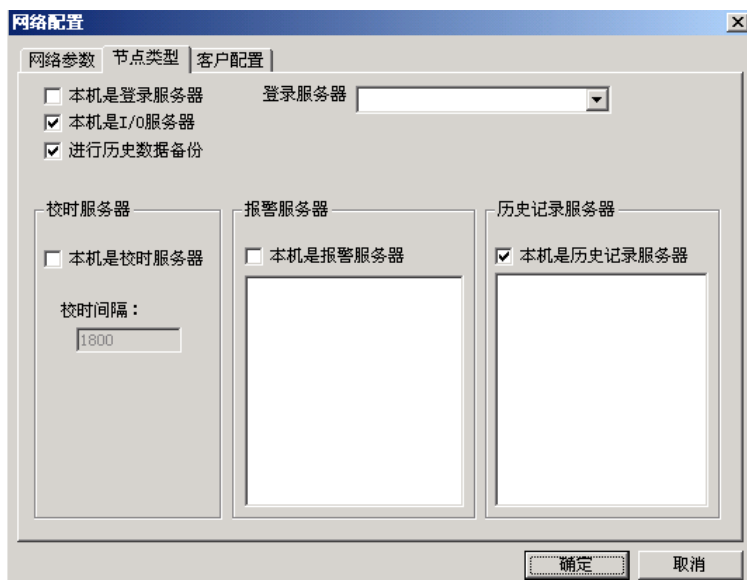


图 24.9 I/O 服务器上选择历史库备份

在 I/O 服务器上选择该项,则系统运行时, I/O 服务器自动记录本机产生的历史记录。

在历史记录服务器上建立远程站点时，可以看到 IO 服务器记录历史记录选项，如图 24.10 所示。

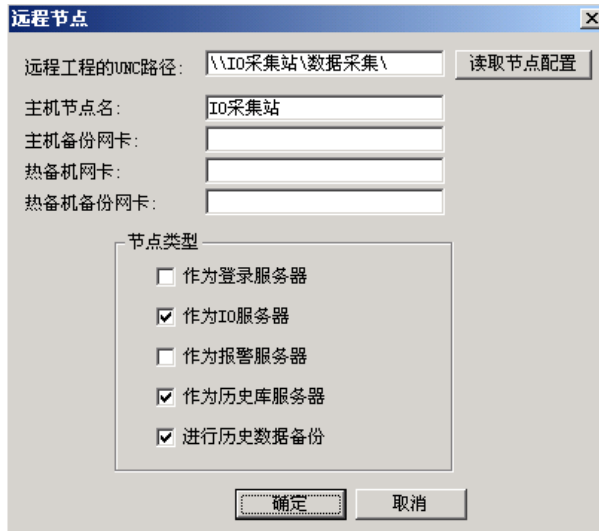



图 24.10 在历史记录服务器上建立远程站点

 注意：

进行历史数据备份的站点，必须在“网络配置”中选择“本机是历史记录服务器”，否则无法进行历史记录。

然后定义历史记录服务器的节点类型，如图 24.11 所示。

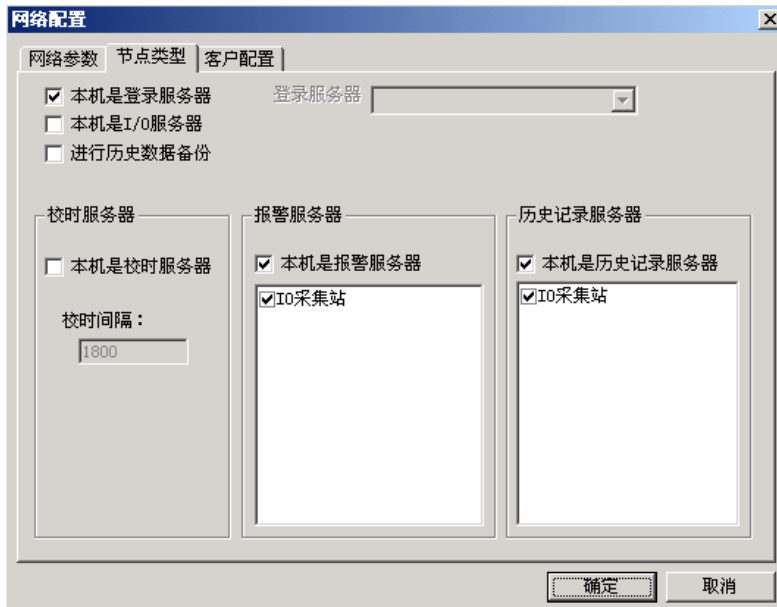


图 24.11 定义历史记录服务器的节点类型

这样当系统运行时，无论网络连通与否，历史记录服务器都不会记录来自 I/O 服务器上的变量的实时库中的值。在网络连通时，需要用户通过命令语言调用组态王提供的历史库备份函数——BackUpHistData（）来将 I/O 服务器上的历史数据传送到历史记录服务器上。

BackUpHistData (Str *chMchinename, Long ftEndtime)

功能：通过网络与远程站点建立连接，将远程站点的数据从上一次备份的时间到 ftEndtime 的时间段之间的数据备份到服务器中。

返回值:BOOL 型 当函数调用成功函数返回 TRUE, 否则返回 FALSE, 当函数返回 FALSE。

参数 1: chMchinename 字符串型, 远程站点 I/O 服务器名称, 服务程序通过该名称

与远程站点建立连接。

参数 2: ftEndtime 代表时间的长整型秒数, 指定备份的结束时间。可以使用

HTConvertTime () 函数来获得结束时间的整型。

例如: 在历史记录服务器上备份“IO 采集站”的历史记录到当前时间。在命令语言中按照以下方法实现。

```
long endTime;  
  
endTime=HTConvertTime($年,$月,$日,$时,$分,0);  
  
BackUpHistData("IO 采集站", endTime);
```

历史库每次备份完成后, 都会自动记录本次备份的结束时间, 下次备份时, 起始时间为上次备份结束的时间, 所以在使用备份函数时, 只需要定义结束时间即可。另外, 为了让用户更清楚的了解到备份的过程进度和结果, 以及获得出现的错误信息等, 组态王提供了以下函数供用户使用。

1. 获取备份进度函数

```
int GetBackupProgress( Str szStationName);
```

参数: 远程站点名称。

返回值: 0~100 间的整型数值。

功能: 得到正在备份的站点的进度百分比。

2. 获取备份状态函数

```
BOOL GetStationStatus(Str szStationName);
```

参数: 远程站点名称。

功能：得到站点的状态值，返回值>0 正在备份数据 返回值 =0 空闲。

3. 停止备份函数

```
BOOL StopBackupStation(Str szStationName);
```

参数：远程站点名称。

功能：强制停止正在备份的远程站点。



例如：

当用户使用 BackUpHistData () 函数后，可以在组态王的应用程序命令语言——运行时，或画面命令语言——存在时等里面周期性调用 GetBackupProgress () 函数和 GetStationStatus () 函数，获得备份进行的进度和备份的完成状态。如果用户想停止备份，可以在按钮中使用 StopBackupStation () 函数，立即停止历史记录备份。

24.5 组态王历史库服务程序

为了方便多用户访问组态王历史库，历史库程序采用了服务的方式，在组态王 7.0 安装完成后，请重新启动计算机，以使历史库服务启动。在系统控制面板中打开“服务”窗口，可以在服务列表中找到“HistorySvr”，请保证其状态为“已启动”。如果运行系统出现异常，可以在此处重新启动历史库服务。

如果在运行系统启动时，历史库服务没有启动，则系统会自动提示“历史库：历史库服务没有启动”。如果不启动历史库服务，则系统无法存储和查询历史数据。

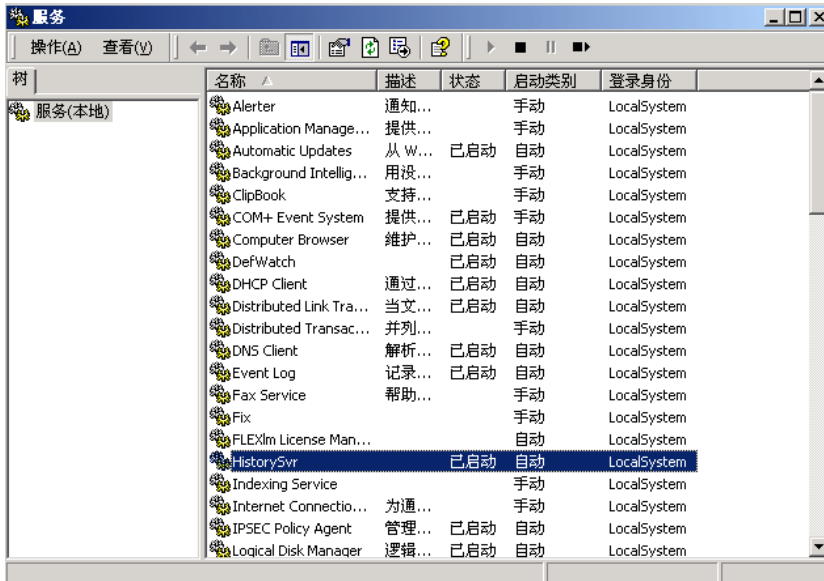


图 24.12 历史库服务

除本章所描述的内容之外，组态王 7.0 为了方便用户对历史库的访问，提供了可供用户使用 VBA 编程访问的接口，具体请参见本手册的 26.2.1 节。

第二十五章 其它程序访问组态王数据的方式

- ☞ 介绍访问组态王实时数据的方法
- ☞ 介绍访问组态王历史数据的接口

25.1 访问组态王实时数据

25.1.1 通过 OPC 访问组态王

通过组态王 OPC 服务器功能,用户可以更方便的实现其他支持 OPC 客户的应用程序与组态王之间的数据通讯和调用。使用方法在“第二十章 OPC 设备”中详细介绍,这里就不再介绍。

25.1.2 通过 DDE 访问组态王

“组态王”支持动态数据交换 (DDE: Dynamic Data Exchange),能够和其他支持动态数据交换的应用程序方便地交换数据。通过 DDE, 工程人员可以利用 PC 机丰富的软件资源来扩充“组态王”的功能。使用方法在“第十八章 组态王与其它应用程序的动态数据

交换（DDE）”中详细介绍，这里就不再介绍。

25.1.3 通过 ActiveX 控件访问

组态王提供客户端控件 KvTcpiClientOcx.ocx，用户通过在第三方程序中调用该控件，可以获得组态王实时数据，也可以订阅组态王实时数据。使用该控件之前，需要先运行组态王运行系统。KvTcpiClientOcx.ocx 提供了控件方法和控件事件供用户调用。下面将对这些方法和事件进行说明。

25.1.3.1 KvTcpiClientOcx.ocx 控件提供的方法说明

提供的控件方法	方法函数
登录组态王的实时变量服务器	Method_LoginServer
退出登录组态王的实时变量服务器	Method_LogoutServer
载入指定路径下的工程文件，并运行该工程	SetProjectPath(LPCTSTR project_path)
取得站点数	Method_GetStationNumber
取得站点名	Method_GetStationName
取得某站点下变量数	Method_GetVariableNumber
取得某站点下某变量的变量名称	Method_GetVariableName
取得某站点下某变量的变量值(根据变量ID)	Method_GetVariableValueByVariableId
取得某站点下某变量的变量值(根据变量名)	Method_GetVariableValueByVariableName
取得某站点下某变量的带时间和质量戳的变量值(根据变量ID)	Method_GetVariableValueWithStampByVariableId

取得某站点下某变量的带时间和质量戳的变量值(根据变量名)	Method_GetVariableValueWithStampByVariableName
订阅某站点下某变量的当前值变化, 质量戳变化, 时间戳变化	Method_SubscribeVariable
根据变量ID设置某站点下某变量的当前值	SetVariableValueByVariableId
根据变量名设置某站点下某变量的当前值	SetVariableValueByVariableName

方法说明:

```
short Method_LoginServer(LPCTSTR server_ip_address, unsigned short server_port, LPCTSTR user_name, LPCTSTR user_password)
```

功能: 登录组态王的实时变量服务器。

参数: server_ip_address: 服务器(组态王) IP地址

server_port: 服务气端口号

user_name: 用户名

user_password: 用户密码

返回值: 0: 登录失败

- 1: 登录通讯五秒钟内没有应答
- 2: 登录成功(用户具有最高级别 999 时才具有写入权限)
- 3: 登录通讯五秒钟内有应答, 但客户 CliendId 没有正确分配
- 4: 已处在登录状态
- 11: 非法用户, 验证失败

short Method_LogoutServer()

功能：退出登录组态王的实时变量服务器。

返回值：0: 尚未登录

1: 退出登录成功

short SetProjectPath(LPCTSTR project_path)

功能：载入指定路径下的工程文件，并运行该工程。

参数：project_path是组态王的工程路径，比如D:\Kingview\Project\注意，这个路径下就是工程文件了。

返回值：0 表示载入成功

1 表示创建窗口失败

3 表示工程文件路径错误

3 表示创建、启动实时库失败。

short Method_GetStationNumber(unsigned short * nStationNum)

功能：获取站点的数量。

参数：nStationNum 要求传入一个USHORT型变量地址

返回值：0 表示成功

1 表示失败，一般都是因为Web控件还没有加载工程导致的。

short Method_GetStationName(unsigned short nStation_ID, BSTR * strStation_Name)

功能：获取指定站点的名称，并将内容写入strStation_Name指定的字符串中。

参数：nStation_ID指站点ID，可以通过上一个方法来获得站点的总数量；

strStation_Name是字符串地址，用来接收Web控件写入的站点名。

返回值：0 成功；

1 Web控件未启动；

2 通过ID在实时库中读取名称失败。

说明：如果nStation_ID相应的站点在服务端不存在，则strStation_Name="NULL"。

short Method_GetVariableNumber(unsigned short station_id, unsigned long * variable_number)

功能：这个方法是通过站点序列号，获得该站点下的所有变量的数量。

参数： station_id: 站点ID

variable_number: 变量数

返回值：0 成功

1 失败

short Method_GetVariableName(unsigned short station_id, unsigned long variable_id, BSTR * variable_name)

功能：通过站点ID，变量ID，获取指定变量的名称。

参数: station_id: 站点ID

variable_id: 变量ID

variable_name: 变量名

返回值: 0 成功

1 Web控件未启动

2 在实时库中获取名称失败

`short Method_GetVariableValueByVariableId(unsigned short station_id, unsigned long variable_id, short * variable_value_type, BSTR* variable_value_string)`

功能: 取得某站点下某变量的变量值(根据变量ID)

参数: station_id: 站点ID

variable_id: 变量ID

variable_value_type: 变量类型, 变量类型的描述为

整型 3

实型 4

离散型 11

字符串型 16400

variable_value_string: 变量值

返回值: 0 成功

1 web控件未启动

2 访问本地库或者远程库失败

```
short Method_GetVariableValueByVariableName(unsigned short station_id, LPCTSTR  
variable_name, short * variable_value_type, BSTR* variable_value_string)
```

功能：取得某站点下某变量的变量值(根据变量名)。

参数： station_id: 站点ID

variable_name: 变量名

variable_value_type: 变量类型，变量类型的描述为

整型 3

实型 4

离散型 11

字符串型 16400

variable_value_string: 变量值

返回值：0 成功

1 web控件未启动

2 访问本地库或者远程库失败

```
short Method_GetVariableValueWithStampByVariableId(unsigned short station_id,  
unsigned long variable_id, short * variable_value_type, BSTR*  
variable_value_string, unsigned long * stamp_time_hi, unsigned long *
```

`stamp_time_lo, unsigned short * stamp_quality)`

功能：取得某站点下某变量的带时间戳和质量戳的变量值(根据变量ID)

参数： station_id: 站点ID

variable_id: 变量ID

variable_value_type: 变量类型，变量类型的描述为

整型 3

实型 4

离散型 11

字符串型 16400

variable_value_string: 变量值

stamp_time_hi: 时间戳的高位

stamp_time_lo: 时间戳的低位

stamp_quality: 变量的质量戳

返回值：0 成功

1 web控件未启动

2 访问本地库或者远程库失败

`short Method_GetVariableValueWithStampByVariableName(unsigned short station_id, LPCTSTR variable_name, short * variable_value_type, BSTR* variable_value_string, unsigned long * stamp_time_hi, unsigned long * stamp_time_lo, unsigned short *`

stamp_quality)

功能：取得某站点下某变量的带时间和质量戳的变量值(根据变量名)

参数： station_id: 站点ID

variable_name: 变量名

variable_value_type: 变量类型, 变量类型的描述为

整型 3

实型 4

离散型 11

字符串型 16400

variable_value_string: 变量值

stamp_time_hi: 时间戳的高位

stamp_time_lo: 时间戳的低位

stamp_quality: 变量的质量戳

返回值： 0 成功

1 web控件未启动

2 访问本地库或者远程库失败

```
short Method_SubscribeVariable(unsigned short station_id, unsigned long  
variable_id, unsigned short subscribe_type, unsigned short nSubOrNot)
```

功能：订阅某站点下某变量的当前值变化, 质量戳变化, 时间戳变化

参数: station_id: 站点ID从0开始的

variable_id: 变量ID从0开始的

subscribe_type: 订阅类型

第0位决定是否订阅值的变化 (1为订阅 0为不订阅)

第1位决定是否订阅质量戳的变化 (1为订阅 0为不订阅)

第2位决定是否订阅时间戳的变化 (1为订阅 0为不订阅)

返回值 0: 订阅或取消订阅成功

1: 控件没有启动

2: 订阅不存在的变量。

short Method_SetVariableValueByVariableId(unsigned short station_id, unsigned long variable_id, LPCTSTR value_string)

功能: 根据变量ID设置某站点下某变量的当前值

参数: station_id: 站点ID

variable_id: 变量ID

LPCTSTR value_string: 待写入的变量值

返回值: 0 OK

1 Web控件未启动

2 传入的变量是空的

3 请求不存在变量的值

- 4 不可对内部保留变量设置值
- 5 不可对只读变量设置值
- 6 变量值太短
- 7 变量值太长
- 8 输入的值与变量类型不符

short Method_SetVariableValueByVariableName(unsigned short station_id, LPCTSTR variable_name, LPCTSTR value_string)

功能：根据变量名设置某站点下某变量的当前值。

参数：station_id：站点ID

variable_name：变量名

LPCTSTR value_string：变量值

返回值：0 OK

- 1 Web控件未启动
- 2 传入的变量是空的
- 3 请求不存在变量的值
- 4 不可对内部保留变量设置值
- 5 不可对只读变量设置值
- 6 变量值太短
- 7 变量值太长

8 输入的值与变量类型不符

25.1.3.2 Kvtcpclientocx.ocx 中提供的事件说明

Event_LoginServerOK

功能：登录服务器成功并取得客户号通知。

Event_LoginServerOKKvtcpclientocxctrl(unsigned long client_id)

参数： client_id: 客户端ID

Event_LogoutServerOk

Event_LogoutServerOkKvtcpclientocxctrl()

功能：退出登录服务器成功通知。

Event_ServerClose

Event_ServerCloseKvtcpclientocxctrl()

功能：服务器关闭服务通知。

Event_ServerDisconnect

Event_ServerDisconnectKvtcpclientocxctrl()

功能：与服务器失去连接通知。

Event_VariableValueChanged

功能：变量值改变通知(带时间戳和质量戳)。

```
Event_VariableValueChangedKvtcpipclientocxctrl(unsigned short station_id,  
unsigned long variable_id, short variable_value_type, LPCTSTR  
variable_value_string, unsigned long stamp_time_hi, unsigned long  
stamp_time_lo, unsigned short stamp_quality)
```

参数：station_id: 站点ID

variable_id: 变量ID

variable_value_type: 变量类型

variable_value_string: 变量值

stamp_time_hi: 时间戳的高位

stamp_time_lo: 时间戳的低位

stamp_quality: 变量的质量戳

Event_VariableStampTimeChanged

功能：变量时间戳改变通知。

```
Event_VariableStampTimeChangedKvtcpipclientocxctrl(unsigned short station_id,  
unsigned long variable_id, unsigned long stamp_time_hi, unsigned long
```


`stamp_time_lo)`

参数: `stamp_time_hi`: 时间戳的高位
`stamp_time_lo`: 时间戳的低位

Event_VariableStampQualityChanged

功能: 变量质量戳改变通知(带时间戳)。

**Event_VariableStampQualityChangedKvtcpipclientocxctrl(unsigned short
station_id, unsigned long variable_id, unsigned long stamp_time_hi, unsigned
long stamp_time_lo, unsigned short stamp_quality)**

参数: `station_id`: 站点ID
`variable_id`: 变量ID
`stamp_time_hi`: 时间戳的高位
`stamp_time_lo`: 时间戳的低位
`stamp_quality`: 变量的质量戳

Event_OcxMessage

功能: 控件中其他(出错和调试信息)消息通知。

Event_OcxMessageKvtcpipclientocxctrl(unsigned long message_type, LPCTSTR

message_buf)

25.1.3.3 stamp_time_hi 和 stamp_time_lo 说明

stamp_time_hi 和 stamp_time_lo 分别对应 c++ 语言中的 FILETIME 数据类型中的 dwHighDateTime 和 dwLowDateTime。

可以使用如下函数将 stamp_time_hi 和 stamp_time_lo 转化成时间字符串。

```
void ChangeStampTimeToString(const DWORD stamp_time_hi, const DWORD  
stamp_time_lo, CString &string_buf)
```

```
{
```

```
FILETIME aFILETIME;
```

```
aFILETIME.dwHighDateTime=stamp_time_hi;
```

```
aFILETIME.dwLowDateTime=stamp_time_lo;
```

```
FILETIME aFILETIME_Local;
```

```
FileTimeToLocalFileTime(&aFILETIME, &aFILETIME_Local) ;
```

```
SYSTEMTIME aSYSTEMTIME;
```

```
FileTimeToSystemTime(&aFILETIME_Local, &aSYSTEMTIME) ;
```

```
string_buf.Format(_T("%04d-%02d-%02d %02d:%02d:%02d:%03d"), aSYSTEMTIME.wYear,  
aSYSTEMTIME.wMonth, aSYSTEMTIME.wDay, aSYSTEMTIME.wHour, aSYSTEMTIME.wMinute, aSY
```

```
STEMTIME.wSecond, aSYSTEMTIME.wMilliseconds);  
}
```

25.1.3.4 应用 KvRealDBClient 控件注意事项

1. 关于版本

在高于VC6.0版本的开发环境中使用该控件开发自己的应用时，需要显式指明使用早期版本映射方式，如将'USHORT*'等映射为VTS_PI2等。

操作步骤：

第一步：在工程中插入该控件，IDE会自动为工程添加CKvTcpipClient0cx类（类名可能稍有差异）

第二步：在CKvTcpipClient0cx类的CPP文件中作如下修改：

```
short Method_GetStationNumber(.....)  
  
{  
  
    short result;  
  
    static BYTE parms[] = VTS_PUI2 ;// 修改为 VTS_PI2  
  
    .....  
  
}  
  
...  
  
short Method_GetVariableNumber(.....)  
  
{  
  
    short result;
```

```
static BYTE parms[] = VTS_UI2 VTS_PUI4 ;//修改为 VTS_PI4
.....
}
...
short Method_GetVariableValueWithStampByVariableId(.....)
{
    short result;

    static BYTE parms[] = VTS_UI2 VTS_UI4 VTS_PI2 VTS_PBSTR VTS_PUI4
    VTS_PUI4 VTS_PUI2;// 修改为 VTS_PI4 VTS_PI4 VTS_PI2
    .....
}

short Method_GetVariableValueWithStampByVariableName(.....)
{
    short result;

    static BYTE parms[] = VTS_UI2 VTS_BSTR VTS_PI2 VTS_PBSTR VTS_PUI4
    VTS_PUI4 VTS_PUI2;// 修改为 VTS_PI4 VTS_PI4 VTS_PI2
    .....
}
```

2. 关于参数 `variable_value_type` 的返回类型

“`variable_value_type`”是一些函数的参数，可以通过其返回值，获得变量的数据类

型。具体为：

3：整型

4：实型

11：离散

16400：字符串

25.1.3.5 KvRealDBClient 工具使用简介

KvRealDBClient.exe 是亚控公司使用 VC++开发的客户端访问组态王实时数据的演示工具，该工具中调用了 KvTcpiClient0cx.ocx 控件的方法和事件。

KvRealDBClient.exe 位于组态王安装路径...\\Kingview 下，在使用该工具之前请先启动服务端的组态王运行系统。运行 KvRealDBClient.exe，界面如图 25.1 所示：

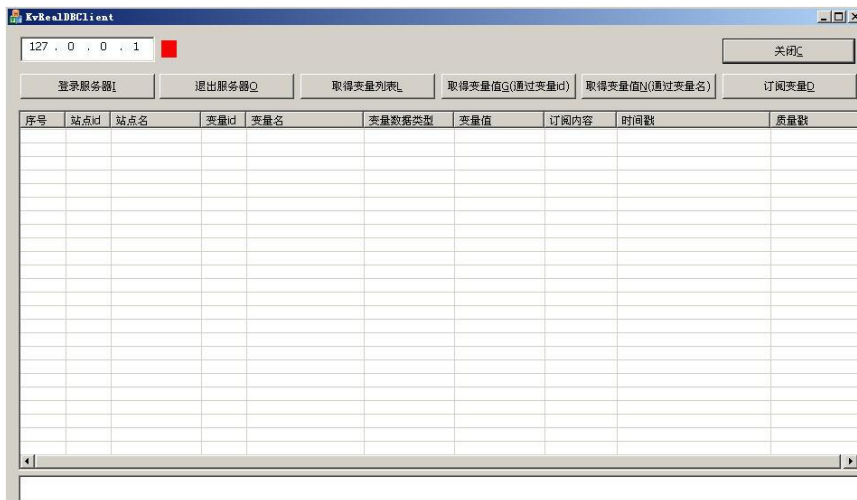


图 25.1 KvRealDBClient 启动界面

◇ **登录服务器：** 在该按钮上侧的编辑框内输入服务器 IP 地址，点击“登录服务器”

按钮，弹出“用户验证”信息框。



输入正确的用户名和密码，即可登录。用户的优先级为 999 时，可以对数据进行读写操作，如图 25.2 所示。当用户的优先级小于 999 时，用户只具有读数据的权限，如图 25.3 所示。



图 25.2 读写操作界面

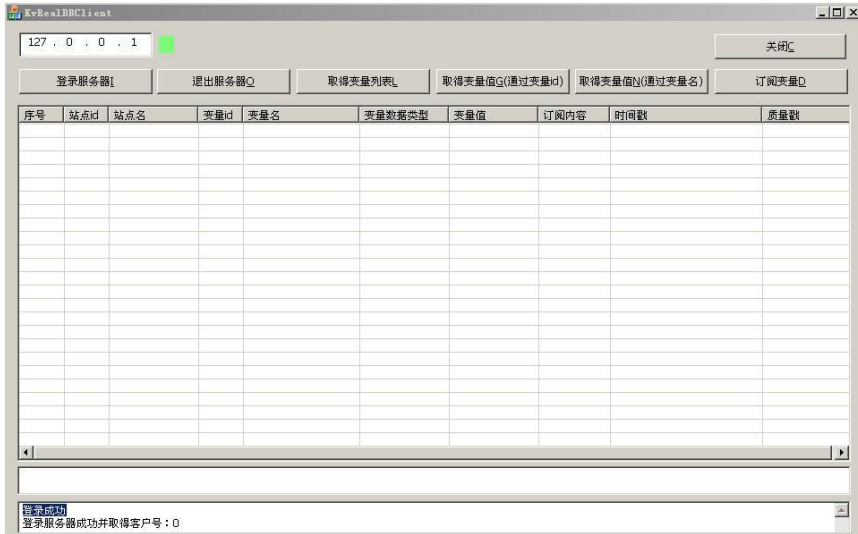
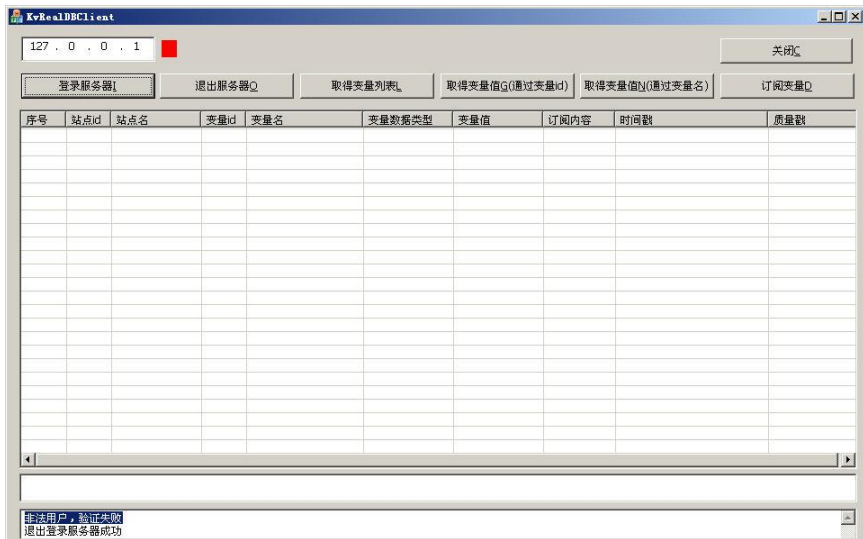


图 25.3 读操作界面

如果输入错误的用户名和密码，则返回登陆失败提示信息，如下图所示。



在进行组态王实时数据访问之前必须先登录服务器。下面以读写操作登录界面为例，对各个按钮作相应的介绍。

- ◇ **退出服务器：** 点击该按钮，退出登录。
- ◇ **取得变量列表：** 点击该按钮，取得服务器的变量列表。如图 25.4 所示。



图 25.4 取得变量列表

- ◇ **取得变量值（通过变量 id）：**在取得的变量列表中选择单个或多个变量，点击该按钮，通过变量 id 取得变量值（带时间戳和质量戳）。如图 25.5 所示。



图 25.5 取得变量值

- ✧ **取得变量值（通过变量名）**：在取得的变量列表中选择单个或多个变量，点击该按钮，通过变量名取得变量值（带时间戳和质量戳）。功能上与“取得变量值（通过变量 id）”按钮相同。
- ✧ **订阅变量**：在取得的变量列表中选择单个或多个变量，点击该按钮，系统弹出如图 25.6 所示的对话框。

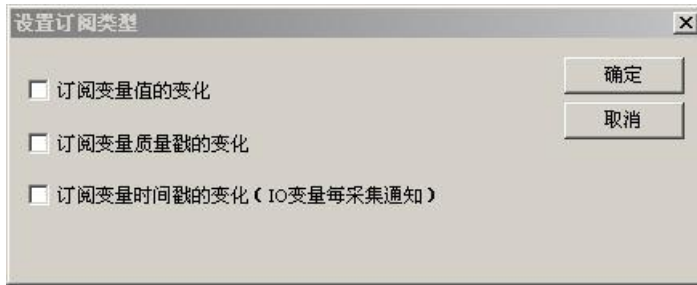


图 25.6 设置订阅类型

在该对话框中设置订阅类型，确定，按照设置的类型进行实时数据的订阅。

- ✧ **设置变量值 (通过变量 id):** 在取得的变量列表中选择单个或多个变量，点击该按钮，通过变量 id 取得变量值 (带时间戳和质量戳)。




输入要设置的值，单击“确定”按钮，实现数据的设置。如图 25.7 所示。



图 25.7 设置变量值

- ◇ **设置变量值（通过变量名）**：在取得的变量列表中选择单个或多个变量，点击该按钮，通过变量名取得变量值（带时间戳和质量戳）。功能上与“设置变量值（通过变量 id）”按钮相同。

 **注意：**

不可对内部保留变量和只读变量进行数值设置。

- ◇ **关闭**：关闭 KvRealDBCClient.exe。

25.2 访问组态王历史数据

25.2.1 AccessHist 接口

25.2.1.1 接口函数

“组态王”提供的 VBA 接口为“AccessHist.dll”，该文件位于组态王安装目录下。
提供的接口函数有以下几种：

LoadProject(ByVal projectpath As String) As Integer

功能：加载要查询历史数据的工程。如果是单机则为当前工程路径。如果是网络模式，则应该为历史记录服务器工程的路径。

参数：projectpath 字符串型 工程完整路径和工程信息文件名称

返回值：整型 为 1 时，表示加载成功；

为 0 时，表示加载失败。

SearchVba (ByVal sStationName As String, ByVal sTagName As String, ByVal StartTime As Long, ByVal EndTime As Long, ByVal Interval As Long, pArray As Double, ByVal VarNum As Long) As Integer

功能：查询符合条件的历史数据。

参数：sStationName：字符串型 表示要查询变量所在的站点名称。如果为单机，则该值为[\\本站点](#)。

sTagName：字符串型 表示要查询的变量名称。

StartTime：长整型 表示查询要查询数据的起始时间，该时间为自 1970 年 1 月 1 日 00:00:00 即 UCT 起到该时刻所经过的秒数。

EndTime：长整型 表示查询要查询数据的结束时间，该时间为自 1970 年 1 月 1

日 00:00:00 即 UCT 起到该时刻所经过的秒数。在查询时，结束时间必须大于起始时间。

Interval: 长整型 查询间隔时间。单位为毫秒。

PArray: 数组 表示存储查询到数据集的数组。

VarNum: 长整型 表示每个变量查找到的数据个数。

返回值: 整型 为 1 时，表示查询成功
为 0 时，表示查询失败

UnloadProject () As Integer

功能: 卸载查询时加载的工程

参数: 无

返回值: 整型 为 1 时，表示卸载成功
为 0 时，表示卸载失败

说明: 在每次加载工程后，无论查询正常与否，必须进行工程卸载。

ConvertTime(ByVal Year As Integer, ByVal Month As Integer, ByVal Day As Integer, ByVal Hour As Integer, ByVal minute As Integer, ByVal Second As Integer) As Long

功能: 将查询所使用的年、月、日、时、分、秒表示的时间转换成自 1970 年 1 月 1 日 00:00:00 即 UCT 起到该时刻所经过的秒数。

参数: Year: 年，此值必须介于 1970 和 2032 之间

Month: 月，此值必须介于 1 和 12 之间

Day: 日，此值必须介于 1 和 31 之间

Hour: 小时, 此值必须介于 0 和 23 之间

Minute: 分钟, 此值必须介于 0 和 59 之间

Second: 秒, 此值必须介于 0 和 59 之间

返回值: 长整型 表示自 1970 年 1 月 1 日 00:00:00 即 UCT 起到该时刻所经过的秒数。


ascTime(ByVal lStartTime As Long, ByVal lInterval As Long) As String

功能: 返回指定的以长整型数表示的秒数的时间字符串。

参数: lStartTime: 查询的起始时间。自 1970 年 1 月 1 日 00:00:00 即 UCT 起到该时刻所经过的秒数。

lInterval: 查询的间隔。为每个查询的起始基准点到每个查询点时间的毫秒数。

返回值: 表示查询点时间的时间字符串。

 注意:

以上函数为组态王 6.51 以前版本提供的 API 接口函数, 在组态王 6.53 及以后的版本中, 提供下面新的 API 接口函数。所有这些函数在组态王 6.53 及以后版本中都支持。用户可以根据需要使用。

GetLogVarNameByIndex (ByVal sStationName As String, _ByVal lIndex As Long) As String

功能: 通过索引号得到工程中设定有历史记录属性的变量名称。

参数: sStationName: 工作站名称。

lIndex: 变量的索引 (索引为: 0~有历史记录属性的变量个数-1, 具体个数由

GetLogVarCount () 函数获得)

返回值：变量名称。

GetLogVarCount (ByVal sStationName As String) As Long

功能：得到工程中设定有历史记录属性的变量的总数。

参数：sStationName：工作站名称。

返回值：有历史记录属性的变量的个数。

GetStationCount () As Long

功能：得到工程中定义为历史库服务器站点的个数。

参数：无。

返回值：历史库服务器站点的个数。

GetStationNameByIndex (ByVal lIndex As Long) As String

功能：通过索引得到工程中定义为历史库服务器站点的名称,常用来列举所有站点的名称

参数：lIndex：站点的索引。

返回值：历史库服务器站点的名称。

GetStationIDByName (ByVal sStation As String) As Long

功能：得到工程中定义为历史库服务器站点的索引。

参数：sStationName：工作站名称。

返回值：工作站的索引。

GetSearchDataCountFromKvDB_ForVB (ByVal sStation As String, ByVal VarName As String, ByVal StartTime As Long, ByVal EndTime As Long, ByVal lSearchFlag As Long,

ByVal lParam As Long, lVarNum As Long) As Integer

功能：根据查询的参数得到查询返回数据的个数。

参数：sStationName：工作站名称

VarName：查询的变量的名称。

StartTime：查询的开始时间。值为据格林威治基准时间 1970 年 1 月 1 日 00 时 0 分 0 秒的秒数。

EndTime：查询的结束时间。值为据格林威治基准时间 1970 年 1 月 1 日 00 时 0 分 0 秒的秒数。

LSearchFlag： 数据查询的方式：

SEARCH_FLAG_FULL--查询全部数据

SEARCH_FLAG_INTERVAL--间隔查询

SEARCH_FLAG_EIGEN--特征值查询

LParam：用于不同的查询方式应该传入的不同参数，查询全部数据方式时此值不起作用，间隔查询时传入间隔的时间（秒）。

LVarNum：输出参数，为返回的数据的个数。

返回值：是否调用正确。0 -- 失败

1 --- 成功。

SearchDataFromKvDB_ForVB (ByVal sStation As String, ByVal VarName As String, ByVal lStartTime As Long, ByVal lEndTime As Long, ByVal lSearchFlag As Long, ByVal lParam As Long, pArrdblVarValue As Double, pArrftTime As FILETIME_VB,

`pArrlVarQuality As Long, ByVal lVarNum As Long) As Integer`

功能：得到查询返回的数据。

参数：sStationName：工作站名称

VarName：查询的变量的名称。

lStartTime：查询的开始时间。值为据格林威治基准时间 1970 年 1 月 1 日 00 时 0 分 0 秒的秒数。

lEndTime：查询的结束时间。值为据格林威治基准时间 1970 年 1 月 1 日 00 时 0 分 0 秒的秒数。

LSearchFlag：数据查询的方式：

SEARCH_FLAG_FULL--查询全部数据

SEARCH_FLAG_INTERVAL--间隔查询

SEARCH_FLAG_EIGEN--特征值查询

LParam：用于不同的查询方式应该传入的不同参数，查询全部数据方式时此值不起作用，间隔查询时传入间隔的时间（秒）。

PArrdblVarValue：返回的包含变量值的数组。

ParrftTime：返回的包含时间戳的数组。

PArrlVarQuality：返回的包含质量戳的数组。

LVarNum：输出参数，为返回的变量的个数。

返回值：是否调用正确。0 -- 失败

1 --- 成功。

ascFileTime Lib (ftVarTime As FILETIME_VB, ByVal lFmtType As Long) As String

功能：得到时间戳的用字符串表示的时间。

参数：ftVarTime：需要表示的时间戳。

LFmtTypeee：不同的时间表示方式。

返回值：表示时间的字符串。

25.2.1.2 接口函数的使用

在 VBA 中调用函数，首先需要声明函数。如下图所示，打开 EXCEL，进入 VBA 编辑器，建立一个模块，在模块中声明函数。声明函数时动态库所在的路径根据组态王在本机安装的路径确定。函数声明内容如图：



函数声明完成之后，就可以建立对话框体，允许用户输入查询的时间、站点名称、变量名称等参数。输入完成后，可以开始查询。组态王为此提供了一个演示用的程序——

“kingReport.xls”，该文件位于组态王安装目录下。供用户使用时参考。下面简单介绍该程序的使用方法。

25.2.2 KingReport

Microsoft Excel 是 Microsoft 公司推出的具有强大功能的报表生成系统，将“组态王”与“Excel”结合起来，可实现各种复杂的报表。

组态王历史库提供的历史数据查询 VBA 接口提供三种查询方式：

◇ 等间隔查询：

按照用户提供的起止日期、时间，和给定的查询时间间隔，按照等间隔的时间从组态王历史库中查询数据，如果某个等间隔时间点上没有记录，则系统自动查找历史记录中离被查询点前面最近的一个数据返回。

◇ 全部查询：

返回给定时间范围内被查询变量的所有原始历史记录数据。

◇ 特征值查询：

查询给定时间范围内变量的特征值记录数据。（关于特征值的定义和记录方式请参见《组态王使用手册》“第二十五章 组态王历史库”）。查询返回的数据量大时，建议用户采用特征值查询方式。

在 Excel 中利用 Visual Basic 将“组态王”历史库中的数据读入到 Excel 的表格中，示例“kingreport.xls”演示了如何将几个变量的历史数据显示到 Excel 中。在 Kingreport.xls 示例中，上述三种查询方式都包含在内。下面介绍该示例的使用方法。

启动 Excel，打开“kingreport.xls”，此时菜单中自动增加一项内容“历史报表”，点此菜单。如下图所示。

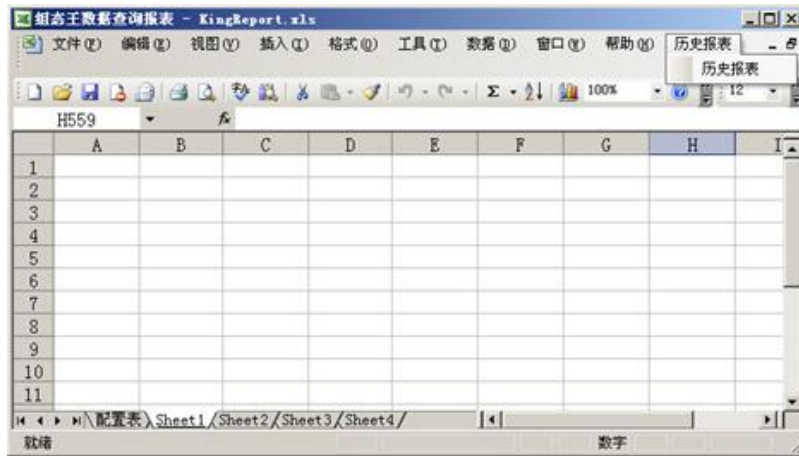


图 25.8 组态王历史数据查询报表

继续单击菜单“历史数据查询”，则会弹出“查询参数设置”对话框，如下图所示。




图 25.9 设置查询参数值

参数设置对话框上各项含义如下：

- **请输入工程路径：** 在编辑框中输入要查询历史数据的变量所在工程的路径及工程信息文件——project.pro，如“C:\Program Files\Kingview\Example\Kingdemo3\project.pro”。或点击“？”按钮选择工程路径及project.pro文件，如下图所示。




图 25.10 选择工程路径

 注意：

- 1、 输入工程路径时必须包含工程信息文件——project.pro。
- 2、 工程信息文件 project.pro，只有组态王在运行系统启动后才会生成。为保证信息文件中的内容正确，请在执行查询前启动组态王运行系统。特别是在对有历史记录属性的变量，或其它与历史记录相关的属性进行修改后，要首先启动组态王运行系统，重新生成新的 project.pro 文件，保证查询顺利进行。

- **请选择采集站名称：**输入或选择工程所在的采集站的计算机名称。

 注意：

在输入站点名称时，直接输入节点名称，在名称前不能加表示 UNC 路径的“\\”。

如果是本机，工程是单机模式，直接输入“本站点”。

- **起始时间、结束时间：**在日期和时间控件中选择要查询历史数据的起止日期和时间。注意，选择的结束时间不能小于起始时间。
- **查询类型：**在列表框中选择要查询的方式。关于查询方式的说明请参见本节开始处的内容。
 - 如果选择“间隔查询”，界面上显示“查询间隔”编辑框。在编辑框中输入查询历史数据需要的时间间隔，可以输入毫秒。



查询类型: 间隔查询

查询间隔: 60 秒 0 毫秒

- 如果选择“全部查询”，则不需要输入额外的参数。查询后，会返回起始和结束时间的所有历史数据。
 - 如果选择“特征值查询”，界面显示“查询参数”编辑框。如下图所示，在编辑框中输入查询的参数。

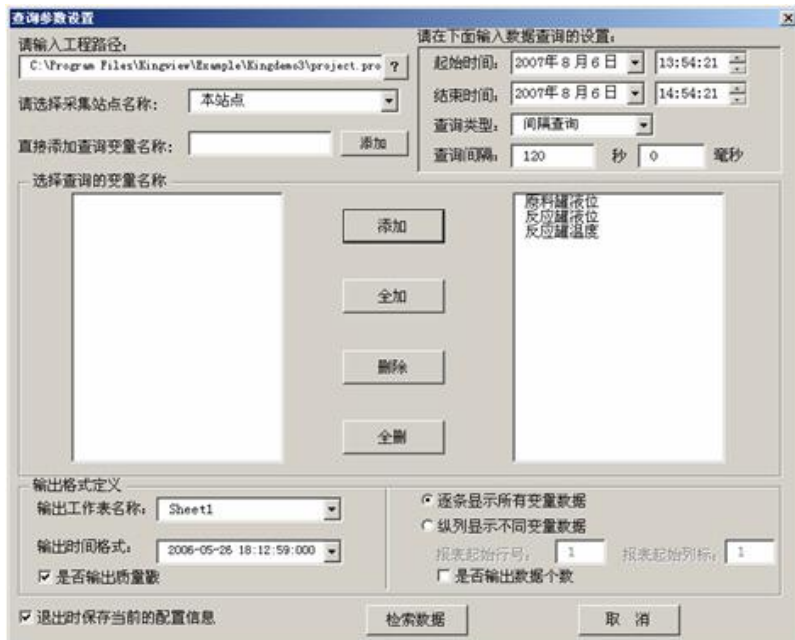


查询类型: 特征值查询

查询参数: 2068

- **选择查询的变量名称：**在列表中显示了连接的工程中定义有历史记录属性的变量，选择列表中的变量名称，通过按钮“全加”、“添加”、“删除”、“全删”等，选择要进行历史数据查询的变量，被选中的变量显示在右侧的列表框中。
- **输出表格式定义：**定义历史数据查询后要填充数据的表格，以及相关格式。

- **输出工作表名称:** 选择输出到当前 Excel 文件的哪个工作表, 默认为 sheet1。
- **输出时间格式:** 选择数据填充表格后, 数据对应时间的显示格式, 单击下拉列表框选择格式样式。
- **是否输出质量戳:** 如果选中该项, 在历史数据查询结果集中, 每条记录将包含对应数据的质量戳数据, 并填充到表格中。否则不会显示。
- **逐条显示所有变量数据:** 查询结束后, 在填充表格时, 先将第一个变量的所有数据填充到表格中, 然后第二个……。如下图所示, 选择输出质量戳、逐条显示所有变量数据。



点击“检索数据”按钮，执行后，数据填充到 sheet1 表中，如下图所示。



The screenshot shows a Microsoft Excel spreadsheet titled '组态王数据查询报表 - KingReport.xls'. The spreadsheet contains a table with the following data:

	A	B	C	D	E
1	变量名	时间	数据值	质量戳	
2	原料罐液位	2007-08-31 13:42:21:000	265	192	
3	原料罐液位	2007-08-31 13:43:21:000	25	192	
4	原料罐液位	2007-08-31 13:44:21:000	130	192	
5	原料罐液位	2007-08-31 13:45:21:000	415	192	
6	原料罐液位	2007-08-31 13:46:21:000	260	192	
7	原料罐液位	2007-08-31 13:47:21:000	115	192	
8	反应罐液位	2007-08-31 13:42:21:000	370	192	
9	反应罐液位	2007-08-31 13:43:21:000	520	192	
10	反应罐液位	2007-08-31 13:44:21:000	670	192	
11	反应罐液位	2007-08-31 13:45:21:000	610	192	
12	反应罐液位	2007-08-31 13:46:21:000	530	192	
13	反应罐液位	2007-08-31 13:47:21:000	370	192	
14	反应罐温度	2007-08-31 13:42:21:000	128.828705	192	
15	反应罐温度	2007-08-31 13:43:21:000	150.237457	192	
16	反应罐温度	2007-08-31 13:44:21:000	168.722565	192	
17	反应罐温度	2007-08-31 13:45:21:000	161.605469	192	
18	反应罐温度	2007-08-31 13:46:21:000	151.546783	192	
19	反应罐温度	2007-08-31 13:47:21:000	128.828705	192	

图 25.11 逐条显示变量数据

- **纵列显示不同变量数据：**每个变量占一列显示所有数据。
- **报表起始行号、报表起始列标：**在选择“纵列显示不同变量数据”时定义数据填充的起始行、列。查询参数做如下图所示配置，选择“纵列显示不同变量数据”。




点击“检索数据”按钮，执行后，数据填充到 sheet1 表中，如下图所示。

	A	B	C	D
1		原料罐液位	反应罐液位	反应罐温度
2	共有 6个数据			
3	时间	数据值	数据值	数据值
4	2007-08-31 13:42:21:000	265.00	370	128.8287048
5	2007-08-31 13:44:21:000	130.00	670	168.7225647
6	2007-08-31 13:46:21:000	260.00	530	151.5467834
7	2007-08-31 13:48:21:000	70.00	310	119.057312
8	2007-08-31 13:50:21:000	355.00	190	96.12069702
9	2007-08-31 13:52:21:000	265.00	310	119.057312
10				

图 25.12 逐列显示变量数据

- **是否输出数据个数：** 查询结束后，在表格中显示每个变量查询到的历史数据个数。
 - **退出时保存当前的配置信息：** 选中该项，当关闭参数配置对话框时，对话框上设置的参数将会被保存到“配置表”工作表中，方便使用。
-

 **注意：**

如果在查询时连续出现多次查询错误，一般情况下需要重新启动组态王的历史库服务程序。在控制面板中的“服务”中重新启动“HistorySrv（组态王历史库服务程序）”。该操作可能会影响组态王运行系统的历史数据存储，请谨慎操作。

使用 EXCEL 查询只能查询存储在本机上的历史库，远程的历史库暂时无法实现查询。

组态王提供的历史库 VBA 接口函数的详细示例代码请参见 Kingreport.xls 中的 VBA 编辑器中的内容。

第二十六章 API 接口及对象属性导入导出

- ☞ 介绍 xml 导出导入接口
- ☞ 介绍工程组态对象 xml 对象属性
- ☞ 介绍导入导出

26.1 xml 导出导入接口

XML 导出和导入接口是将组态王工程中的组态配置信息导出到 XML 文件中,或是从 XML 文件中组态对象信息导入到组态王工程中,下面介绍组态王 XML API 接口相关函数。

接口函数说明:

☆ `KV_RET KV_XML_API ConnectKvExplorer(IN KvString strUserName, IN KvString strPassword)`

此函数是连接组态王开发系统。

参数:

strUserName: 用户名, 组态王工程用户的名称

strPassword: 用户密码, 与用户相对应

返回值:

0: 成功

其他是负值, 完成校验功能后开放。

◇ KV_RET KV_XML_API DisconnectKvExplorer()

此函数是与组态王开发系统断开连接。

参数:

无

返回值:

0: 成功

其他是负值, 完成校验功能后开放。

◇ KV_RET KV_XML_API GetKvObjectNameList(IN KV_OPERATOR_OBJECT_TYPE opType, OUT std::vector<KvString> &strNameList)

此函数是从组态王工程中获取某种类型对象的名称列表。

参数:

opType: 对象类型

--0: 设备对象

--1: 变量对象

--2: 画面对象

- 3: 脚本对象
- 4: 非线性表对象
- 5: 记录表对象
- 6: 系统配置对象
- 7: 表格模板对象
- 8: 配方对象

StrNameList : 存储返回的该类型对象名称列表。

返回值:

0: 成功

其他是负值，完成校验功能后开放。

◇ **KV_RET KV_XML_API Export (IN KV_OPERATOR_OBJECT_TYPE opType, IN KvString strObjectName, IN KvString strXmlFilePathName)**

此函数是从组态王工程中导出某种类型对象到xml文件中。

参数:

opType: 对象类型

- 0: 设备对象
- 1: 变量对象
- 2: 画面对象
- 3: 脚本对象

--4: 非线性表对象

--5: 记录表对象

--6: 系统配置对象

--7: 表格模板对象

--8: 配方对象

返回值:

0: 成功

其他是负值, 完成校验功能后开放。

◇ `KV_RET KV_XML_API Import(IN KV_OPERATOR_OBJECT_TYPE opType, IN KvString strXmlFileFullPathName, IN KV_IMPORT_OPTION option)`

此函数是从xml文件中导入某种类型对象到组态王工程中。

参数:

opType: 对象类型

--0: 设备对象

--1: 变量对象

--2: 画面对象

--3: 脚本对象

--4: 非线性表对象

--5: 记录表对象

--6: 系统配置对象

--7: 表格模板对象

--8: 配方对象

strXmlFilePathName: 存储路径及xml文件名称

option: 导入处理方式。

--0: 有错误不导入

--1: 正确部分导入

--2: 全部正确合并导入

--3: 全部正确覆盖导入

返回值:

0: 成功

其他是负值, 完成校验功能后开放。

26.2 组态对象属性

26.2.1 设备

设备对象是指组态王开发系统中提供的各种类型的设备配置, 导出到 xml 文件中生成一组结构属性, 下面介绍各类设备配置导出到 xml 文件中对应的属性信息。

1、类型说明

设备对象类型	说明
逻辑设备 (LogicDeveice)	PLC、仪表及变频器设备
DDE设备 (DDE)	DDE连接方式的设备
板卡设备 (Borad)	板卡连接方式的设备
Opc设备 (OPC)	连接的OPC服务器
远程站点 (NET)	通过远程站点方式连接组态王服务器
虚拟设备 (DTU)	通信链路采用GPRS连接方式
串口参数 (COM)	逻辑设备串口通信时连接的串口链路
Modem参数 (ModemCOM)	通信链路采用Modem连接方式
虚拟串口参数 (KVCOM)	通信链路采用虚拟串口连接方式

2、设备属性

◇ 逻辑设备XML对象属性说明

属性	说明	属性类型 (组态王)
LogicName	设备名称	字符串 (参照设备名称定义)
DeviceName	驱动中设备名称	字符串 (参照驱动设备名称)
DeviceProducter	设备厂家	字符串, 参照驱动设备厂家名称
CommunicationMode	通信方式	字符串, 逻辑设备中默认为COM
DeviceAddress	设备地址	字符串, 参照驱动设备地址定义
CommunicationCard Name	通信口名称	字符串, 路基设备默认为COM*, 可选择不同串口
TryingInterval	尝试恢复间隔	整数, 5到2147484647秒
MaxRecoveryTime	最长恢复时间	整数, 0到2147484647小时

DynamicOptimization	是否动态优化	整数，0不优化，1优化
RedundancyType	冗余类型	整数，0不冗余，1冗余主，2冗余从
RedundancyDeviceName	冗余设备名称	字符串，参照驱动设备定义
VariableAutoCreated	自动创建变量	字符串，默认为Y, 用户无需更改设置

◇ DDE设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串（参照设备名称定义）
DeviceName	驱动中设备名称	字符串，默认为DDE
DeviceProducter	设备厂家	字符串，默认为DDE
ServerName	服务器名称	字符串，连接应用程序名
TopicName	话题名称	字符串，用户定义应用程序连接
DataChangeType	动态交换方式	整数，0高速块交换，1标准的windows项目交换

◇ 板卡设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串（参照设备名称定义）
DeviceName	驱动中设备名称	字符串，驱动中设备的名称
DeviceProducter	设备厂家	字符串，驱动的设备厂家
CommunicationMode	通信方式	字符串，默认为Board，不允许修改

DeviceAddress	设备地址	整数，参见板卡设备驱动说明
InitialCode	初始化学	字符串，参见板卡设备驱动说明
SingleOrDoubleEnded	单双端模式	整数，0单端，1双端
RedundancyType	冗余类型	整数，0不冗余，1冗余主，2冗余从
RedundancyDeviceName	冗余设备名称	字符串，参照设备定义
VariableAutoCreated	自动创建变量	字符串，默认为Y, 用户无需更改设置

◇ OPC服务器设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串（由IP地址和OPC服务器名称构成）
DeviceName	驱动中设备名称	字符串，OPC服务器名称
DeviceProducter	设备厂家	字符串，OPC服务器名称
CommunicationMode	通信模式	字符串，默认为OPC，不允许修改
CommunicationCardName	通信卡名称	字符串，OPC服务器的IP地址
OpcWay	OPC路径	字符串，不采用路径连接时，为空
TryingInterval	尝试恢复间隔	整数，5到2147484647秒
MaxRecoveryTime	最长恢复时间	整数，0到2147484647小时
DynamicOptimizati	是否动态优化	整数，0不优化，1动态化

on		
QueryFrequency	订阅频率	整数，55到30000秒
ReadType	读方式	整数，0异步读，1同步读
WriteType	写方式	整数，0异步写，1同步写

◇ 远程站点设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串（机器名称或ip地址）
DeviceName	驱动中设备名称	字符串，同上
BackNode	备份节点名称	字符串（机器名称或ip地址）
IsBackup	是否有备份节点	整数，0无，1有

◇ 虚拟设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串，默认为“虚拟设备”
DeviceName	驱动中设备名称	字符串，参见驱动的设备名称
DeviceProducter	设备厂家	字符串，参见驱动的设备厂家
DTUId	DTU序号	字符串，参见驱动的设置
CommunicationCard Name	通信卡名称	字符串，KVCOM*, 根据虚拟串口号设置
PortNumber	端口号	字符串，参见驱动的设置

◇ 串口属性说明

属性	说明	属性类型（组态王）
BaudRate	波特率	整数，波特率数值，如9600

DataBit	数据位	整数，数据位数，如8位
StopBit	停止位	整数，停止位数，如1位
TimeOut	超时时间	整数，200到32767
CommunicationType	通信方式	字符串，RS232, RS422, RS485

◇ MODEM设备属性说明

属性	说明	属性类型（组态王）
LogicName	设备名称	字符串，Modem_COM1
DeviceName	驱动中设备名称	字符串，IPModem不允许修改
CommunicationCard Name	通信卡名称	字符串，COM*, 根据串口号设置
PortNumber	端口号	字符串，参见驱动的设置
MajorModemATCW	主控制字	字符串，参见Modem设置
PriModemATCW	从控制字	字符串，参见Modem设置

3、设备导入导出

1、准备工作：

进入组态王工程浏览器，在设备管理视图下，创建多个类型的设备，比如：在 PLC 下创建一个莫迪康 Modbus 串口设备，如下图所示：

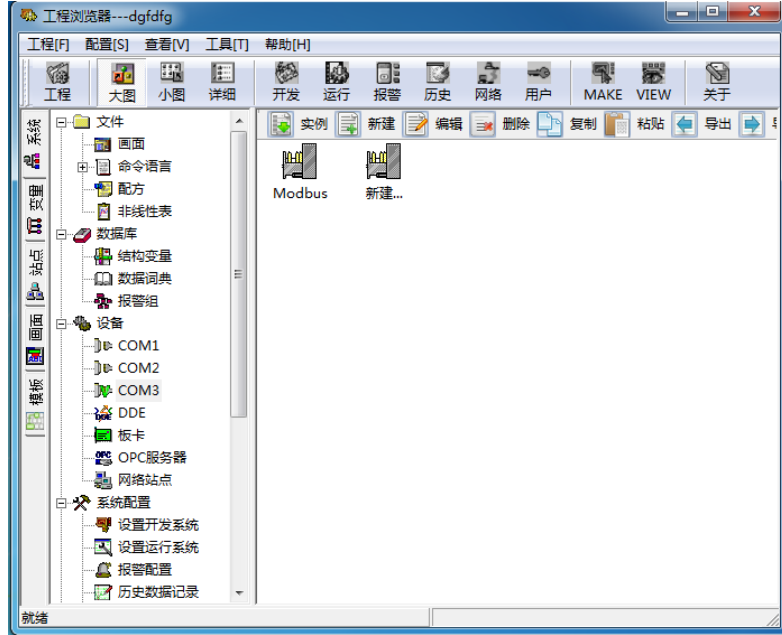


图 26.1 创建设备

2、导出为 XML 文件进行编辑或修改

选中新建的 Modbus 设备，右键选择“导出”（如果新建了多个设备，可选择“全部导出”）或者点击工具条上的“导出”按钮，弹出路径选择对话框，如下图所示：

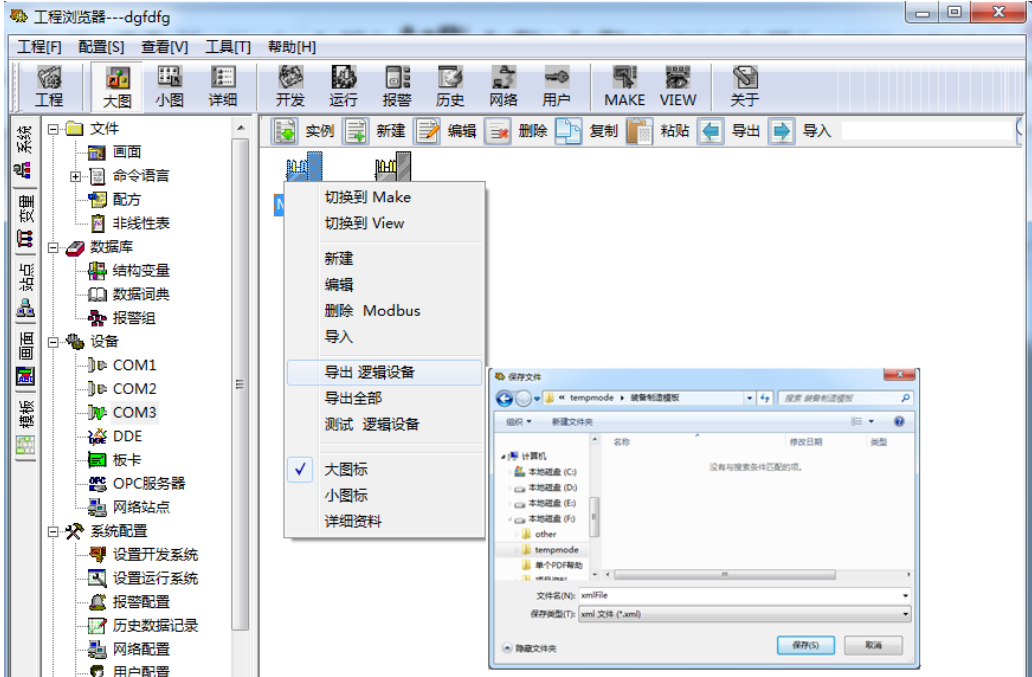


图 26.2 设备导出为 XML 文件

为导出的 XML 文件指定存放路径，如：F:\tempmode\装备制造模板，输入文件名：

Modbus。

到指定路径下查看导出后的 XML 描述信息，如下图所示：

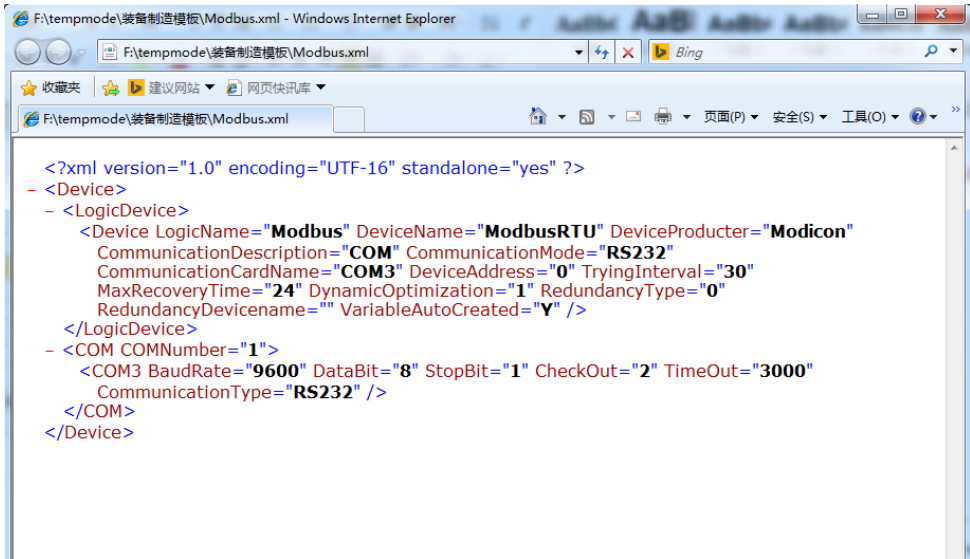


图 26.3 设备 XML 描述信息

- LogicName: 设备逻辑名称
- DeviceName: 驱动中的设备名称
- DeviceProducter: 设备厂家
- CommunicationDescription: 通讯描述
- CommuncationMode: 通讯方式
- CommunicationCardName: 通讯口名称
- DeviceAddress: 设备地址
- TryingInterval: 尝试恢复时间间隔
- MaxRecoveryTime: 最长恢复时间

- DynamicOptimization: 是否动态优化
- RedundancyType: 是否冗余
- RedundancyDevicename: 冗余设备名
- VariableAutoCreated: 是否自动创建变量
- BaudRate: 波特率
- DataBit: 数据位
- StopBit: 停止位
- TimeOut: 超时时间
- CommunicationType: 通讯方式
- Checkout: 奇偶校验
- ComNumber: Com 口数量

3、XML 文件导入到组态王工程

可以在 XML 文件中复制多个同类设备信息进行编辑, 或者在 XML 中集中修改各设备参数信息, 也可以利用自己开发的组态工具快速创建 IO 设备对象后存为 XML 文件, 最后导入到组态王工程, 导入的过程如下:

进入组态王工程浏览器, 在设备对应的内容显示区点击右键, 选择“导入”或者点击工具条的“导入”按钮, 弹出路径选择对话框, 选择仿真 PLC 设备对象对应的 XML 文件, 如下图所示:

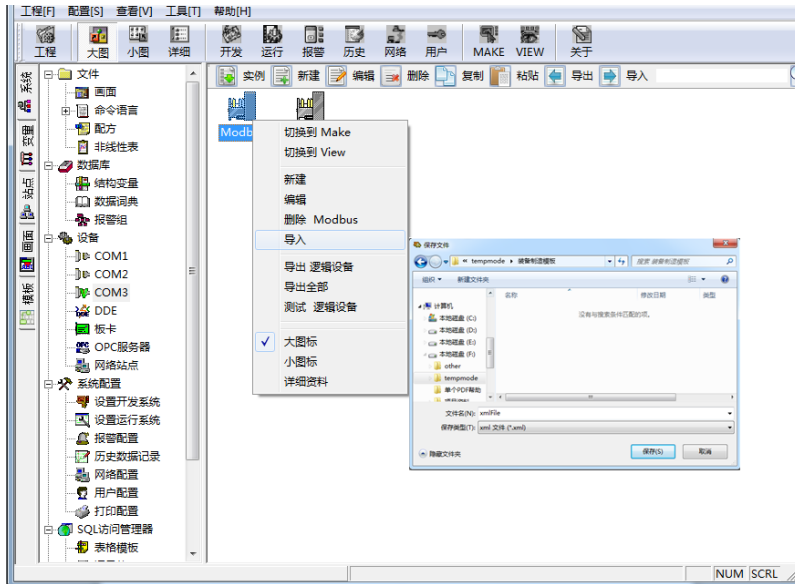


图 26.4 设备 XML 导入组态王工程

导入成功后即在组态王工程浏览器下看到新导入的设备了，如下图所示：

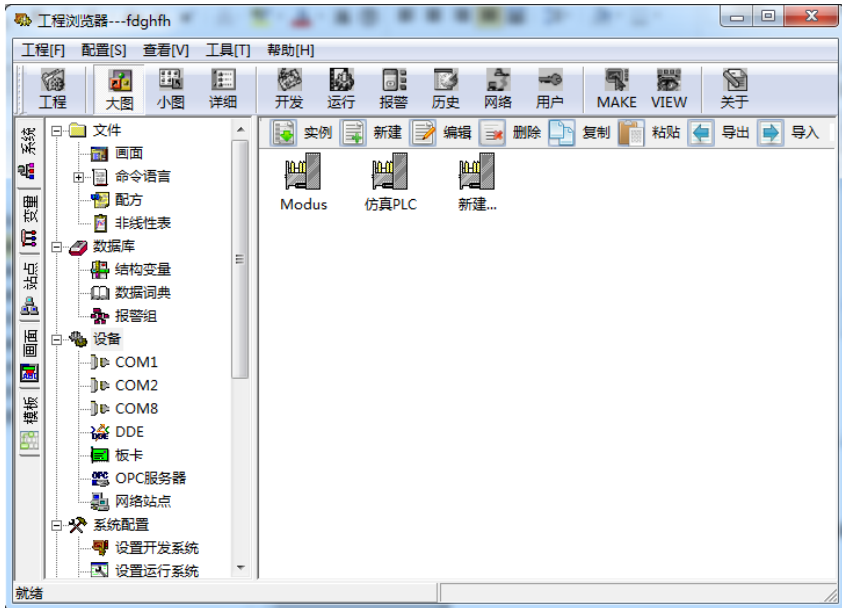


图 26.5 通过 XML 导入创建设备

26.2.2 变量

变量对象属性是指将数据词典中变量、结构模板及结构变量导出到 xml 文件中对应的数据对象。

1、变量 xml 节点属性

变量分组类型节点：

BaseTag：基本变量分支节点，包括内存基本变量和 I/O 基本变量

Struct：结构分支节点，包括结构模板及结构模板（Define 和 Member）

Define：结构模板分支节点

Member: 结构变量节点

数据类型分支节点:

MemInt: 内存整型变量节点

MemBool: 内存离散型变量节点

MemFloat: 内存浮点型变量节点

MemString: 内存字符串节点

IOInt: IO 整型变量节点

IOBool: IO 离散型变量节点

IOFloat: IO 浮点型变量节点

IOString: IO 字符串节点

2、变量属性

◇ 内存离散型变量 (MemBool) Tag 属性结构:

VarName: 变量名称, 如“memtag1”

TagType: 变量类型, 默认为“Bool”, 离散型不能修改

InitialValue: 变量初始值, “0”或“1”

SaveData: 保存数值, “N”或“Y”

SaveParameter: 保存参数, “N”或“Y”

AlarmGroup: 变量所属报警组, 如“RootNode”

Priority: 变量操作优先级, 如“1”

AlarmEnabled: 变量报警使能, “N”或“Y”

AlarmType: 报警类型, “0” 关报警, “1”开报警, “2”, 切换报警

OffAlarmText: 关闭报警文本, 默认为“关”

OnAlarmText: 打开报警文本, 默认为“开”

OnOffAlarmText: 从开切换到关报警文本, 默认为“开到关”

OffOnAlarmText: 从关切换到开报警文本, 默认为“关到开”

ExternField1: 扩展域 1 文本, 默认为“”, 空

ExternField2: 扩展域 2 文本, 默认为“”, 空

HistoryRecordWay: 历史存储方式, “0” 不记录, “1”变化记录, “2”定时记录, “3”
备份记录

RecordInterval: 历史记录时间间隔, “0”

ProduceOperateEvent: 产生操作事件, “N”或 “Y”

SecurityArea: 变量安全区, 如“无”

GroupPath: 变量组路径, 变量所在分组

Comment: 变量描述, 可为空“”

◇ 内存整型变量 (MemInt) Tag 属性结构:

VarName: 变量名称, 如“memtag2”

TagType: 变量类型, 内存整型默认为“Int”

ChangeSensitivity: 变化灵敏度

InitialValue: 变量初始值

MinValue: 变量最小值, 默认值为“0”

MaxValue: 变量最大值, 默认值为“999999999”

SaveData: 保存数据, “N”或 “Y”

SaveParameter: 保存参数, “N”或 “Y”

AlarmGroup: 报警组路径, 如“RootNode”

Priority: 优先级, 如“1”

LoLoEnabled: 低低限报警使能, “N”或 “Y”

LoLoLimit: 低低限报警限值, 默认为“0”

LoLoAlarmText: 低低限报警文本, 默认为“低低”

LoEnabled: 低限报警使能, “N”或 “Y”

LoLimit: 低限报警限值, 默认为“10”

LoAlarmText: 低限报警文本, 默认为“低”

HiEnabled: 高限报警“N”

HiLimit: 高限报警限值

HiAlarmText: 高限报警文本, 默认为“高”

HiHiEnabled: 高高限报警使能, “N”或 “Y”

HiHiLimit: 高高限报警限值

HiHiAlarmText: 高高限报警文本, 默认为“高高”

LimitDeadEnabled: 限值死区使能, “N”或 “Y”

LimitDeadZone: 限值死区值

AlarmDelayEnabled: 报警延时使能, "N"或 "Y"

AlarmDelay: 报警延时时间, 默认为"0"

ChangeRateEnabled: 变化率报警使能, "N"或"Y"

ChangeRate: 变量化率值 0-100

ChangeRateUnit: 变化率单位, "0" 秒, "1"分, "2"时

OffsetTargetValue: 偏差报警目标值

SmallOffsetEnabled: 小偏差报警使能, "N"或"Y"

SmallOffset: 小偏差值

BigOffsetEnabled: 大偏差报警使能, "N"或"Y"

BigOffset: 大偏差报警值

OffsetDeadZoneEnabled: 偏差死区报警使能, "N"或"Y"

OffsetDeadZone: 偏差死区值

ExternField1: 扩展域 1 文本, 默认为""空

ExternField2: 扩展域 2 文本, 默认为""空

HistoryRecordWay: 历史存储方式, "0" 不记录, "1"变化记录, "2"定时记录, "3"备份记录

RecordInterval: 历史记录时间间隔, "0"

ProduceOperateEvent: 产生操作事件, "N"或 "Y"

SecurityArea: 变量安全区, 如“无”

GroupPath: 变量组路径, 变量所在分组

Comment: 变量描述, 可为空”

◇ 内存实数型变量 (MemFloat) Tag 属性结构:

变量结构属性与内存整型相同。

◇ 内存字符串型变量 (MemString) Tag 属性结构:

VarName: 变量名称, 如“mentag4”

TagType: 变量类型, 默认为“String”

InitialValue: 初始值, 默认为“0.000000”

SaveData: 保存数值, “N”或“Y”

SaveParameter: 保存参数“N”或“Y”

SecurityArea: 安全区, 默认为“无”

GroupPath: 变量组路径

Comment: 变量描述

◇ IO 离散型变量 (MemBool) Tag 属性结构:

VarName: 变量名称, 如“I0tag1”

TagType: 变量类型, 默认为“I0Bool”

InitialValue: 变量初始值, “0”或“1”

SaveData: 保存数值, “N”或“Y”

SaveParameter: 保存参数, “N”或“Y”

DeviceName: 关联设备名称, 如“ModbusDevice”

RegisterName: 关联设备寄存器, 如“10001”

DataType: 关联寄存器类型, 如“Bit”

RWAttribute: 寄存器读写属性, 如“只读”

CollectFrequency: 采集频率, 如“1000”

DDEAccess: 是否允许 dde 访问, “N”或“Y”

AlarmGroup: 变量所属报警组, 如“RootNode”

Priority: 变量操作优先级, 如“1”

AlarmEnabled: 变量报警使能, “N”或“Y”

AlarmType: 报警类型, “0”关报警, “1”开报警, “2”, 切换报警

OffAlarmText: 关闭报警文本, 默认为“关”

OnAlarmText: 打开报警文本, 默认为“开”

OnOffAlarmText: 从开切换到关报警文本, 默认为“开到关”

OffOnAlarmText: 从关切换到开报警文本, 默认为“关到开”

ExternField1: 扩展域 1 文本, 默认为“”, 空

ExternField2: 扩展域 2 文本, 默认为“”, 空

HistoryRecordWay: 历史存储方式, “0”不记录, “1”变化记录, “2”定时记录, “3”备份记录

RecordInterval: 历史记录时间间隔, "0"

ProduceOperateEvent: 产生操作事件, "N"或 "Y"

SecurityArea: 变量安全区, 如"无"

GroupPath: 变量组路径, 变量所在分组

Comment: 变量描述, 可为空""

◇ IO 整型变量 (MemInt) Tag 属性结构:

VarName: 变量名称, 如"memtag2"

TagType: 变量类型, 内存整型默认为"Int"

ChangeSensitivity: 变化灵敏度

InitialValue: 变量初始值

MinValue: 变量最小值, 默认值为"0"

MaxValue: 变量最大值, 默认值为"999999999"

MinOriginalValue="0"

MaxOriginalValue="999999999"

SaveData: 保存数据, "N"或 "Y"

SaveParameter: 保存参数, "N"或 "Y"

DeviceName="ModbusDevice"

RegisterName="40001"

DataType="SHORT"

RWAttribute="读写"

CollectFrequency="1000"

CovertWay: 变换方式, "1"线性, "2"开方

AdvanceConvertWay: 高级变换方式, "0"无变换, "1"查表, "2"累计

NonLinearTableName: 非线性表名称

AddUpWay: 累计方式, "0"差值累计, "1"直接累计

MinAccumulateValue: 最小累计值

MaxAccumulateValue: 最大累计值

DDEAccess: 是否允许 DDE 访问, "N"或 "Y"

AlarmGroup: 报警组路径, 如"RootNode"

Priority: 优先级, 如"1"

LoLoEnabled: 低低限报警使能, "N"或 "Y"

LoLoLimit: 低低限报警限值, 默认为"0"

LoLoAlarmText: 低低限报警文本, 默认为"低低"

LoEnabled: 低限报警使能, "N"或 "Y"

LoLimit: 低限报警限值, 默认为"10"

LoAlarmText: 低限报警文本, 默认为"低"

HiEnabled: 高限报警"N"

HiLimit: 高限报警限值

HiAlarmText: 高限报警文本, 默认为“高”

HiHiEnabled: 高高限报警使能, “N”或“Y”

HiHiLimit: 高高限报警限值

HiHiAlarmText: 高高限报警文本, 默认为“高高”

LimitDeadEnabled: 限值死区使能, “N”或“Y”

LimitDeadZone: 限值死区值

AlarmDelayEnabled: 报警延时使能, “N”或“Y”

AlarmDelay: 报警延时时间, 默认为“0”

ChangeRateEnabled: 变化率报警使能, “N”或“Y”

ChangeRate: 变量化率值 0-100

ChangeRateUnit: 变化率单位, “0”秒, “1”分, “2”时

OffsetTargetValue: 偏差报警目标值

SmallOffsetEnabled: 小偏差报警使能, “N”或“Y”

SmallOffset: 小偏差值

BigOffsetEnabled: 大偏差报警使能, “N”或“Y”

BigOffset: 大偏差报警值

OffsetDeadZoneEnabled: 偏差死区报警使能, “N”或“Y”

OffsetDeadZone: 偏差死区值

ExternField1: 扩展域 1 文本, 默认为“”空

ExternField2: 扩展域 2 文本, 默认为""空

HistoryRecordWay: 历史存储方式, "0" 不记录, "1"变化记录, "2"定时记录, "3"备份记录

RecordInterval: 历史记录时间间隔, "0"

ProduceOperateEvent: 产生操作事件, "N"或 "Y"

SecurityArea: 变量安全区, 如"无"

GroupPath: 变量组路径, 变量所在分组

Comment: 变量描述, 可为空""

◇ IO 实数型变量 (IOFloat) Tag 属性结构:

变量结构属性与 IO 整型相同。

◇ IO 字符串型变量 (IOString) Tag 属性结构:

VarName: 变量名称, 如"memtag4"

TagType: 变量类型, 默认为"String"

InitialValue: 初始值, 默认为"0.000000"

SaveData: 保存数值, "N"或"Y"

SaveParameter: 保存参数"N"或"Y"

DeviceName: 关联设备名称, 如"SimPLC"

RegisterName: 关联寄存器名称, 如"STRING1"

DataType: 关联数据类型, "String"

RWAttribute: 寄存器读写属性, 如“读写”

CollectFrequency: 采集频率, 如“1000”

DDEAccess: 是否允许 DDE 访问, “N”或“Y”

SecurityArea: 安全区, 默认为“无”

GroupPath: 变量组路径

Comment: 变量描述

◇ 结构模板定义 (Struct) 结构:

结构分支属性 Name: 结构模板名称, 如“StructTag”

结构成员分支属性与基本变量相同 (参见基本变量结构属性)

◇ 结构变量 (Member) 结构:

结构变量基本属性

Name: 结构变量名称, 如“struct1”

TemplateName: 使用结构模板名称, 如“StructTag”

GroupPath: 变量所在分组

Comment: 结构变量描述

结构变量成员属性与基本变量相同 (参见基本变量结构属性)

3、变量导入导出

变量对象属性信息可以选中多个或全部导出到 XML 格式的文件, 也可以从 XML 格式的文件单独或全部导入到组态王工程, 具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

26.2.3 非线性表

1、非线性表属性

- ✧ 非线性表基本分支 Table 属性

TableName: 表名称, 如"ff"

Dot: 小数位数, 如"6"

- ✧ 非线性表数据分支 Value 属性

OrinalValue: 原始值, 如"12"

ProjectValue: 工程值, 如"22"



例如:

```
<NoLineTable>
```

```
  <Table TableName="ff" Dot="6">
```

```
    <Value OrinalValue="12" ProjectValue="22"/>
```

```
  </Table>
```

```
  <Table TableName="f2" Dot="6">
```

```
    <Value OrinalValue="23" ProjectValue="56"/>
```

```
  </Table>
```

```
</NoLineTable>
```

2、非线性表导入导出

非线性表可以单独或全部导出到 XML 格式的文件, 也可以从 XML 格式的文件单独或全

部导入到组态王工程，具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

26.2.4 配方

1、配方属性

配方属性包括配方变量和配方参数，可以将多个配方文件导出到一个 xml 文件中。

配方变量分支属性 (VarList)

VarName: 变量名称，如“\\本站点\memtag2”

VarType: 变量类型，如“整型”

配方参数分支属性 (RecipePara)

Name: 配方名称，如“工艺三”

配方数据分支属性 (ParaValue)

Value: 配方数据，如“23”



例如：

```
<RecipeList >
```

```
  <Recipe Name="新配方">
```

```
    <VarList>
```

```
      <Var VarName="\\本站点\memtag2" VarType="整型"/>
```

```
      <Var VarName="\\本站点\memtag3" VarType="实型"/>
```

```
    </VarList>
```

```
  </Para>
```

```
<RecipePara Name="工艺一">
    <ParaValue Value="12"/>
    <ParaValue Value="22"/>
</RecipePara>
<RecipePara Name="工艺二">
    <ParaValue Value="45"/>
    <ParaValue Value="32"/>
</RecipePara>
<RecipePara Name="工艺三">
    <ParaValue Value="23"/>
    <ParaValue Value="45"/>
</RecipePara>
</Para>
</Recipe>
</RecipeList>
```

2、配方导入导出

配方配置信息可以单独或全部导出到 XML 格式的文件，也可以从 XML 格式的文件单独或全部导入到组态王工程，具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

26.2.5 系统配置对象

系统配置对象属性

1、开发系统配置属性

开发系统设置节点 (ExploreSys) 属性

BShowDir: 是否显示工程路径, "0"显示, "1"不显示

Title: 开发系统名称, 默认为"开发系统"



例如:

```
<SystemConfig>
```

```
<ExploreSys BShowDir="0" Title="开发系统"/></SystemConfig>
```

2、运行系统配置属性

1) 运行系统外观设置节点 (View) 属性

StartState: view 运行时状态, "1" 缩略成图标, "0"最大化

Title: 运行系统标题, 如"运行系统"

MemuPic: 运行窗口是否有窗口菜单, "1"是, "0"否

MaxBox: 运行窗口是否显示最大化按钮, "1"是, "0"否

MinBox: 运行窗口是否显示最小化按钮, "1"是, "0"否

IsDir: 标题条是否显示工程路径, "1"是, "0"否

IsSize: 运行窗口边框是否可变大小, "1"是, "0"否

IsPicMenu: 是否显示画面菜单, "1"是, "0"否

IsSpecMenu: 是否显示特殊菜单, "1"是, "0"否

IsTestMenu: 是否显示调试菜单, "1"是, "0"否

IsNavMenu: 是否显示导航菜单, “1”是, “0”否

IsAboutMenu: 是否显示关于菜单

2) 运行系统主画面设置节点 (MainPic) 属性

pic1: 主画面 1 画面名称

pic2: 主画面 2 画面名称

pic3: 主画面 3 画面名称

pic4: 主画面 4 画面名称

pic5: 主画面 5 画面名称

pic6: 主画面 6 画面名称

pic7: 主画面 7 画面名称

pic8: 主画面 8 画面名称

pic9: 主画面 9 画面名称

pic10: 主画面 10 画面名称

3) 运行系统特殊功能设置节点 (SpecConfig) 属性

BaseTime: 基准频率, 默认为“500”

VariableTime: 变量刷新频率, 默认为“1000”

TagDisplay=“0”

NoAlt: 是否禁用 Alt 键, “0” 不禁用, “1”禁用

NoClose: 是否禁用“0”

VirKeyboard: 是否使用虚拟键盘, "1"是, "1"否

NoSwitch: 是否禁止任务切换, "1"是, "1"否

BeepInPut: 是否触敏对象有声音, "1"是, "1"否

MultiScreen: 是否多屏显示, "1"是, "1"否

WriteTagChanged: 是否变量变化下写数据, "1"是, "1"否

InitTagOnWrite: 是否初始运行下写数据, "1"是, "1"否

AlarmAckDef: 是否报警状态默认为已确认, "1"是, "1"否

PacketModeDef: 结构变量采用专用打包, "1"是, "1"否



例如:

```
<SystemConfig>
```

```
  <RunSysConfig>
```

```
    <View StartState="1" Title="运行系统" MemuPic="1"
```

```
MaxBox="1" MinBox="1" IsDir="0" IsSize="1" IsPicMenu="1" IsSpecMenu="1"
```

```
IsTestMenu="1" IsNavMenu="0" IsAboutMenu="1"/>
```

```
    <MainPic pic1="" pic2="" pic3="" pic4="" pic5="" pic6=""
```

```
pic7="" pic8="" pic9="" pic10="" />
```

```
    <SpecConfig BaseTime="500" VariableTime="1000"
```

```
TagDisplay="0" NoAlt="0" NoClose="0" VirKeyboard="1" NoSwitch="0" BeepInPut="1"
```

```
MultiScreen="0" WriteTagChanged="1" InitTagOnWrite="0" AlarmAckDef="0"
```

```
PacketModeDef="0"/>
```

```
</RunSysConfig></SystemConfig>
```

3、报警配置属性

1) 文件配置节点 (SaveFile) 属性

BufferSize: 报警缓存区大小, 默认为"100"

IsProjDir: 是否为指定路径, "1" 指定路径, "0" 默认工程路径

FilePath: 指定存储路径, 如="c:\program files\kingview\example\kingdemo1\"

RecordTime: 文件记录时间, 默认为"24"

StartTime: 开始记录时刻, 默认为"8"

SaveTime: 文件保存天数, 默认为"10"

GroupName: 报警组, 如"RootNode"

Priority: 报警优先级, 如"1"

BAlarm: 是否记录报警事件, "1"记录, "0"记录

BEnter: 是否记录登陆事件, "1"记录, "0"不记录

BOperate: 是否记录操作事件, "1"记录, "0"不记录

BStation: 是否记录工作站事件, "1"记录, "0"不记录

2) 数据库配置节点 (SaveDB) 属性

BUsrDefDataSource: 是否使用默认数据源, "1"是, "0"否

UsrName: 数据库登陆用户名称

Password: 数据库密码

DataSource: 数据源名称

DataSourceType: 数据源类型, "0" 系统数据源, "1" 用户数据源

GroupName: 报警组, 如"RootNode"

Priority: 优先级, 如"1"

BAlarm: 是否记录报警事件, "1" 记录, "0"不记录

BEnter: 是否记录登陆事件, "1" 记录, "0"不记录

BOperate: 是否记录操作事件, "1" 记录, "0"不记录

BStation: 是否记录工作站事件, "1" 记录, "0"不记录

3) 打印配置节点 (PrintConfig) 属性

PrintName: 打印机名称, 如"LPT1"

GroupName: 报警组, 如"RootNode"

Priority: 优先级, 如"1"

BAlarm: 是否记录报警事件, "1" 记录, "0"不记录

BEnter: 是否记录登陆事件, "1" 记录, "0"不记录

BOperate: 是否记录操作事件, "1" 记录, "0"不记录

BStation: 是否记录工作站事件, "1" 记录, "0"不记录

4) 记录报警事件分支节点 (AlarmEvent) 属性

BMonthSave: 是否按月保存, "1" 记录, "0"不记录 (存储 DB 时包含属性)

BDateTimeType: 是否按日期保存, "1" 记录, "0"不记录 (存储 DB 时包含属性)

BSave: 是否保存毫秒字段, "1" 记录, "0"不记录 (存储 DB 时包含属性)

BAlarmDate: 是否记录报警日期, "1" 记录, "0"不记录

AlarmDateFormat: 报警日期格式, "0"到 "5"六种日期格式

BAlarmTime: 是否记录报警时间, "1" 记录, "0"不记录

BAlarmHour: 是否记录小时, "1" 记录, "0"不记录

BAlarmMinute: 是否记录分钟, "1" 记录, "0"不记录

BAlarmSecond: 是否记录秒, "1" 记录, "0"不记录

BAlarmMilliSecond: 是否记录毫秒, "1" 记录, "0"不记录

BEventDate=: 是否记录事件日期, "1" 记录, "0"不记录

EventFormat: 事件日期格式, "0"到 "5"六种日期格式

BEventTime=: 是否记录事件时间, "1" 记录, "0"不记录

BAlarmHour: 是否记录小时, "1" 记录, "0"不记录

BAlarmMinute: 是否记录分钟, "1" 记录, "0"不记录

BAlarmSecond: 是否记录秒, "1" 记录, "0"不记录

BAlarmMilliSecond: 是否记录毫秒, "1" 记录, "0"不记录"

BVarName: 是否记录变量名称, "1" 记录, "0"不记录

VarNameSize: 变量名大小

BGroupName: 是否记录报警组名称, "1" 记录, "0"不记录

GroupNameSize: 报警组名长度

BAlarmValue: 是否记录报警值, "1" 记录, "0" 不记录

AlarmValueSize, 报警值长度

BLimitValue: 是否记录限值, "1" 记录, "0" 不记录

LimitValueSize: 限值长度

BVarNameComment: 是否记录变量描述

VarNameCommentSize: 变量描述长度

BAlarmType: 是否记录报警类型

AlarmTypeSize: 报警类型长度

BMachineName: 是否记录机器名

MachineNameSize: 机器名长度

BEventType: 是否记录事件类型

EventTypeSize: 事件类型长度

BResumeValue: 是否记录恢复值

ResumeValueSize: 恢复值长度

BOperatorName: 是否记录操作员

OperatorNameSize: 操作员长度

BIOServerName: 是否记录 IO 服务器

IOServerNameSize: IO 服务器长度

BQuality: 是否记录质量戳

BPriority: 质量戳长度

5) 记录操作事件分支节点 (OperateEvent) 属性

BEventDate=: 是否记录事件日期, "1" 记录, "0"不记录

EventFormat: 事件日期格式, "0"到 "5"六种日期格式

BEventTime=: 是否记录事件时间, "1" 记录, "0"不记录

BAlarmHour: 是否记录小时, "1" 记录, "0"不记录

BAlarmMinute: 是否记录分钟, "1" 记录, "0"不记录

BAlarmSecond: 是否记录秒, "1" 记录, "0"不记录

BEventType: 是否记录事件类型, "1" 记录, "0"不记录

EventTypeSize: 事件类型长度

BSuccess: 是否记录成功标志, "1" 记录, "0"不记录

SuccessSize: 成功标志长度

BVarName: 是否记录变量名称, "1" 记录, "0"不记录

VarNameSize: 变量名称长度

BVarComment: 是否记录变量描述, "1" 记录, "0"不记录

VarCommentSize: 变量描述长度

BMachineName: 是否记录机器名, "1" 记录, "0"不记录

MachineNameSize: 机器名长度

BOperateName: 是否记录操作员名, "1" 记录, "0"不记录

OperateNameSize: 操作员名长度

BNewValue: 是否记录新数值, "1" 记录, "0"不记录

NewValueSize=: 新数值长度

BFieldName: 是否记录域名, "1" 记录, "0"不记录

FieldNameSize: 域名长度

BOldValue: 是否记录旧值, "1" 记录, "0"不记录

OldValueSize: 旧值长度

BIOServerName: 是否记录 IO 服务器名称, "1" 记录, "0"不记录

IOServerNameSize: IO 服务器名称长度

6) 记录登录事件分支节点 (LoginEvent) 属性

BLoginDate: 是否记录登陆日期, "1" 记录, "0"不记录

LoginFormat: 登陆日期格式, "0"-"5"六种日期格式

BLoginTime: 是否记录登陆时间, "1" 记录, "0"不记录

BLoginHour: 是否记录登陆小时, "1" 记录, "0"不记录

BLoginMinute: 是否记录登陆分钟, "1" 记录, "0"不记录

BLoginSecond: 是否记录登陆秒, "1" 记录, "0"不记录

BLoginType: 是否记录登陆类型, "1" 记录, "0"不记录

LoginTypeSize: 登陆类型长度

BSuccess: 是否记录成功标志, "1" 记录, "0"不记录

SuccessSize: 成功标志长度

BMachineName: 是否记录机器名, "1" 记录, "0"不记录

MachineNameSize: 机器名长度

BOperateName: 是否记录操作员, "1" 记录, "0"不记录

OperateNameSize: 操作员长度

7) 记录工作站事件分支节点 (StationEvent) 属性

BEventDate: 是否记录事件日期, "1" 记录, "0"不记录

EventFormat: 事件日期格式, "0"-"5" 六种日期格式

BEventTime: 是否记录事件时间, "1" 记录, "0"不记录

BEventHour: 是否记录事件小时, "1" 记录, "0"不记录

BEventMinute: 是否记录事件分钟, "1" 记录, "0"不记录

BEventSecond: 是否记录事件秒, "1" 记录, "0"不记录

BEventType: 是否记录事件类型, "1" 记录, "0"不记录

EventTypeSize: 事件类型长度

BMachineName: 是否记录事件机器名, "1" 记录, "0"不记录

MachineNameSize: 机器名长度

4、历史数据记录属性

1) 历史记录节点 (HisConfig) 属性

IsHisSave: 是否历史记录, "1" 记录, "0"不记录

2) 历史库记录分支节点 (HisDB) 属性

IsHisDB: 是否保存历史库, "1" 记录, "0"不记录

SaveDay: 保存天数, 如"864000"为秒

AlarmDiskSize: 历史空间报警大小 M, 如"500"

Isdefaltdir: 是否默认存储路径, "0"是, "0"否

HisDBdir: 历史存储路径

3) 工业库记录节点 (KH) 属性

IsHisSave: 是否历史记录, "1" 记录, "0"不记录

IsKH: 是否存储到工业库, "1" 记录, "0"不记录

Server: 服务器 IP 地址

Port: 端口号, 如"5678"

User: 用户名称, 如"sa"

Secret: 密码, 如"sa"

TimeOut: 超时时间, 默认为"0"

4) 工业库服务器列表节点 (KHServerList) 属性

Server: 服务器 IP 地址

Port: 端口号, 如"5678"

User: 用户名称, 如"sa"

Secret: 密码, 如"sa"

TimeOut: 超时时间, 默认为"0"



例如：

```
<SystemConfig>
  <HisConfig IsHisSave="1">
    <HisDB IsHisDB="1" SaveDay="864000" AlarmDiskSize="500"
Isdefaltdir="0" HisDBdir=""/>
    <KH IsKH="1" User="sa" Secret="sa" TimeOut="0"/>
    <KHServerList Server="127.0.0.2" Port="5678" User="sa"
Secret="sa" TimeOut="1000">
      <KHServer/>
      <KHServer/>
    </KHServerList>
  </HisConfig></SystemConfig>
```

5、网络配置属性

1) 网络配置网络参数节点 (NetPara) 属性

BSingle: 是否为单机, "0"单机, "1"网络

WebServer: 本站点类型, "0" 为主机, "1"从机

RemoteRedunAdd: 从站点名称

RedunRemoteAdd: 从站点备份网卡

RedunRemoteRedunAdd: 从站点历史路径

MasterCheckInterval: 冗余心跳检测时间

LocalName: 本站点名, 如“LR4PJDF93DBDOLH”

NetKeyServer: 备份网卡

NetPacketSize: 包大小

NetPollFreq: 主机等待从机连接时间

NetKeepAliveFreq: 心跳检测时间

NetCallTimeout: 失效尝试恢复时间

NetSendTimeout: 心跳检测次数

2) 网络配置节点类型 (NetPara) 属性

ClockServer: 是否为校时服务器, “0”否, “1”是

ClockAdjustFreq: 校时时间间隔, 默认“1800”

IsBackHis: 是否为历史备份服务器, “0”否, “1”是

IsLogonServer: 是否为登陆服务器, “0”否, “1”是

IsIoServer: 是否为 IO 服务器, “0”否, “1”是

AlarmServer: 是否有报警服务器, “0”否, “1”是

AlarmSerList: 报警服务器列表

HistoryServer: 是否有历史服务器, “0”否, “1”是

HisSerList: 历史服务器列表

3) 网络配置客户端配置 (Client) 属性

IsClient: 本机是否为客户端, "0"否, "1"是

IoSerList: IO 服务器列表

AlarmSerList: 报警服务器列表

HisServerList: 历史服务器列表



例如:

```
<SystemConfig>
```

```
  <NetConfig>
```

```
    <NetPara BSingle="0" ClockServer="0" ClockAdjustFreq="1800"
```

```
WebServer="0" RemoteRedunAdd="" RedunRemoteAdd="" RedunRemoteRedunAdd=""
```

```
CommucationPort="2001" REDUN_Port="2002" REDUN_timelapse="0"
```

```
MasterCheckInterval="5" LocalName="LR4PJDF93DBD0LH" NetKeyServer=""
```

```
NetPacketSize="8192" NetPollFreq="20" NetKeepAliveFreq="10"
```

```
NetCallTimeout="10" NetSendTimeout="3"/>
```

```
  <Node IsBackHis="1" IsLogonServer="1" IsIoServer="" AlarmServer="0"
```

```
AlarmSerList="" HistoryServer="0" HisSerList=""/>
```

```
  <Client IsClient="0" IoSerList="" AlarmSerList="" HisServerList=""/>
```

```
</NetConfig>
```

```
</SystemConfig>
```

6、用户配置属性

1) 用户组分支节点 (UserGroup-Group) 属性

GroupName: 用户组名称“系统管理员组”

GroupCommet: 用户组描述, 如“系统管理员”

GroupSection: 组安全区, 如“A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S”

2) 用户分支节点 (UserList-User) 属性

UserName: 用户名称, 如“系统管理员”

UserGroup: 用户组名称, 如“系统管理员组”

Secret: 密码, 如“kingview”

LonginTimeOut: 登陆超时时间, 如“255”

UserCommet: 用户描述, 默认为空”

Priority: 操作优先级, 如“999”

GroupSection: 安全区, 如“A, B, C, D, E, F, G, H, I, J, K, L, M, N”

3) 安全区分支节点 (SectionList) 属性

SectionList:安全区属性, 包括全部安全区名称, “无,

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD, EE, FF, G

G, HH, II, JJ, KK, LL, MM, NN, OO, PP, QQ, RR, SS, TT, UU, VV, WW, XX, YY, ZZ, AAA, BBB, CCC,

DDD, EEE, FFF, GGG, HHH, III, JJJ, KKK”



例如:

<SystemConfig>

```
<UserConfig>
  <UserGroup>
    <Group GroupName="系统管理员组" GroupCommet="系统管理员组"
GroupSection="A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD
, EE, FF, GG, HH, II, JJ, KK, LL, MM, NN, OO, PP, QQ, RR, SS, TT, UU, VV, WW, XX, YY, ZZ, AAA, BBB, CC
C, DDD, EEE, FFF, GGG, HHH, III, JJJ, KKK"/>
    <Group GroupName="无组" GroupCommet="无组" GroupSection="无"/>
  </UserGroup>
  <UserList>
    <User UserName="系统管理员" UserGroup="系统管理员组"
Secret="kingview" LonginTimeOut="255" UserCommet="" Priority="999"
GroupSection="A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD
, EE, FF, GG, HH, II, JJ, KK, LL, MM, NN, OO, PP, QQ, RR, SS, TT, UU, VV, WW, XX, YY, ZZ, AAA, BBB, CC
C, DDD, EEE, FFF, GGG, HHH, III, JJJ, KKK"/>
    <User UserName="无" UserGroup="无组" Secret="" LonginTimeOut="0"
UserCommet="" Priority="0" GroupSection="无"/>
    <User UserName="admin" UserGroup="无组" Secret="123"
LonginTimeOut="0" UserCommet="" Priority="999" GroupSection="A"/>
  </UserList>
```

```
<Section SectionList="
```

```
无, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD, EE, FF, GG, H  
H, II, JJ, KK, LL, MM, NN, OO, PP, QQ, RR, SS, TT, UU, VV, WW, XX, YY, ZZ, AAA, BBB, CCC, DDD, EEE, F  
FF, GGG, HHH, III, JJJ, KKK">
```

```
</Section>
```

```
</UserConfig></SystemConfig>
```

7、打印配置属性

1) 打印机节点 (PrintConfig) 属性

PicPrinter: 画面打印机名称, 如"LPT1"

RealAlarmPrinter: 实时报警打印机, 如"LPT1"

ReportPrinter: 报表打印机, 如"LPT1"



例如:

```
<SystemConfig>
```

```
<PrintConfig PicPrinter="LPT1" RealAlarmPrinter="LPT1" ReportPrinter="LPT1"/>
```

```
</SystemConfig>
```

8、系统配置导入导出

系统配置信息可以单独或全部导出到 XML 格式的文件, 也可以从 XML 格式的文件单独或全部导入到组态王工程。导入导出的具体做法参见设备的导入导出。

26.2.6 命令语言对象

1、命令语言属性

1) 程序命令语言

应用程序命令语言节点 (AppCommandScript) 属性

IntervalTime: 运行周期, 如“3000”毫秒

分支节点 (Script) 属性

Name: 命令运行执行方式, “启动时”, “运行时”, “停止时”

<![CDATA[

]]>: 代码程序段, 如下

<![CDATA[

\\本站点\tag=1;

]]>



例如:

```
<AppCommandScript IntervalTime="3000">
```

```
  <Script Name="启动时">
```

```
    <![CDATA[
```

```
]]>
```

```
  </Script>
```

```
  <Script Name="运行时">
```

```
    <![CDATA[
```

```
\\本站点\tag=1;
```

```
]]>
```

```
</Script>
```

```
<Script Name="停止时">
```

```
<![CDATA[
```

```
]]>
```

```
</Script>
```

```
</AppCommandScript>
```

2) 数据改变命令语言属性

数据改变命令语言节点 (DataChangeScript) Script 属性

Name: 数据改变表达式, 如“\\本站点\tag”

```
<![CDATA[
```

```
]]>: 代码程序段, 如下
```

```
<![CDATA[
```

```
\\本站点\tag=1;
```

```
]]>
```



例如:

```
<DataChangeScript>
```

```
<Script Name="\\本站点\tag">
```

```
<![CDATA[
```

```
ClosePicture("tab");
```

```
]]>
```

```
</Script>
```

```
</DataChangeScript>
```

3) 事件命令语言属性

事件命令语言节点 (EventScript) Event 属性

condTime 节点: 事件存在周期执行脚本间隔, 如 3000

Comment: 事件描述表达式, CDATA 类型如

```
<![CDATA[
```

```
]]>
```

事件 script 分支节点属性

Name: 事件条件, 默认为“事件条件”, “存在时”, “发生时”, “消失时”

CDATA 数据段: 条件表达式, 如

```
<![CDATA[
```

```
\\本站点\tag==1
```

```
]]>
```



例如:

```
<EventScript>
```

```
<Event>
```

```
<condTime>3000</condTime>

<Comment>

    <![CDATA[

]]>

    </Comment>

    <Script Name="事件条件">

        <![CDATA[

\\本站点\tag==1

]]>

        </Script>

        <Script Name="发生时">

            <![CDATA[

\\本站点\tag=0;

]]>

            </Script>

            <Script Name="存在时">

                <![CDATA[

]]>

                </Script>
```

```
<Script Name="消失时">
  <![CDATA[
]]>
```

```
</Script>
</Event>
</EventScript>
```

4) 热键命令语言属性

事件命令语言节点 (HotKeyScript) Script 属性

Name: 热键组合, 如"Ctrl+Shift+B"

Section: 安全区, 如"0"

OptTouch: 优先级, 如"1"

CDATA 数据: 脚本程序, 如

```
<![CDATA[
  \\本站点\tag=1;
]]>
```



例如:

```
<HotKeyScript>
  <Script Name="Ctrl+Shift+B" Section="0" OptTouch="1">
    <![CDATA[
```



```
\\本站点\tag=1;
```

```
]]>
```

```
</Script>
```

```
</HotKeyScript>
```

5) 自定义函数命令语言属性

自定义函数节点 (UserDefinedFunction) Script 属性

ReturnType: 返回值类型, 如"VOID"

FunctionName: 函数名称, 如" \$System_RealAlarm"

ParaNum: 参数个数, 如"4"

Para*: 函数参数, 如"RealTag rTag"

CDATA 数据段, 脚本程序



例如:

```
<UserDefinedFunction>
```

```
<Script ReturnType="VOID" FunctionName=" $System_RealAlarm" ParaNum="4"
```

```
Para1="RealTag rTag" Para2=" long nTime" Para3=" long nEvent" Para4=" long
```

```
nAction">
```

```
<![CDATA[
```

```
//实型变量产生的事件。nAction == 1时表示双击报警条,
```

```
//nAction == 0 时表示报警或事件产生
```

```
    \\本站点\tag=2;  
  ]]>  
  
</Script>  
  
</UserDefinedFunction>
```

2、命令语言导入导出

命令语言可以单独或全部导出到 XML 格式的文件,也可以从 XML 格式的文件单独或全部导入到组态王工程。具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

26.2.7 关系库对象

1、关系表属性

1) 表格模板属性

表名称节点 (TableTemplate) 属性

TableName: 表格模板名称, 如"tab1"

字段分支节点 (Field) 属性

FieldName: 字段名称, 如"时间"

VarType: 字段数据类型, "0"布尔型, "1"整型, "2"浮点型, "3"布尔型,

"3" 定长字符串型, "4" 变长字符串型

FildSize: 字段长度, 如"20"

IndexType: 索引类型, "0" 有 (唯一), "1" 有 (不唯一), "2" 没有

AllowNull: 是否允许为空, "0" 不允许, "1" 允许



例如：

```
<SqlTableTemplate >
    <TableTemplate TableName="tbl1">
        <Field FieldName="时间" VarType="4" FieldSize="20" IndexType="2"
AllowNull="0"/>
        <Field FieldName="变量名" VarType="4" FieldSize="20" IndexType="2"
AllowNull="0"/>
        <Field FieldName="变量描述" VarType="3" FieldSize="20" IndexType="2"
AllowNull="0"/>
        <Field FieldName="变量值" VarType="1" FieldSize="1" IndexType="2"
AllowNull="0"/>
    </TableTemplate>
</SqlTableTemplate>
```

2) 记录体属性

记录体名称节点 (SqlRecord) 属性

RecordName: 记录体名称, 如"bind"

字段分支节点 (Field) 属性

FieldName: 字段名称, 如"时间"

VarName: 变量名称, 如"\\本站点\tag"



例如：

```
<SqlRecordList >  
    <SqlRecord RecordName="bind">  
        <Field FieldName="时间" VarName="\本站点\tag"/>  
        <Field FieldName="变量名" VarName="\本站点\tag1"/>  
        <Field FieldName="数据" VarName="\本站点\tag1"/>  
    </SqlRecord>  
</SqlRecordList>
```

2、关系库对象导入导出

表格模板和记录体可以单独或全部导出到 XML 格式的文件，也可以从 XML 格式的文件单独或全部导入到组态王工程。具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

26.2.8 图形对象

图形对象是指在图形画面上绘制的各种图形图素，主要包括对象名称、位置、大小、外观、数据连接、动画连接等属性信息。下面介绍各种图素存储到 xml 文件中节点属性。

1、基本图素属性

1) 直线 (Line) 类型节点

Name: 直线对象名称，如"Graph8"

Top: 图素左上顶点 x 坐标值，整数，如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth: 画笔宽度, 整数, 如"1"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 操作提示文本, 字符串, 如"罐子背景"



例如:

```
<Line Name="Graph8" Top="30" Left="20" Bottom="30" Right="130" PenWidth="1"
PenStyle="0" PenColor="#040404" BrushStyle="0" BrushColor="#FFFFFF"
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#FFFFFF"
BrushToColor="#000000" IsLock="0" Comment="">
```

</Line>

2)、椭圆 (Ellipse) 类型节点

Name: 图素名称, 如"Graph2"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth: 画笔宽度, 整数, 如"1"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 操作提示文本, 字符串, 如"运行开启"



例如:

```
<Ellipse Name="Graph2" Top="230" Left="40" Bottom="331" Right="211"  
PenWidth="1" PenStyle="0" PenColor="#040404" BrushStyle="0"  
BrushColor="#040404" BrushHatch="0" BrushFillStyle="0"  
BrushFromColor="#040404" BrushToColor="#040404" IsLock="0" Comment="">
```

3)、扇形 (Sector) 类型节点

Name: 图素名称, 如"Graph2"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth: 画笔宽度, 整数, 如"1"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 操作提示文本, 字符串, 如“运行开启”

StartPtX: 起始点 X 坐标, 如“98”

StartPtY: 起始点 Y 坐标, 如“113”

EndPtX: 结束点 X 坐标, 如“42”

EndPtY: 结束点 Y 坐标, 如“77”



例如:

```
<Sector Name="Graph9" Top="70" Left="30" Bottom="120" Right="110" PenWidth="1"
PenStyle="0" PenColor="#040404" BrushStyle="0" BrushColor="#040404"
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#040404"
BrushToColor="#040404" IsLock="0" Comment="" StartPtX="98" StartPtY="113"
EndPtX="42" EndPtY="77">
```

4)、矩形 (Rect) 类型节点

Name: 矩形图素对象的名称, 字符串, 如“Graph11”

Top: 图素左上顶点 x 坐标值, 整数, 如“20”

Left: 图素左上顶点 Y 坐标值, 整数, 如“160”

Bottom: 图素右下顶点 Y 坐标值, 整数, 如“70”

Right: 图素右下顶点 X 坐标值, 整数, 如“291”

PenWidth: 画笔宽度, 整数, 如“1”

PenStyle: 画笔样式, 整数, 如“0”

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 提示文本信息, 字符串, 如"罐子背景"

Width: 矩形图素圆角区宽度, 整数, 如"131"

Height: 矩形图素圆角区高度, 整数, 如"50"



例如:

```
<Rect Name="Graph11" Top="20" Left="160" Bottom="70" Right="291" PenWidth="1"  
PenStyle="0" PenColor="#040404" BrushStyle="0" BrushColor="#040404"  
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#040404"  
BrushToColor="#040404" IsLock="0" Comment="" Width="0" Height="0">
```

5)、折线 (BrokenLine) 类型节点

Name: 折线图素的名称, 字符串, 如"Graph12"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth: 画笔宽度, 整数, 如"1"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 操作提示文本, 字符串, 如"折线管道"

PointArr: 折线拐点坐标, 如"{{330, 30}, {470, 50}, {470, 50}, {360, 60}}"



例如:

```
<BrokenLine Name="Graph12" Top="30" Left="330" Bottom="60" Right="470"
```

```
PenWidth="1" PenStyle="0" PenColor="#040404" BrushStyle="1"
```

```
BrushColor="#040404" BrushHatch="0" BrushFillStyle="0"
```

BrushFromColor="#FFFFFF" BrushToColor="#000000" IsLock="0" Comment=""

PointArr="{{330, 30}, {470, 50}, {470, 50}, {360, 60}}">

6)、多边形 (Polygon) 类型节点

Name: 多边形图素对象名称, 字符串, 如"Graph14"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth: 画笔宽度, 整数, 如"1"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定,

Comment: 操作提示文本, 字符串, 如"罐子背景"

PointArr: 多边形拐点坐标序列值, 如"{{260, 100}, {320, 130}, {230, 130}, {230, 110}}"



例如:

```
<Polygon Name="Graph14" Top="100" Left="230" Bottom="130" Right="320"  
PenWidth="1" PenStyle="0" PenColor="#040404" BrushStyle="0"  
BrushColor="#040404" BrushHatch="0" BrushFillStyle="0"  
BrushFromColor="#040404" BrushToColor="#040404" IsLock="0" Comment=""  
PointArr="{{260, 100}, {320, 130}, {230, 130}, {230, 110}}">  
    </Polygon>
```

7)、管道 (Pipe) 类型节点

Name="Graph13"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定

Comment: 操作提示文本, 默认为空, 如""

PointArr: 管道拐点坐标值, 如"{{540, 30}, {700, 40}, {700, 40}, {590, 70}}"

PipeWidth: 管道宽度, 整数, 如"16"

FlowLineWidth: 流动线宽度, 整数, 如"12"

FlowLineLength: 流动线长度, 整数, 如"10"

PipeColor: 管道颜色, 颜色数值, 如"0"

FlowLineColor: 流动线颜色, 颜色数值, 如"65280"

PipeFlowState: 管道状态, 如"0"



例如:

```
<Pipe Name="Graph13" Top="30" Left="540" Bottom="70" Right="700"
PenWidth="1" PenStyle="0" PenColor="#040404" BrushStyle="1"
BrushColor="#040404" BrushHatch="0" BrushFillStyle="0"
BrushFromColor="#FFFFFF" BrushToColor="#000000" IsLock="0" Comment=""
PointArr="{{540, 30}, {700, 40}, {700, 40}, {590, 70}}" PipeWidth="16"
FlowLineWidth="12" FlowLineLength="10" PipeColor="0" FlowLineColor="65280"
PipeFlowState="0">
```

</Pipe>

8)、文本标签 (Label) 类型节点

Name: 文本标签对象名称, 如"Graph15"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenStyle: 画笔样式, 整数, 如"0"

PenColor: 画笔颜色, 颜色数值, 如"#040404"

BrushStyle: 画刷样式, 整数, 如"0"

BrushColor: 画刷颜色, 颜色数值, 如"#040404"

BrushHatch: 阴影画刷样式, 整数, 如"0"

BrushFillStyle: 画刷填充样式, 整数, 如"0"

BrushFromColor: 画刷过渡色起始颜色, 颜色数值, 如"#040404"

BrushToColor: 画刷过渡色结束颜色, 颜色数值, 如"#040404"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定

Comment: 操作提示文本, 默认为空, 如""

FontColor: 字体颜色, 颜色数值, 如"#040404"

FontStyle=: 字体样式, 如 0"

FontHeight: 字体高度, 如"16"

FontWidth: 字体宽度, 如"0"

FontIndex: 字体索引值, 如"16"

FontHeight="16"

FontEscapement="0"

FontOrientation="0"

FontWeight="0"

FontItalic="0"

FontUnderline="0"

FontStrikeOut="0"

FontCharSet="0"

FontOutPrecision="0"

FontClipPrecision="0"

FontQuality="0"

FontPitchAndFamily: 字符集, 如"22"

FontName: 文本字体, 如"宋体"

TextCont: 标签文本内容, 字符串, 如"dfaf"

Director: 文本方向, "0"横排, "1"竖排



例如:

```
<Label Name="Graph15" Top="487" Left="89" Bottom="503" Right="121" PenWidth="1"
```

```
PenStyle="0" PenColor="#000000" BrushStyle="0" BrushColor="#FFFFFF"
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#FFFFFF"
BrushToColor="#000000" IsLock="0" Comment="" FontColor="#040404"
FontStyle="0" FontHeight="16" FontWidth="0" FontIndex="16" FontHeight="-16"
FontEscapement="0" FontOrientation="0" FontWeight="0" FontItalic="0"
FontUnderline="0" FontStrikeOut="0" FontCharSet="0" FontOutPrecision1="0"
FontClipPrecision="0" FontQuality="0" FontPitchAndFamily="22" FontName="宋体"
TextCont="dfaf" Director="0">
    </Label>
```

9)、按钮 (Button) 类型节点

Name="Graph16"

Top: 图素左上顶点 x 坐标值, 整数, 如"20"

Left: 图素左上顶点 Y 坐标值, 整数, 如"160"

Bottom: 图素右下顶点 Y 坐标值, 整数, 如"70"

Right: 图素右下顶点 X 坐标值, 整数, 如"291"

PenWidth="1"

PenStyle: 画笔样式, 如"0"

PenColor: 画笔颜色, 如"#000000"

BrushStyle="0"

BrushColor="#FFFFFF"

BrushHatch="0"

BrushFillStyle="0"

BrushFromColor="#FFFFFF"

BrushToColor="#000000"

IsLock="0"

Comment=""

FontColor="#040404"

FontStyle="0"

FontHeight="16"

FontWidth="0"

FontIndex="16"

FontHeight="-16"

FontEscapement="0"

FontOrientation="0"

FontWeight="0"

FontItalic="0"

FontUnderline="0"

FontStrikeOut="0"

FontCharSet="0"

```
FontOutPrecision="0"  
FontClipPrecision="0"  
FontQuality="0"  
FontPitchAndFamily="22"  
FontName="宋体"  
TextCont="文本"  
Director="0"  
IsPress="0"  
LineType="0"  
LineStyle="0"  
SideNum="4"  
Tip=""  
BitType="0"
```



例如:

```
<Button Name="Graph16" Top="370" Left="50" Bottom="410" Right="150" PenWidth="1"  
PenStyle="0" PenColor="#000000" BrushStyle="0" BrushColor="#FFFFFF"  
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#FFFFFF"  
BrushToColor="#000000" IsLock="0" Comment="" FontColor="#040404"  
FontStyle="0" FontHeight="16" FontWidth="0" FontIndex="16" FontHeight="-16"
```

```
FontEscapement="0" FontOrientation="0" FontWeight="0" FontItalic="0"  
FontUnderline="0" FontStrikeOut="0" FontCharSet="0" FontOutPrecision1="0"  
FontClipPrecision="0" FontQuality="0" FontPitchAndFamily="22" FontName="宋体"  
TextCont="文本" Director="0" IsPress="0" LineType="0" LineStyle="0" SideNum="4"  
Tip="" BitType="0">  
  
    </Button>
```

10)、位图 (Bitmap) 类型节点

Name: 图素对象名称, 字符串, 如"Graph34"

Top: 图素左上顶点 Y 坐标值, 如"20"

Left: 图素左上顶点 X 坐标值, 如"750"

Bottom: 图素右下顶点 Y 坐标值, 如"130"

Right: 图素右下顶点 X 坐标值, 如"961"

PenWidth: 画笔宽度, 默认值为"1"

PenStyle: 画笔样式, 默认为"0"

PenColor: 画笔颜色, 颜色值, 默认为"#040404"

BrushStyle: 画刷样式, 默认为"1"

BrushColor: 画刷颜色, 颜色值, 默认为"#040404"

BrushHatch: 阴影画刷样式, 默认为"0"

BrushFillStyle: 画刷填充样式, 默认为"0"

BrushFromColor: 画刷过渡色开始色, 颜色值, 如"#FFFFFF"

BrushToColor: 画刷过渡色结束色, 颜色值, 如"#000000"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定

Comment: 图素操作提示文本, 字符串, 默认为空, 如""

BitmapType: 位图类型, 如"0"

FileName: 填充位图存储路径, ""



例如:

```
<Bitmap Name="Graph34" Top="20" Left="750" Bottom="130" Right="961" PenWidth="1"
PenStyle="0" PenColor="#040404" BrushStyle="1" BrushColor="#040404"
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#FFFFFF"
BrushToColor="#000000" IsLock="0" Comment="" BitmapType="0" FileName="">
</Bitmap>
```

11)、菜单 (Menu) 类型节点

Name: 菜单图素对象名称, 文本字符串, 如"Graph35"

Top: 图素绘制矩形区左上顶点 Y 坐标值, 如"370"

Left: 图素绘制矩形区左上顶点 X 坐标值, 如"270"

Bottom: 图素绘制矩形区右下顶点 Y 坐标值, 如"550"

Right: 图素绘制矩形区右下顶点 X 坐标值, 如"601"

PenWidth: 画笔宽度, 默认为"1"

PenStyle: 画笔样式, 默认为"0"

PenColor: 画笔颜色, 默认为"#000000"

BrushStyle: 画刷样式, 默认为"0"

BrushColor: 画刷颜色, 默认为"#FFFFFF"

BrushHatch: 阴影画刷样式, 默认为"0"

BrushFillStyle: 画刷填充样式, 默认为"0"

BrushFromColor: 画刷过渡色开始颜色, 如"#FFFFFF"

BrushToColor: 画刷过渡色结束颜色, 如"#000000"

IsLock: 图素位置是否锁定, "0"不锁定, "1"锁定

Comment: 图素操作提示文本, 默认为空, 如""

FontColor="#040404"

FontStyle="0"

FontHeight="16"

FontWidth="0"

FontIndex="16"

FontHeight="-16"

FontEscapement="0"

FontOrientation="0"

FontWeight="0"

FontItalic="0"

FontUnderline="0"

FontStrikeOut="0"

FontCharSet="0"

FontOutPrecision1="0"

FontClipPrecision="0"

FontQuality="0"

FontPitchAndFamily="22"

FontName="宋体"

TextCont="菜单"

MenuCont: 菜单列表表, 如"{{{111}}}"

<Link LinkType="CommandScript"

Express=""

Priority="1"

Tip=""

Key="无"

Section="1"

<DownLink LinkType="DownLink"><![CDATA[]]></DownLink>

<WhileLink LinkType="WhileLink"><![CDATA[]]></WhileLink>

<UpLink LinkType="UpLink"

```
Interval="0"><![CDATA[]]></UpLink>
```

```
    <MoveInLink
```

```
LinkType="MoveInLink"><![CDATA[]]></MoveInLink>
```

```
    <MoveOutLink
```

```
LinkType="MoveOutLink"><![CDATA[]]></MoveOutLink>
```

```
    </Link>
```

```
  </Menu>
```

```
<Menu Name="Graph35" Top="370" Left="270" Bottom="550" Right="601" PenWidth="1"
```

```
PenStyle="0" PenColor="#000000" BrushStyle="0" BrushColor="#FFFFFF"
```

```
BrushHatch="0" BrushFillStyle="0" BrushFromColor="#FFFFFF"
```

```
BrushToColor="#000000" IsLock="0" Comment="" FontColor="#040404"
```

```
FontStyle="0" FontHeight="16" FontWidth="0" FontIndex="16" FontHeight="-16"
```

```
FontEscapement="0" FontOrientation="0" FontWeight="0" FontItalic="0"
```

```
FontUnderline="0" FontStrikeOut="0" FontCharSet="0" FontOutPrecision1="0"
```

```
FontClipPrecision="0" FontQuality="0" FontPitchAndFamily="22" FontName="宋体"
```

```
TextCont="菜单" MenuCont="{{111}}">
```

```
    <Link LinkType="CommandScript" Express="" Priority="1" Tip=""
```

```
Key="无" Section="1">
```

```
    <DownLink LinkType="DownLink"><![CDATA[]]></DownLink>
```

```
<WhileLink LinkType="WhileLink"><![CDATA[]]></WhileLink>
<UpLink LinkType="UpLink"
Interval="0"><![CDATA[]]></UpLink>
<MoveInLink
LinkType="MoveInLink"><![CDATA[]]></MoveInLink>
<MoveOutLink
LinkType="MoveOutLink"><![CDATA[]]></MoveOutLink>
</Link>
</Menu>
```

2、动画连接属性

1) 线属性动画连接 Link 类型节点

LinkType: 动画连接类型, "LineProperty" 为线属性

Express: 动画连接表达式, "\\本站点\tag"

Pen: 笔属性列表 "{255, 0, 0, 0, 1, 0.000000}, {0, 0, 255, 0, 1, 100.000000}"



例如:

```
<Link LinkType="LineProperty" Express="//本站点\tag"
```

```
Pen="{255, 0, 0, 0, 1, 0.000000}, {0, 0, 255, 0, 1, 100.000000}"/>
```

线属性动画连接 Link 类型节点

LinkType: 动画连接类型, "FullFillProperty" 为线属性类型

Express: 属性表达式, 如 "\\本站点\tag"

Brush: 画刷属性列表“{{255, 0, 0, 0, 0, 0. 000000}, {0, 0, 255, 0, 0, 100. 000000}}”



例如:

```
<Link LinkType="FullFillProperty" Express="\本站点\tag"  
Brush="{{255, 0, 0, 0, 0, 0. 000000}, {0, 0, 255, 0, 0, 100. 000000}}"/>
```

2)、文本色动画连接 Link 类型节点

LinkType: 动画连接类型, "TextColor" 为文本色属性

Express: 动画连接表达式, 如"\本站点\tag+\本站点\tag"

Pen="{{255, 0, 0, 0. 000000}, {0, 0, 255, 100. 000000}}"



例如:

```
<Link LinkType="TextColor" Express="\本站点\tag+\本站点\tag"  
Pen="{{255, 0, 0, 0. 000000}, {0, 0, 255, 100. 000000}}"/>
```

3)、填充动画连接 Link 类型节点

LinkType: 动画连接类型, "FullFill"为填充动画连接

Express: 动画连接表达式, 如"\本站点\tag"

MinHeight: 最小填充高度, 整数, 如"0"

MaxHeight: 最大填充高度, 整数, 如"100"

MinPercent: 最小填充百分比, 0-100 整数, 如"0"

MaxPercent: 最大填充百分比, 0-100 整数, 如"100"

Direction: 填充方向, "1" 向右, "2" 向左, "3" 向上, "4" 向下

DefaultColor: 填充画刷颜色, 颜色值, "0"

DefauleStyle: 填充画刷样式, 如"0"

DefaultHatch: 填充阴影画刷样式, 如"0"



例如:

```
<Link LinkType="FullFill" Express="\本站点\tag" MinHeight="0" MaxHeight="100"  
MinPercent="0" MaxPercent="100" Direction="3" DefaultColor="0" DefauleStyle="0"  
DefaultHatch="0"/>
```

4)、缩放动画连接 Link 类型节点

LinkType: 动画连接类型, "Zoom" 为缩放动画连接

Express: 动画连接表达式, 如"\本站点\tag"

Direction: 缩放方向, "1"向右, "2"向左, "3"向下, "4"向上, "5"中心

MinValue: 缩放最小时, 表达式值, 如"0"

MaxValue: 缩放最大时, 表达式值, 如"100"

MinPercent: 最小缩放占据比, 如"0"

MaxPercent: 最大缩放占据比, 如"100"



例如:

```
<Link LinkType="Zoom" Express="\本站点\tag" Direction="3" MinValue="0"  
MaxValue="100" MinPercent="0" MaxPercent="100"/>
```

5)、旋转动画连接 Link 类型节点

LinkType: 动画连接类型, "Rotate"为旋转动画连接

Express: 动画连接表达式, 如"\\本站点\tag"

CounterClockwiseValue: 最大逆时针方向对应数值, 如"0"

CounterClockwiseAngle: 最大逆时针方向对应角度, 如"0"

ClockwiseValue: 最大顺时针方向对应数值, 如"100"

ClockwiseAngle: 最大顺时针方向对应角度, 如"360"

OffCenterHor: 旋转圆心水平偏离中心距离, 如"0"

OffCenterVer: 旋转圆心垂直偏离中心距离, 如"0"



例如:

```
<Link LinkType="Rotate" Express="\\本站点\tag" CounterClockwiseValue="0"  
CounterClockwiseAngle="0" ClockwiseValue="100" ClockwiseAngle="360"  
OffCenterHor="0" OffCenterVer="0"/>
```

6)、移动动画连接 Link 类型节点

水平移动(Hor)动画连接 Link Type="Slider"

LinkType: 移动类型, "Hor"为水平移动

Express: 动画连接表达式, 如"\\本站点\tag"

LeftValue: 最左对应数值, 如"0"

RightValue: 最右对应数值, 如"100"

MoveLeft: 最左移动距离, 如"0"

MoveRight: 最右移动距离, 如"100"



例如:

```
<Ver LinkType="Ver" Express="\\本站点\tag" UpValue="0" DownValue="100"  
MoveUp="0" MoveDown="100"/>  
  
</Link>
```

垂直移动(Ver)动画连接 Link Type="Slider"

LinkType: 移动动画连接类型, "Ver" 垂直移动动画连接

Express: 动画连接表达式, 如"\\本站点\tag"

UpValue: 最上边对应数值, 如"0"

DownValue: 最下边对应数值, 如"100"

MoveUp: 向上移动距离, 如"0"

MoveDown: 向下移动距离, 如"100"



例如:

```
<Ver LinkType="Ver" Express="\\本站点\tag" UpValue="0" DownValue="100"  
MoveUp="0" MoveDown="100"/>  
  
</Link>
```

7)、模拟值输出动画连接 Link 类型节点

LinkType: 动画连接类型, "AnalogOutput"为模拟值类型

Express: 动画连接表达式, 如"\\本站点\tag"

IntNum: 整数位数, 如“2”

DecimalNum: 小数位数, 如“0”

Binary: 数据显示格式, “0” 十进制, “1” 十六进制, “2” 科学技术法

CurrentOutputContent: 对齐方式, “0”左对齐, “1”中心对齐, “2”右对齐



例如:

```
<Link LinkType="AnalogOutput" Express="\本站点\tag" IntNum="2" DecimalNum="0"
Binary="0" CurrentOutputContent=""/>
```

8)、离散值输出动画连接 Link 类型节点

LinkType: 动画连接类型, “DiscOutput”为离散值输出

Express: 动画连接表达式, 如“\本站点\ \$新报警”

CurrentShowTip="1”

Tiptrue: 为真时, 输出文本信息, 如“打开”

Tipfalse: 为假时, 输出文本信息, 如“关闭”



例如:

```
<Link LinkType="DiscOutput" Express="\本站点\ $新报警" CurrentShowTip="1"
Tiptrue="打开" Tipfalse="关闭"/>
```

9)、字符串输出动画连接 Link 类型节点

LinkType: 动画连接类型, “StringOutput”为字符串输出类型

Express: 动画连接表达式, 如“\本站点\ \$用户名”

CurrentOutputContent: 字符串输出对齐方式, ""



例如:

```
<Link LinkType="StringOutput" Express="\本站点\$用户名"  
CurrentOutputContent=""/>
```

10)、模拟值输入动画连接 Link 类型节点

LinkType: 动画连接类型, "AnalogInput" 为模拟值输入

Express: 动画连接表达式, 如"\本站点>tag"

Priority: 输入操作优先级, 如"0"

Tip: 输入操作提示信息, 文本字符串, 如"请输入"

Key: 输入操作快捷键组合, 如"Shift+F6"

Section: 输入操作安全区, 如"1"

MinValue: 可输入数值最小值, 如"0"

MaxValue: 可输入数值最大值, 如"100"



例如:

```
<Link LinkType="AnalogInput" Express="\本站点>tag" Priority="0" Tip="请输入"  
Key="Shift+F6" Section="1" MinValue="0" MaxValue="100"/>
```

11)、离散值输入动画连接 Link 类型节点

LinkType: 动画连接类型, "DiscInput" 为离散值输入

Express: 动画连接表达式, 如"\本站点\\$启动历史记录"

Priority: 操作优先级, 如“0”

Tip: 输入操作提示信息, 如“开启历史存储”

Key: 输入操作快捷键组合, 如“Ctrl+Shift+B”

Section: 输入操作安全区, 如“1”

TipTrue: 离散值输入为 1 时提示信息, 如“打开”

TipFalse: 离散值输入为 0 时提示信息, 如“关闭”



例如:

```
<Link LinkType="DiscInput" Express="\本站点\$启动历史记录" Priority="0" Tip="开启历史存储" Key="Ctrl+Shift+B" Section="1" TipTrue="打开" TipFalse="关闭"/>
```

12)、字符串输入动画连接 Link 类型节点

LinkType: 动画连接类型, “StringInput”为字符串输入类型

Express: 动画连接表达式, 如“\本站点\\$时间”

Priority: 输入操作优先级, 如“0”

Tip: 输入操作提示, 如“请输入”

Key: 输入操作快捷键组合, 如“Ctrl+Shift+T”

Section: 输入操作安全区, 如“1”

IsPassword: 是否加密输入, “1”是, “0”否



例如:

```
<Link LinkType="StringInput" Express="\本站点\$时间" Priority="0" Tip="请输入"
```

" Key="Ctrl+Shift+T" Section="1" IsPassword="1"/>

13)、鼠标动画连接 Link 类型节点

公共 Link 节点属性

LinkType: 动画连接类型, "CommandScript" 鼠标动画连接

Priority: 鼠标操作优先级, 如"0"

Key: 鼠标操作快捷键组合, 如"Ctrl+Shift+B"

Section: 鼠标操作安全区, 如"1"

鼠标按下时 DownLink 节点属性

LinkType: 鼠标动画连接类型, "DownLink" 为鼠标按下

<![CDATA[ClosePicture("电气符号");]]>: 脚本程序段

鼠标弹起时 UpLink 节点属性

LinkType: 鼠标动画连接类型, "WhileLink" 为鼠标弹起时动画连接

<![CDATA[ShowPicture("电气符号");]]>: 脚本程序段

鼠标按住时 WhileLink 节点属性

LinkType: 鼠标动画连接类型, "UpLink" 鼠标按住时

Interval: 程序执行间隔, 如"1000"

<![CDATA[]>: 脚本程序段

鼠标进入时 MoveInLink 节点属性

LinkType: 鼠标动画连接类型, "MoveInLink" 为鼠标进入时动画连接

<![CDATA[]>: 脚本程序段

鼠标进入时 MoveOutLink 节点属性

LinkType: 鼠标动画连接类型, "MoveOutLink"为鼠标离开时动画连接

<![CDATA[]]>: 脚本程序段



例如:

```
<Link LinkType="CommandScript" Express="" Priority="0" Tip=""
Key="Ctrl+Shift+B" Section="1">
    <DownLink LinkType="DownLink"><![CDATA[ClosePicture("电
气符号");]]></DownLink>
    <WhileLink LinkType="WhileLink"><![CDATA[]]></WhileLink>
    <UpLink LinkType="UpLink"
Interval="1000"><![CDATA[]]></UpLink>
    <MoveInLink
LinkType="MoveInLink"><![CDATA[]]></MoveInLink>
    <MoveOutLink
LinkType="MoveOutLink"><![CDATA[]]></MoveOutLink>
</Link>
```

14)、闪烁动画连接 Link 类型节点

LinkType: 动画连接类型, "Flash"为闪烁动画连接

Express: 动画连接条件表达式, 如"\\本站点\tag==1"

IsBlink: "0"

Speed: 闪烁频率, 如"1000"



例如:

```
<Link LinkType="Flash" Express="\本站点\tag==1" IsBlink="0"  
Speed="1000"/>
```

15)、隐含动画连接 Link 类型节点

LinkType: 动画连接类型, "Hideshow"为隐含动画连接

Express: 动画连接表达式, 如"\本站点\tag==1"

ExpressIsTrue: 表达式为真, 如"1", 默认属性无需修改

BShow: 显示或隐藏, "1"表达式为真时显示, "0"表达式为真时隐藏



例如:

```
<Link LinkType="Hideshow" Express="\本站点\tag==1" ExpressIsTrue="1"  
BShow="1"/>
```

隐含动画连接 Link 类型节点

LinkType: 动画连接类型, "Flow"为流动动画连接

Express: 动画连接表达式, 如"\本站点\tag"

FlowState: 流动状态, 默认为"0"无需修改



例如:

```
<Link LinkType="Flow" Express="\本站点\tag" FlowState="0"/>
```

16)、滑动杆动画连接 Link 类型节点

滑动杆动画连接基本属性

LinkType: 动画连接类型, "Slider"为滑动杆动画

Priority: 输入操作优先级, 如"0"

Key: 输入操作快捷键组合, 如"无"

Section: 输入操作安全区, 如"1"

水平滑动杆输入动画连接节点

LinkType: 滑动杆输入动画连接类型, "HorInput"水平滑动杆输入

Express: 动画连接表达式, 如"\\本站点\tag"

LeftValue: 最左边对应数值, 如"0"

RightValue: 最右边对应数值, 如"100"

MoveLeft: 最左边移动距离, 如"0"

MoveRight: 最右边移动距离, 如"100"

垂直滑动杆输入动画连接节点

LinkType: 滑动杆动画连接类型, "VerInput"为垂直滑动杆输入

Express: 动画连接表达式, 如"\\本站点\tag"

UpValue: 最上边对应数值, 如"0"

DownValue: 最下边对应数值, 如"100"

MoveUp: 最上边移动距离, 如"0"

MoveDown: 最下边移动距离, 如"100"



例如：

```
<Link LinkType="Slider" Express="" Priority="0" Tip="" Key="无" Section="1">  
    <HorInput LinkType="HorInput" Express="\本站点\tag" LeftValue="0" RightValue="100" MoveLeft="0" MoveRight="100"/>  
    <VerInput LinkType="VerInput" Express="\本站点\tag" UpValue="0" DownValue="100" MoveUp="0" MoveDown="100"/>  
</Link>
```

3、精灵图对象属性

用户精灵图是组态王基本图形对象的组合。

公共属性：

Name: 精灵图名称，如"Graph11597"

Top: 精灵图左上顶点 Y 坐标，如"140"

Left: 精灵图左上顶点 X 坐标，如"450"

Bottom: 精灵图右下顶点 Y 坐标，如"271"

Right: 精灵图右下顶点 X 坐标，"701"

PenWidth: 画笔宽度，如"1"，采用默认无需修改

PenStyle: 画笔样式，如"0"，采用默认无需修改

PenColor: 画笔颜色，如"#00000000"，采用默认无需修改

BrushStyle: 画刷样式, 如“0”, 采用默认无需修改

BrushColor: 画刷颜色, 如“#FFFFFF00”, 采用默认无需修改

BrushHatch: 是否采用填充画刷, “0”, 采用默认无需修改

BrushFillStyle: 画刷填充风格, 如“0”, 采用默认无需修改

BrushFromColor: 过渡色开始颜色, 如“#FFFFFF00”, 采用默认无需修改

BrushToColor: 过渡色结束颜色, 如“#00000000”, 采用默认无需修改

IsLock: 是否锁定, “0”否, “1”是

Comment: 描述, 如“”, 采用默认无需修改

Number: 索引号, 如“1”, 采用默认无需修改

私有属性为各基本图形的属性及动画链接, 具体属性参见上面章节。

4、画面导入导出

画面可以单独或全部导出到 XML 格式的文件, 也可以从 XML 格式的文件单独或全部导入到组态王工程。具体做法参见组态王 7.0 初级培训教程第十四讲 XML 导入导出。

第二十七章 模板

- 👉 介绍模板管理器
- 👉 介绍如何通过工程如何生成模板
- 👉 介绍通过模板如何创建工程

组态王开发系统中为了快速的组态工程，提供通过模板创建工程的方式，可从模板管理器中选择模板生成到工程中，也可以将已经开发好的工程中各种配置信息生成模板。

什么是模板？

模板是生成设备、变量、画面、脚本的参考实例，它里面包括完整的配置信息，这些配置信息是在生成模板之前，在组态王开发环境中，以常规的组态方式创建的。组态王中提供的模板类型主要包括“采集模板”、“画面模板”、“脚本模板”和“图形模板”，不同的模板可以生成不同的配置信息。

27.1 模板管理器

组态王安装完成后，系统预定义的默认工作空间下的模板组及相应的模板会被加载到模板管理器界面，当用户在工程浏览器的目录显示区点击“模板”标签项，则弹出登录提示框，初始登录密码默认为空，当用户进入模板管理器后可以看到“默认工作空间”下系统预置的所有模板组及相应的模板，“默认工作空间”的路径为组态王安装目录下的

Template 文件夹，如下图所示：

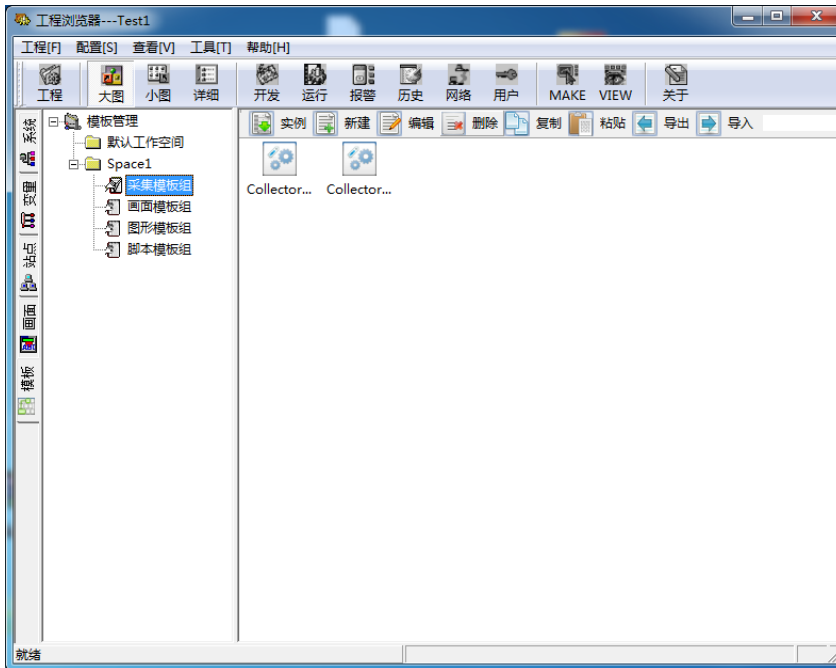


图 27.1 模板管理器

1、管理器设置

1) 登陆模板管理器

进入工程浏览器点击目录显示区的模板标签项，进入模板管理器界面，需要验证登陆权限，输入密码正确时，才能够进入到管理器中，管理器提供初始密码为空，用户可以在管理器中更改此密码。

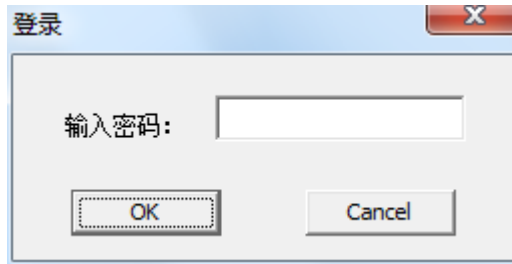


图 27.2 登陆管理器

2)、修改管理器密码

为保护新生成的模板，可以在模板管理器中修改管理器的默认密码，用新的密码对模板进行加密。在管理器浏览视图中，右键菜单选择“密码设置”，弹出密码设置界面，在密码文本框中，输入“新密码”和“确认密码”，点击确认后，即可完成对密码的修改。

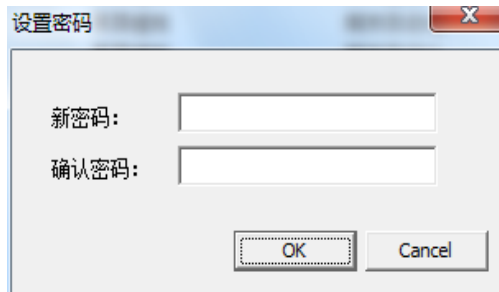


图 27.3 密码设置

2、工作空间

1) 新建工作空间

在模板管理器的目录树下右键点击模板管理根节点，弹出如下所示的列表框：

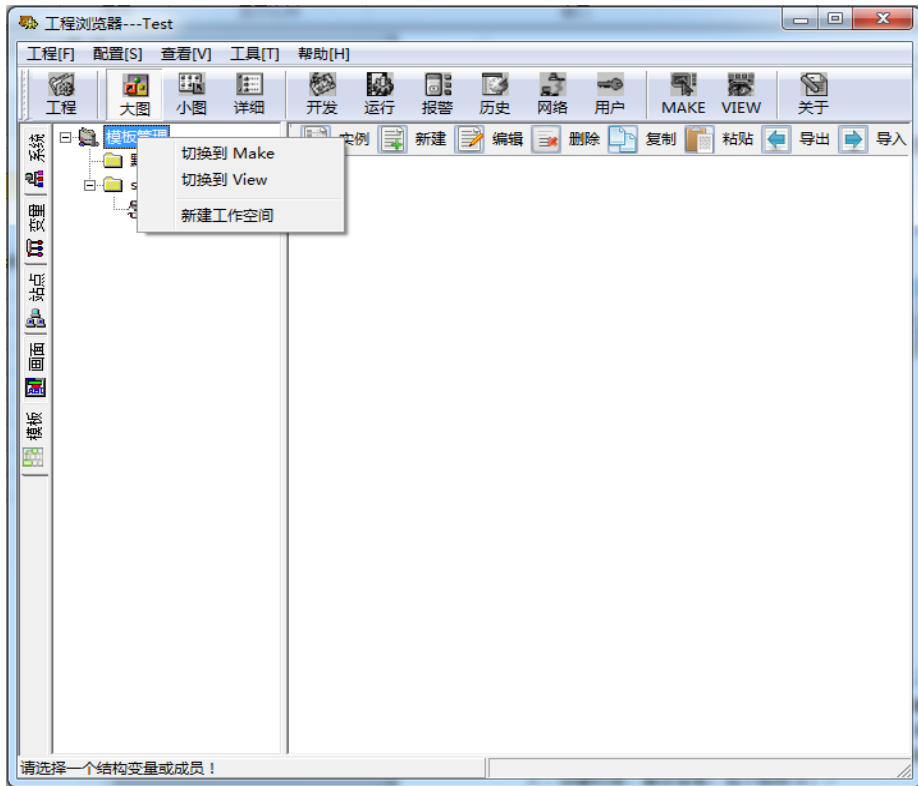


图 27.4 模板管理器根节点列表框

- 新建工作空间：通过点击该项，用户可以新建工作空间，此时弹出“新建工作空间”对话框，通过该对话框，为新建的工作空间设置空间名称、指定路径，并输入创建人、创建时间、描述信息，如下图所示：

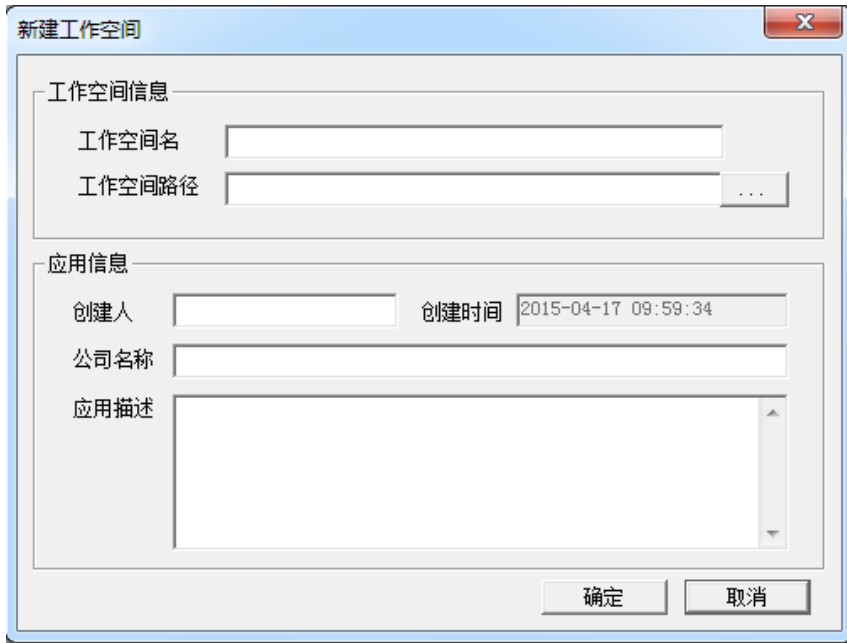


图 27.5 新建工作空间

工作空间名：命名的首字符只能为字符，不能为数字等非法字符，名称中间不允许有空格、算术符号等非法字符存在。名称长度不能超过 31 个字符。

工作空间路径：点击后面的按钮，弹出路径选择对话框，或者直接输入如 D:\ 港机工程。

创建人：由系统自动生成，此时只显示不可修改。

创建时间：由系统自动生成，此时只显示不可修改。

公司名称：输入公司名称，不超过 39 个字符。

应用描述：输入描述信息，不超过 39 个字符。

2) 删除、重命名工作空间

在模板管理器的目录树下右键点击工作空间节点，弹出如下所示的列表框：

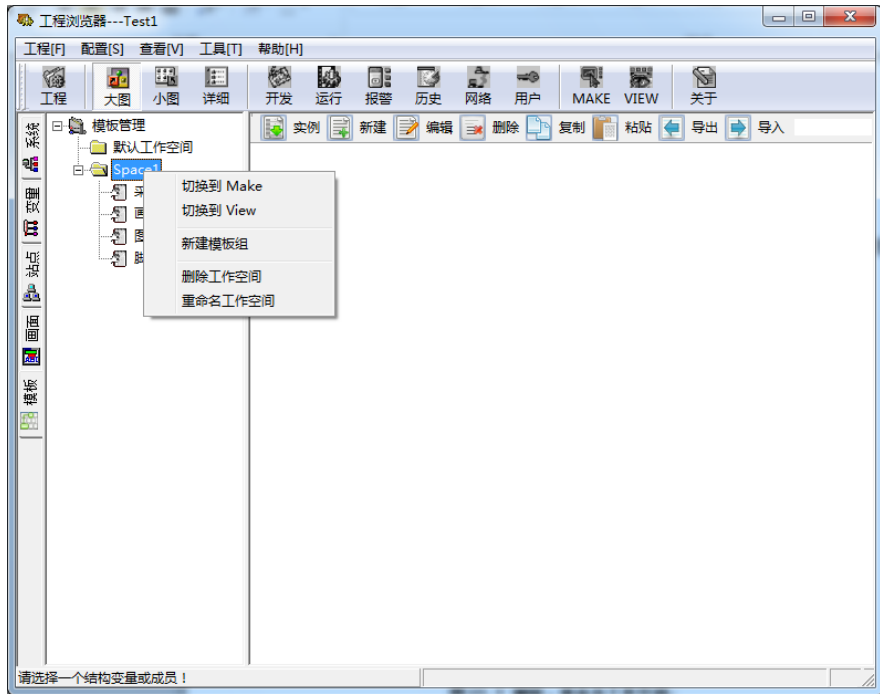


图 27.7 删除、重命名工作空间

- 删除工作空间：当鼠标点击“删除工作空间”则模板管理器中的工作空间不显示。
- 重命名工作空间：为工作空间重新命名。

3、模板组

选择某工作空间，点击鼠标右键，下拉列表显示切换到 Make、切换到 View、新建模板组、删除工作空间、重命名工作空间，如下图所示：

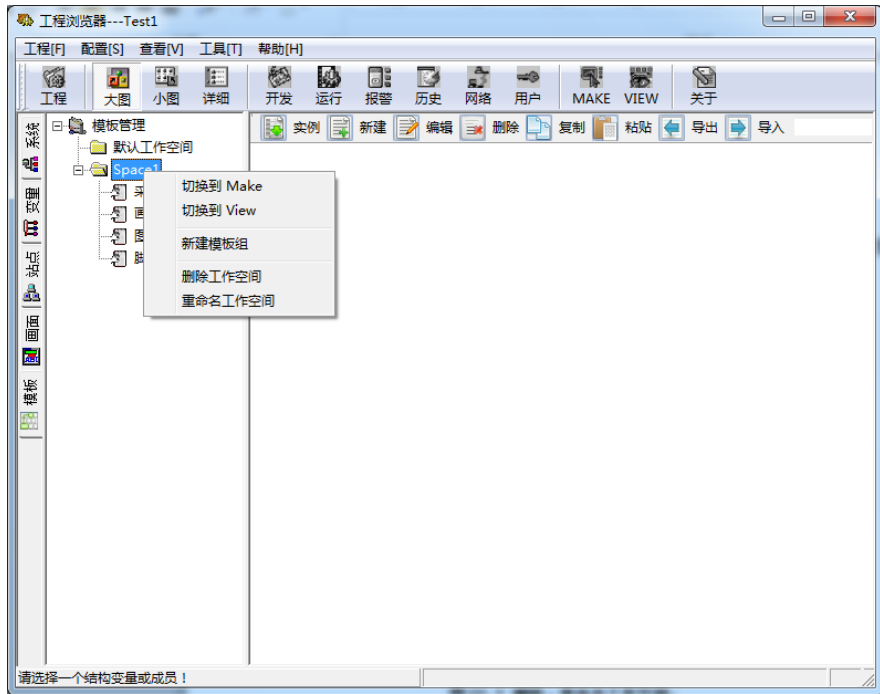


图 27.8 工作空间下的列表框

用户可以在默认工作空间或者新建的工作空间下新建模板组。

◇ 新建模板组

点击模板树下的某个模板组，右键菜单选择“新建模板组”项目，弹出新建模板组窗口，输入模板组名称以及对应的目录名，点击确认后在模板树中生成一个新的分组。

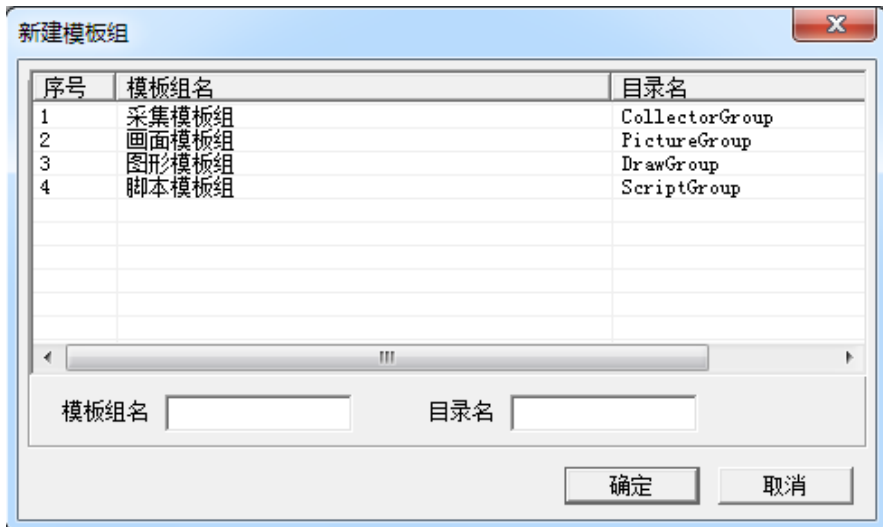


图 27.9 新建模板组

模板组名：命名的首字符只能为字符，不能为数字等非法字符，名称中间不允许有空格、算术符号等非法字符存在。名称长度不能超过 31 个字符。

目录名：不允许输入中文。

◇ 删除模板组

点击某个模板组，右键菜单选择“删除模板组”项目，弹出删除模板组提示窗口，点击确认后，将删除该模板组，若分组下存在模板文件，模板文件不会被删除。

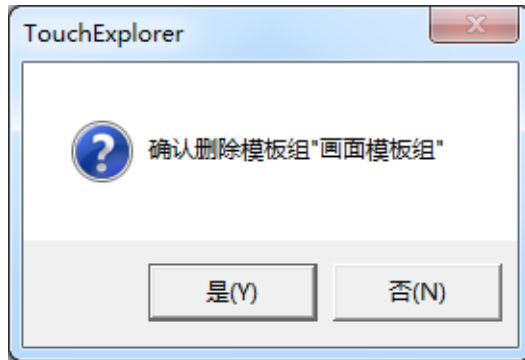


图 27.10 删除模板组

◇ 编辑模板组

点击某个模板组，右键菜单选择“编辑模板组”项目，弹出模板组编辑框，通过该编辑框只允许修改新的模板分组的名称，点击确认后该模板组的名称被修改，目录名不允许被修改。

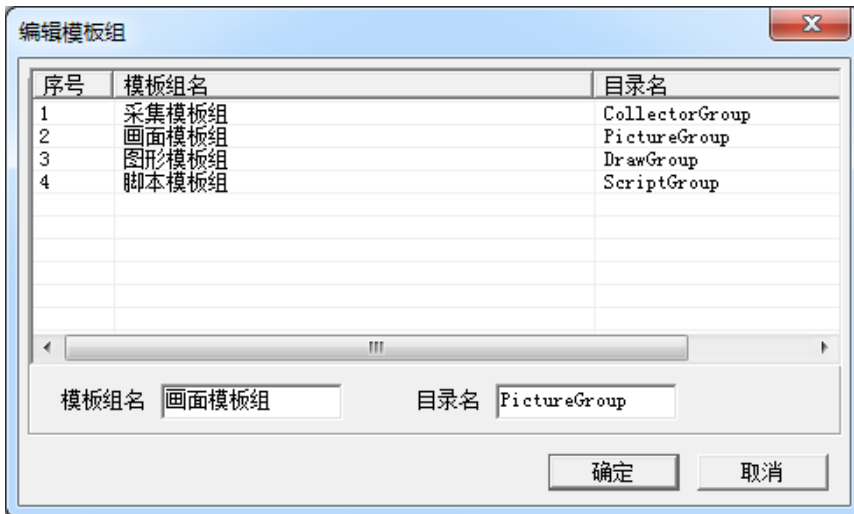


图 27.11 编辑模板组

4、模板

在模板管理器中可对任意分组下的模板文件进行操作，主要是通过工具栏和右键菜单方式，选择模板后并执行功能操作，下面介绍模板管理器中基本操作功能。

◇ 新建模板

点击工具栏上“创建”图标或右键菜单中选择“创建”项目，弹出要创建的模板的类型选择界面，选择某个类型图标，点击确定按钮，跳转到具体模板的向导界面，根据选择界面类型不同，跳转的向导界面不同。具体向导界面生成过程参见下面具体模板说明。

◇ 实例模板

在模板浏览视图窗口中，选择模板文件，点击工具栏上“实例化”图标或右键菜单中选择“实例化”项目，弹出实例化向导界面，根据选择模板类型不同，跳转的向导界面不同。具体向导界面参见下面具体模板说明。

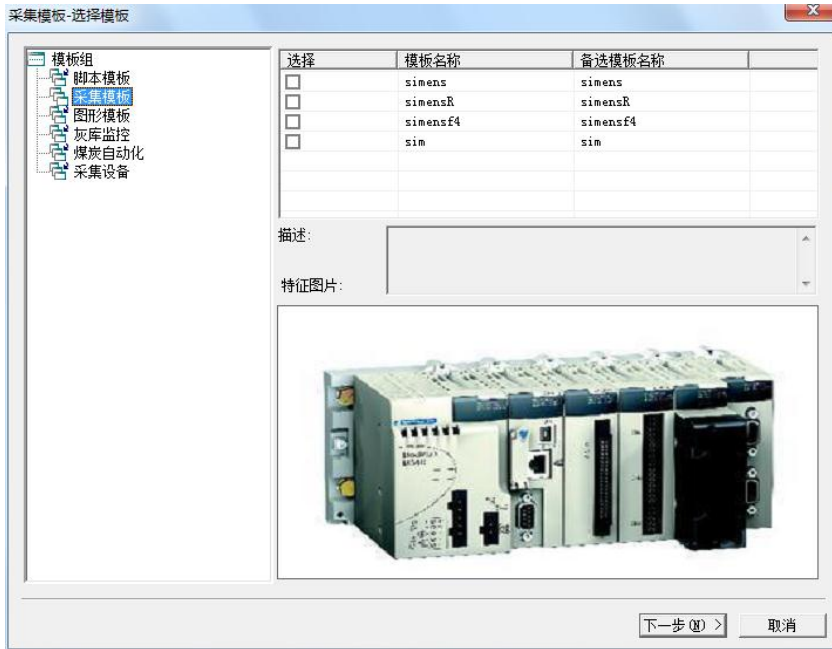


图 27.12 选择模板

◇ 删除模板

在模板浏览视图窗口中，选择模板文件，点击工具栏上“删除”图标或右键菜单中选择“删除”项目，提示用户是否确定删除信息窗口，点击确认后，选择的模板将被从管理器中删除。

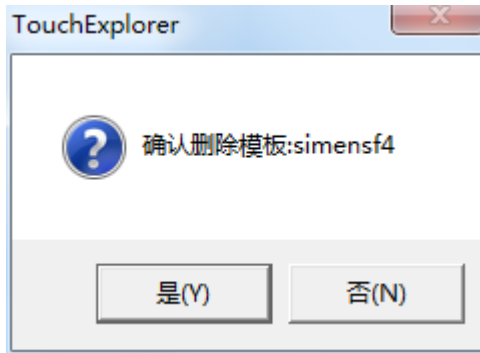


图 27.13 删除提示

◇ 编辑模板

在模板浏览视图窗口中，选择某模板文件，点击工具栏上“编辑”图标或右键菜单中选择“编辑模板”项目，验证管理器是否有权限编辑模板，若无权限，弹出密码验证窗口，输入验证密码，若有权限直接打开模板基本属性界面，具体编辑属性，参见每种类型模板说明。

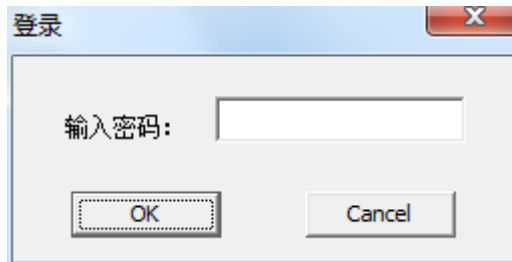


图 27.14 密码验证

◇ 复制、粘贴模板

在模板浏览视图窗口中，选择某模板文件，点击工具栏上“复制”图标或右键菜单

中选择“复制”项目，然后点击模板组，切换到另外一个分组中，点击工具栏中“粘贴”图标或右键菜单中选择“粘贴”项目，将模板粘贴在该分组下，并弹出修改模板属性对话框，需重新输入模板名称等相关属性。



图 27.15 复制模板

◇ 导出模板

在模板浏览视图窗口中，选择模板文件，点击工具栏上“导出”图标或右键菜单中选择“导出”项目，弹出导出存储路径选择窗口，选择路径后，点击确认将选择模板导出到该路径下。

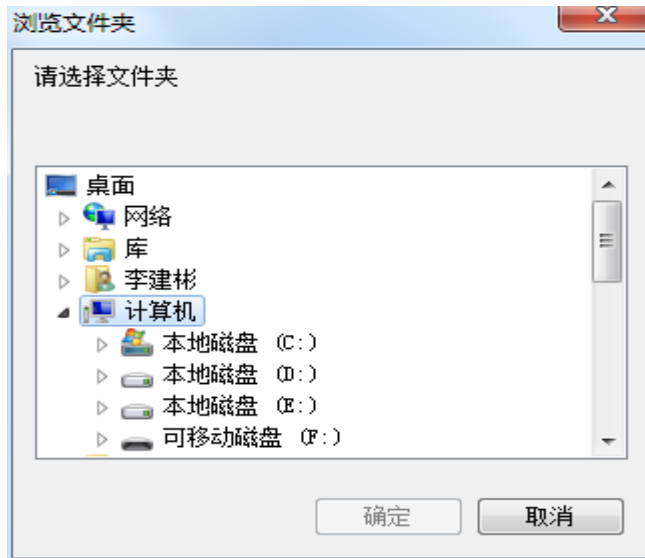


图 27.16 导出模板

◇ 导入模板

在模板浏览视图下，点击工具栏上“导入”图标或右键菜单中选择“导入”项目，弹出导入路径选择窗口，选择路径后，点击确认将该路径下的模板文件导入到当前模板组中。

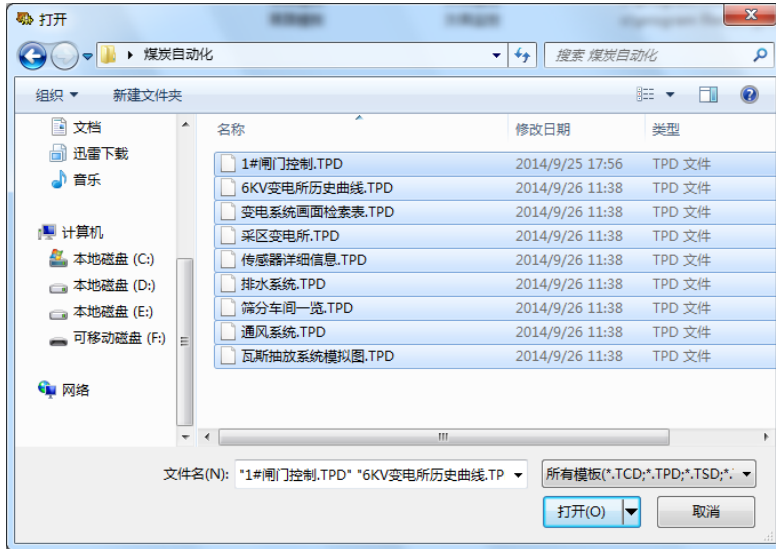


图 27.17 导入模板

◇ 查找模板

在工具栏查找文本框中输入要查询的模板文件名称，点击“查询”在当前组下查找符合条件的模板文件，并筛选显示到浏览视图中。

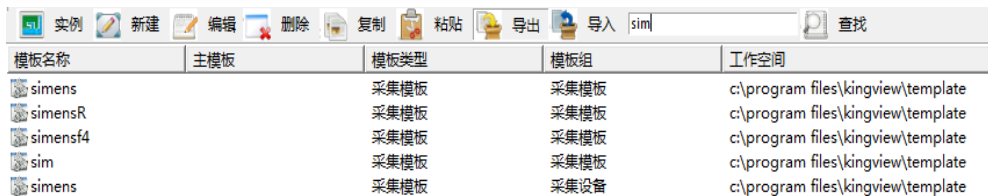


图 27.18 查找模板

27.2 采集模板

在组态王中建立了设备和变量才能够进行数据数据采集，所以采集是针对设备和变量来说的。

什么是采集模板？

采集模板是一种可以在组态王工程中生成设备信息和变量信息参考模板，它能够快速的建立变量和设备，省略掉复杂的设备配置和变量配置过程，不需要一个一个去创建设备和变量。采集模板在创建的时候，预先已经配置好了若干个设备和变量信息，在创建新工程时，直接生成就可以。

27.2.1 如何生成采集模板

准备工作

进入组态王工程浏览器，在设备管理视图下，创建多个类型的设备，这里创建两个设备一个为 modbus 设备，一个为西门子 s200 ppi 设备，设备名称分别为 ModbusRTU 和 simens，如下图。

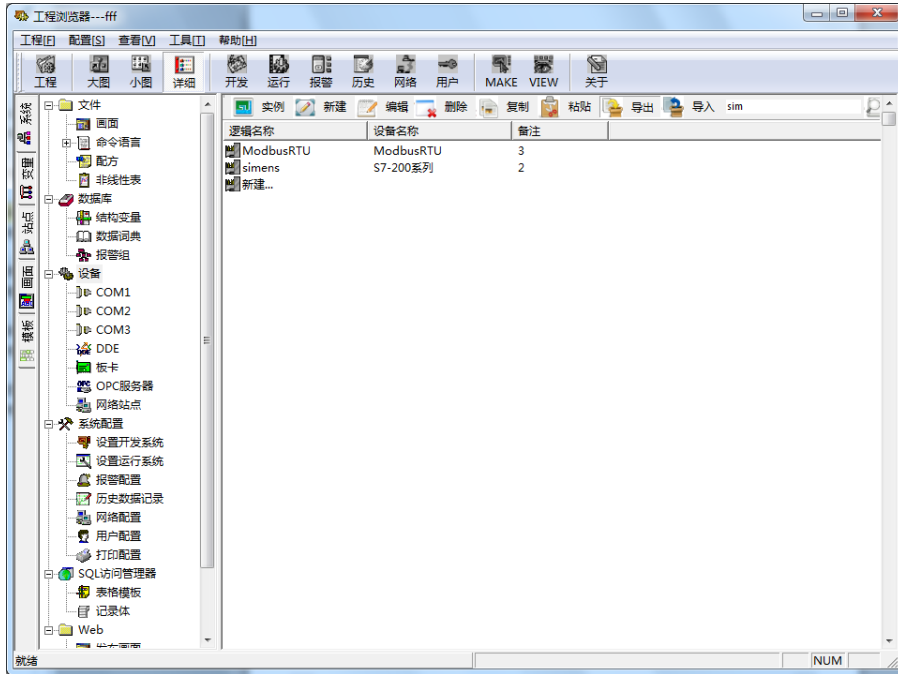


图 27.19 创建设备

在数据词典管理视图下，针对上述创建的设备分别创建变量，这里分别创建 modbus 采集设备的变量 MRTUtag1-Modtag5，s200 采集设备的变量 PPItag1-PPItag5，创建内存变量 Memtag1-Memtag5。

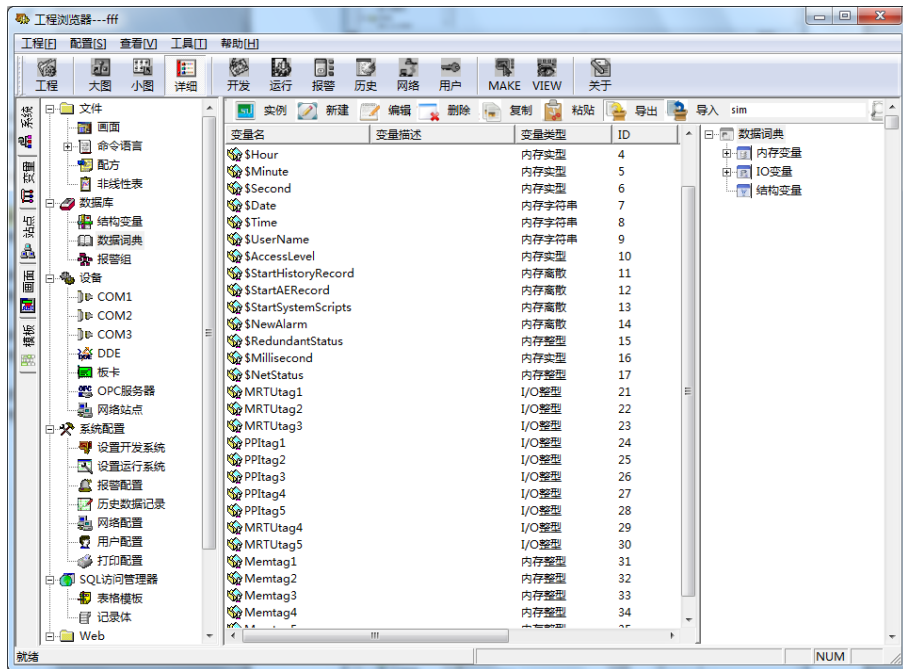


图 27.20 创建变量

生成采集模板

进入模板管理器，在工具栏点击“新建”图标，或右键菜单选择“新建”项目，弹出模板类型选择对话框，选择采集模板，进入采集模板生成向导。

1、设置采集模板属性信息

采集模板属性表述模板基本特征信息，在模板管理器中可编辑，它主要包括以下内容。

- ◇ **模板名称：**采集模板的标识名称，在同一模板组下不允许同名。
- ◇ **模板组：**采集模板存储的分组，可选择模板管理器中的任何一个分组。
- ◇ **模板安全：**模板安全，是采集模板的操作权限设置，它包括加密和无限制。

- 1) 无限制，不受权限限制，任何人都可编辑或实例化模板；
- 2) 加密，受模板管理器密码保护，打开时需要验证密码
- ◇ **主模板：**设置该模板是哪个采集模板的备选模板。
 - 1) 只能选择一个主模板，可以设置无主模板；
 - 2) 从当前模板组中选择主模板（选择组中已经存在的主模板）。
- ◇ **为主模板：**此选项对模板当前模板组有效，设置为主模板，将模板组创建到工程时，默认为选中的模板。
- ◇ **创建人：**生成采集模板时，用户自行输入标识创建模板或编辑模板的用户信息。
- ◇ **模板描述：**用户输入的关于模板的文本描述信息，可以作为此采集模板的一些简单功能说明。
- ◇ **特征图片：**标识采集模板的特征图片（位图），可选择进行添加。
- ◇ **创建时间：**模板的生成时间，文件存储时自动生成，编辑修改后自动更新。



图 27.21 采集模板属性

设置完设备基本属性信息，点击“下一步”进入变量选择界面，设置变量信息。

2、从变量列表中选择要生成采集模板中的变量

进入变量选择界面，在此界面中列举出当前工程中所有变量（系统变量除外），根据变量树可切换浏览各种类型的变量信息。在此列表中选择上面建立的 modbus 采集设备和 ppi 采集设备中的变量以及基本变量中的内存变量，可通过“全选”进行选择。点击“下一步”进入设备选择界面。

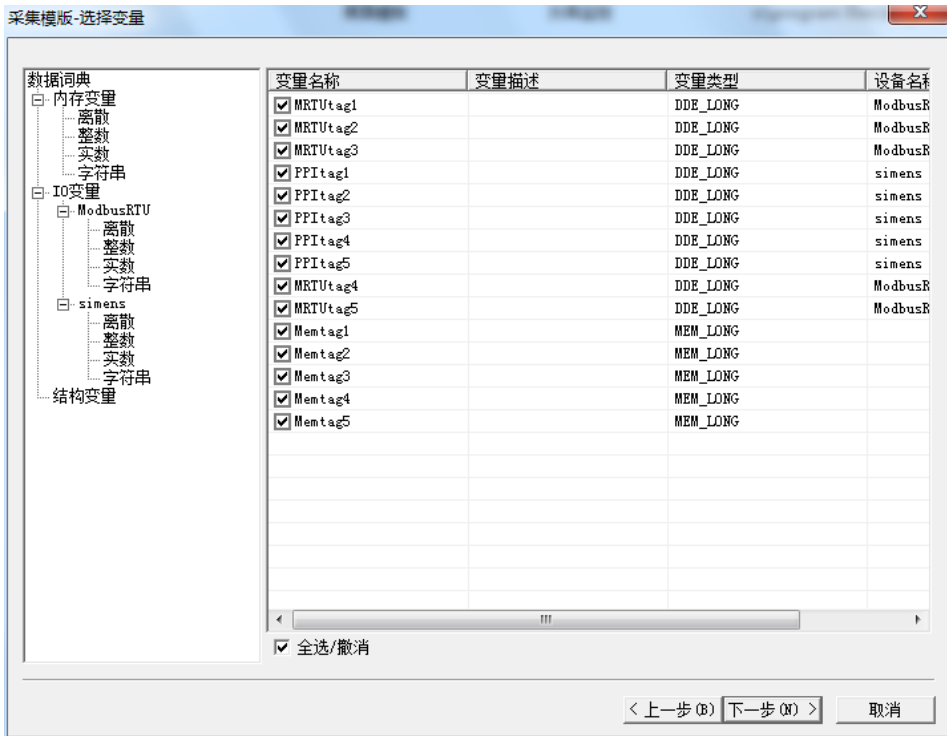


图 27.22 选择变量

3、从设备列表中选择设备

进入设备列表界面，显示当前工程中的设备，在上一步若选择了某个设备下的变量，本界面设备列表中的该设备默认选中，用户还可以选择其它设备。






模板名称	主模板	模板类型	模板组	工作空间
 simens		采集模板	采集模板	c:\program files\kingview\template
 simensR		采集模板	采集模板	c:\program files\kingview\template
 simensf4		采集模板	采集模板	c:\program files\kingview\template
 sim		采集模板	采集模板	c:\program files\kingview\template
 工程采集		采集模板	采集模板	c:\program files\kingview\template

图 27.24 模板管理器中采集模板

在模板管理器中某个分组下，选择一个或多个采集模板，点击工具栏上“实例”图标弹出生成设备向导界面。

1、选择采集模板

进入采集模板选择界面，默认选中在模板管理器选择的模板，还可以重新勾选，若模板存在备选模板，也可以选择备选模板。



图 27.25 模板选择界面

选择模板完成后，点击“下一步”，进入设备属性配置界面。

2、设置设备属性

进入模板中设备列表界面，点击每个设备时，在列表中若存在同名或同链路地址相同的设备，该设备会显示不同的颜色，点击切换设备时，可以显示该设备的配置信息，更具情况进行修改（具体设备配置的修改根据不同的设备有差别，需要参见相关帮助说明）。

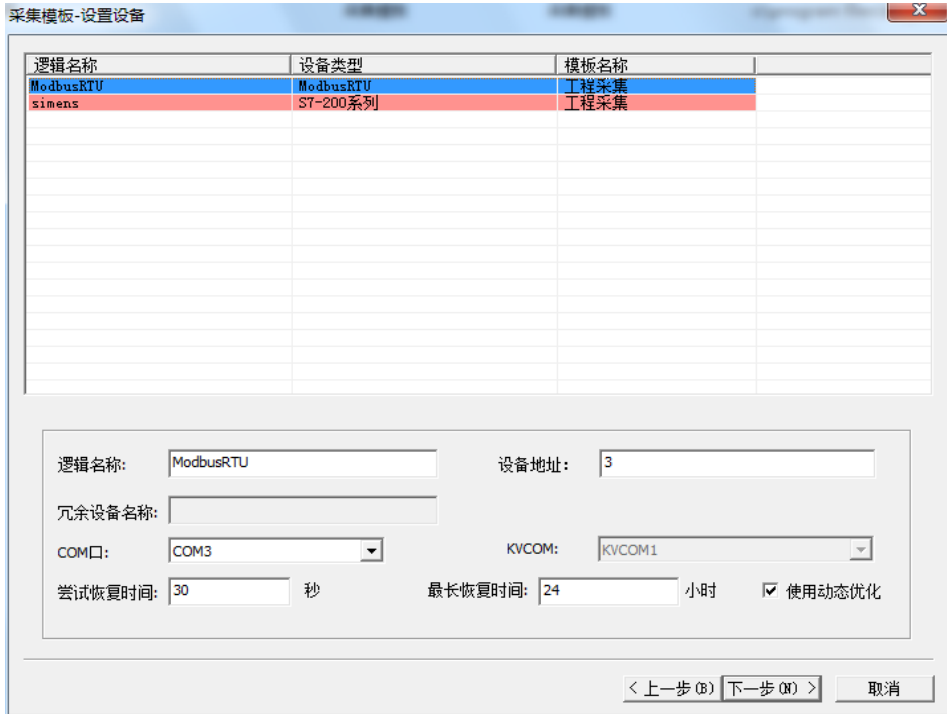


图 27.26 设置设备属性

⚠ 注意：若无重名等信息提示，请不要随意修改设备配置信息。

设备配置信息设置完成，并无提示时，点击“下一步”，进入变量选择界面。

3、变量选择

进入变量选择界面，显示选择模板中设备中的所有变量，通过变量树可以切换不同设备或不同类型的变量，通过变量列表中的选择列选择变量。列表中的变量出现同名或与当前工程中的变量同名，同名的变量行以不同颜色标示，提示必须输入替换变量名。若无

提示重名时，方可操作完成操作。

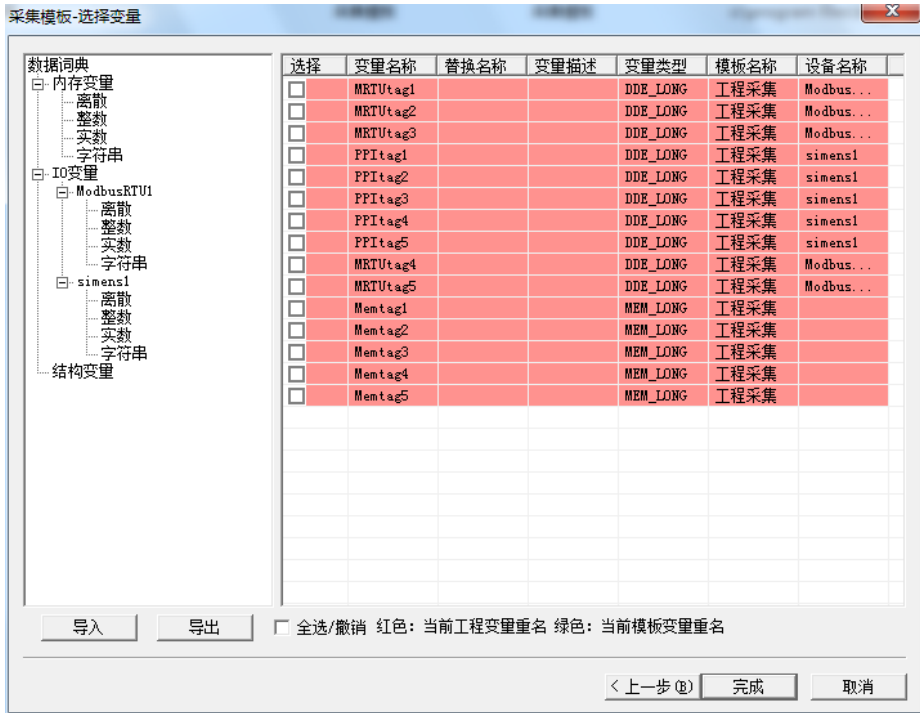


图 27.27 变量选择

变量选择完成后，并无重名提示信息，点击“完成按钮”，即可在当前工程中生成模板中的设备和变量。进入到设备管理视图和数据词典可查看到这些信息。

27.3 画面模板

什么是画面模板？

画面模板是可以生成一个工程画面的参考模板，它是一个完整的画面，模板里面的内容信

息与用户在创建模板是选择的画面相同。利用画面模板可快速的生成画面，为工程的二次开发节省了时间。下面介绍如何从一个画面生成模板和从模板生成画面。

27.3.1 如何生成画面模板

准备工作

进入组态画面管理器，新建一幅监控画面，打开画面，并在画面上绘制图形。在上节中创建的 I0 变量和内存变量，创建图形的动画连接。

图素 1 文本输出，text1 模拟量输出，关联\\local\Modtag1;

图素 2 文本输出，text2 模拟量输出，关联\\ local \PPItag1;

图素按钮 1，模拟值输入，关联\\ local \PPItag2;

图素按钮 2，模拟值输入，关联\\ local \Modtag2;

图素按钮 3，按钮脚本，关联\\ local \MemTag6=“temperature”;

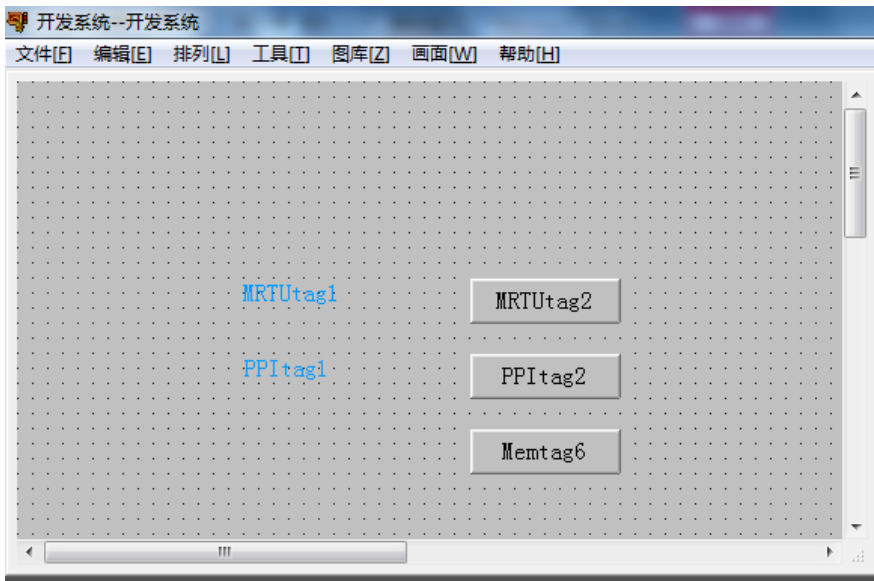


图 27.28 新建画面

生成画面模板

进入模板管理器，在工具栏上点击“创建”图标，或右键菜单选择“创建”项目，弹出生成创建模板类型选择界面，在此界面中选择“画面模板”类型，点击“确定”打开画面选择界面。

1、从画面列表中选择画面

在画面选择列表中显示了当前工程中的所有画面，因为上面只创建了一个画面，所以在本列表中，只显示上面建立的画面，点击该画面，可以显示画面的缩略图，选择该画面，点击“下一步”进入变量选择界面。

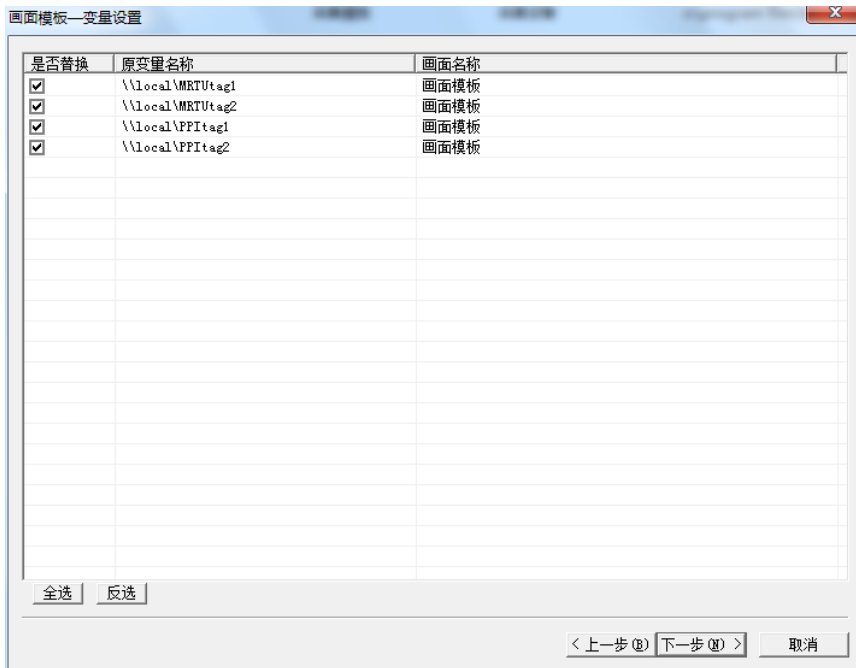


图 27.30 选择替换变量

3、选择能够被替换的字符串

在字符串替换界面中，列举出各个画面中能被替换的字符串信息，这里只显示一个画面中的字符串，脚本中使用的字符串，点击“选择”字段的选择按钮可以选中该字符串，被选中的字符串能够在模板创建画面时被替换。通过点击“全选”按钮，可以选择全部字符串，通过“反选”按钮，可以反向选择字符串。

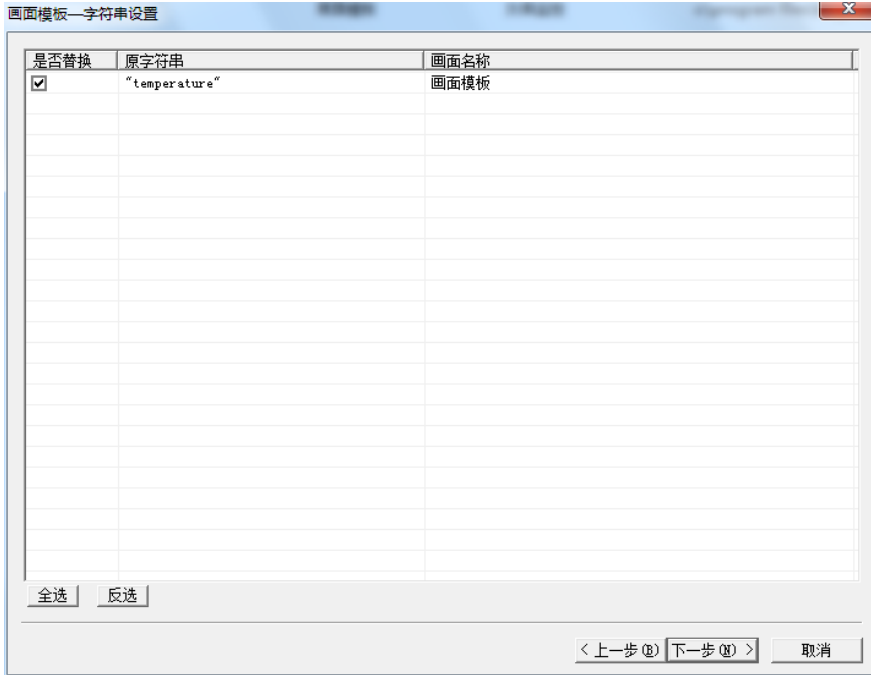


图 27.31 选择替换字符串

选择字符串完成后，点击“下一步”按钮，跳转到模板属性定义界面，设置模板的基本属性信息。

4、设置画面模板基本属性

在画面模板基本属性界面中，可同时定义多个模板的属性，这里只生成一个画面模板，只需定义上面选择的画面模板属性，可设置以下属性信息。

- ◇ **模板名称：**模板的标识名称，在同一模板组下不允许同名，可填写字符串信息，默认为画面名称。
- ◇ **模板安全：**模板安全，是模板的操作权限设置，它包括加密和无限制。

- 1) 无限制，不受权限限制，任何人都可编辑或实例化模板；
- 2) 加密，受模板管理器密码保护，打开时需要验证密码
- ◇ **主模板：**设置该模板是哪个画面模板的备选模板。
 - 只能选择一个主模板，可以设置无主模板；
 - 从当前模板组中选择主模板（选择组中已经存在的主模板）。
- ◇ **为主模板：**此选项对模板当前模板组有效，设置为主模板，将模板组创建到工程时，默认为选中的模板。
- ◇ **依赖模板：**此画面模板中的变量所依赖的采集模板。
- ◇ **创建人：**生成模板时，用户自行输入标识创建模板或编辑模板的用户信息。
- ◇ **模板描述：**用户输入的关于模板的文本描述信息，可以作为此采集模板的一些简单功能说明。
- ◇ **创建时间：**模板的生成时间，存储时自动生成，编辑修改后自动更新。

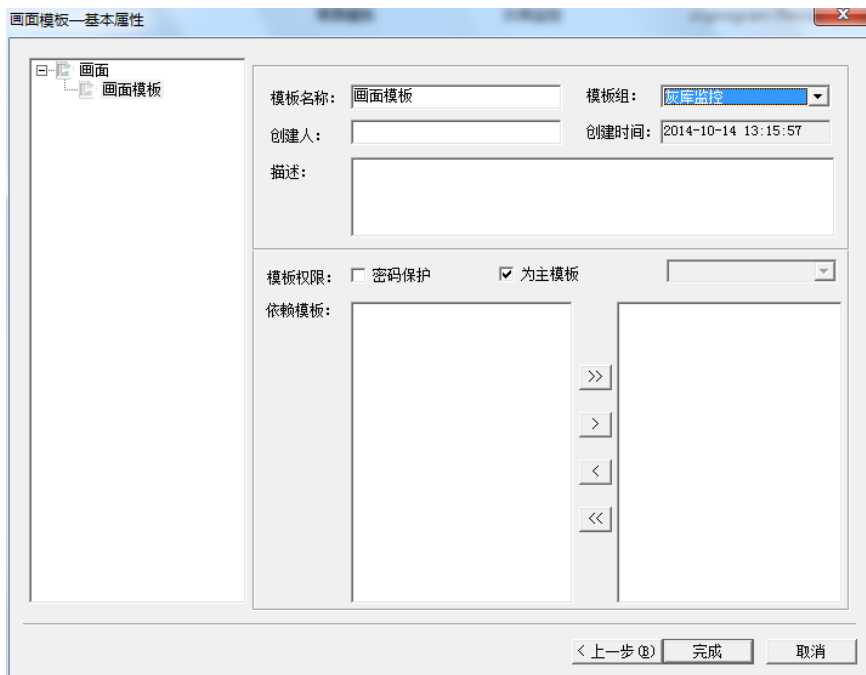


图 27.32 设置模板基本属性

设置完画面模板基本属性，点击“完成”按钮，生成画面模板，并将模板存储到模板管理器指定的分组下。

! 注意：生成的画面模板，在模板管理器中还可以对模板基本属性信息进行编辑，但不会改变画面的内容（包含的变量和字符串及画面上的图形）。

27.3.2 通过画面模板生成画面

模板管理器中的画面模板都可以生成到组态王工程中，一般情况下，画面模板是根据工程的特性或行业特性进行分组的，在生成工程画面时，需要根据需要选择模板。

打开模板管理器，在某个模板分组下，选择类型为“画面模板”的模板，可以多选，点击工具栏上“实例”图标或右键菜单选择“实例”项目，弹出画面模板生成画面向导界面。

模板名称	主模板	模板类型	模板组
石灰石输送		画面模板	灰库监控
参数		画面模板	灰库监控
灰库		画面模板	灰库监控
除灰		画面模板	灰库监控
view		画面模板	灰库监控
画面模板		画面模板	灰库监控

图 27.33 选择画面模板

1、选择模板，设置画面名称

进入画面模板选择界面，通过点击选择按钮，可以“选择”和“取消选择”模板，在替换名称字段中，若画面名称出现同名，需要输入新的画面名称，设置完成后，点击“下一步”跳转到变量替换界面。

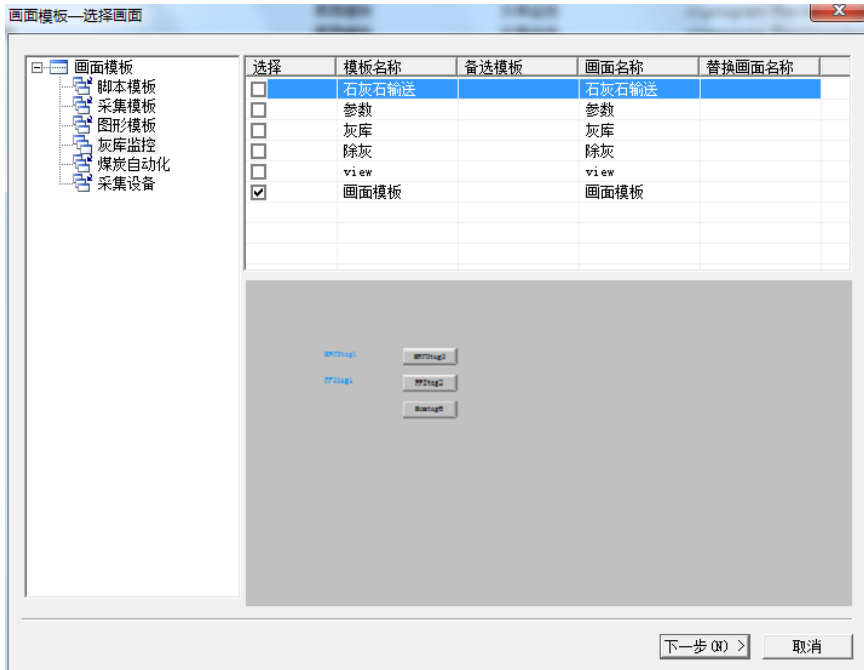


图 27.34 设置画面名称

2、替换画面中使用变量的名称

打开变量替换界面，在此界面中显示所有选模板画面中使用到的所有变量（OCX 控件除外），双击“替换变量名”字段列表，打开变量选择器，可从当前工程数据词典中选择变量，在工程数据词典中不存在的变量均需要替换，替换变量输入完成后，点击“下一步”跳转到字符串替换界面。

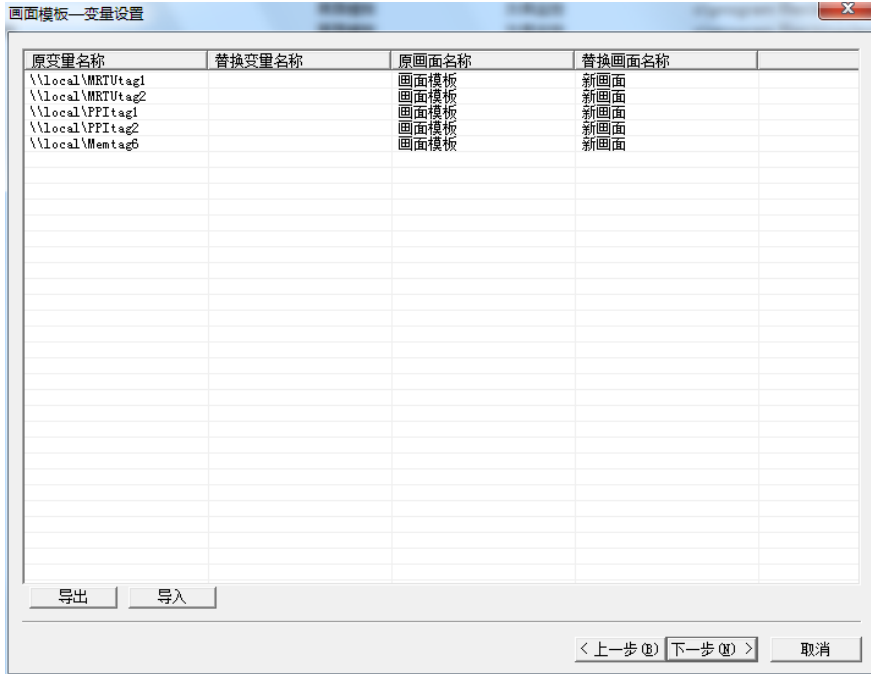


图 27.35 替换变量

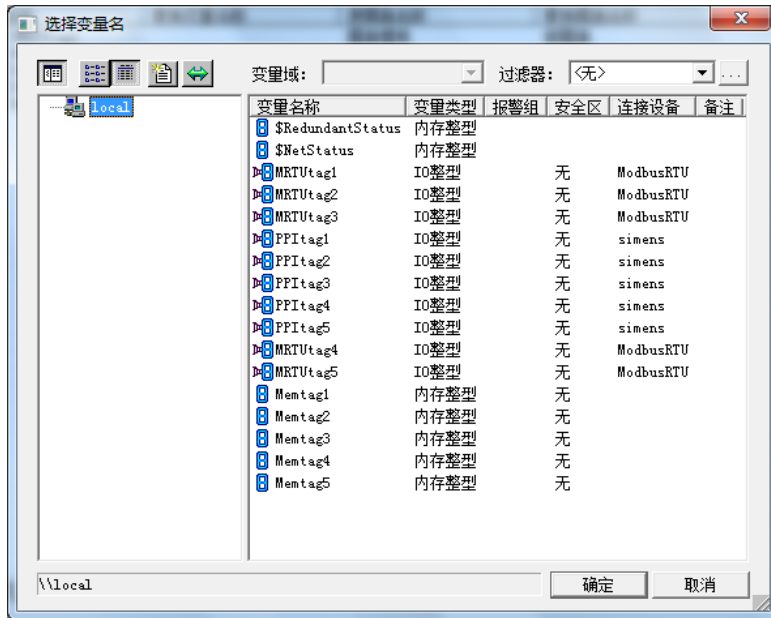


图 27.36 选择变量

3、替换画面中使用的字符串

在字符串替换界面中，显示选择模板画面脚本中使用到的所有字符串，可以根据工程需要进行替换，双击替换字符串字段列表输入编辑即可，输入替换字符串确定无误后，点击“完成”按钮，结束画面生成向导，在工程中生成选择模板中的画面。

命令语言程序。

27.4.1 如何生成脚本模板

要制作脚本模板，首先在组态王工程浏览器下命令语言中编写各种命令语言，这与开发组态王工程是相同的。

准备工作

编写命令语言程序。在这里分别编写应用程序命令语言、数据改变命令语言 (MRTUtag1)、事件命令语言 (\\local\MRTUtag2=10==10)。

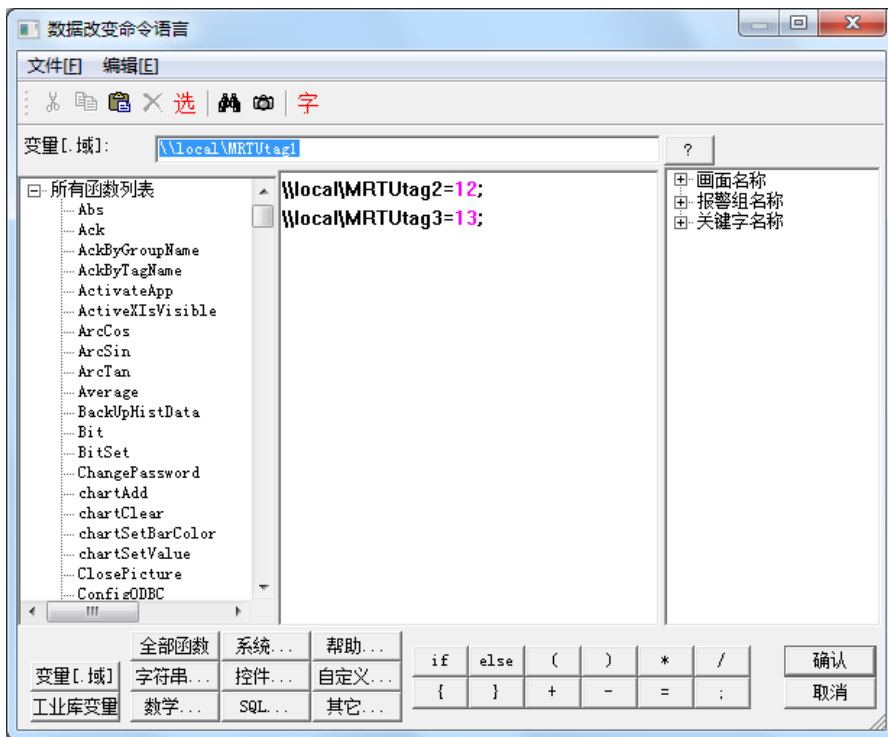


图 27.38 编写命令语言程序

生成模板

1、选择“脚本模板”类型

在模板管理器中，点击工具栏上“新建”或右键菜单选择“新建”项目，弹出新建模板类型选择界面，在模板类型列表中选择“脚本模板”，跳转到脚本模板生成向导界面。

2、选择命令语言

进入命令语言选择界面，列表中显示当前工程中所有编写的命令语言，通过选择按钮，可勾选多个命令语言，在这里勾选上面编写的命令语言，如图所示。选择完成后，点击“下一步”跳转到变量选择界面。

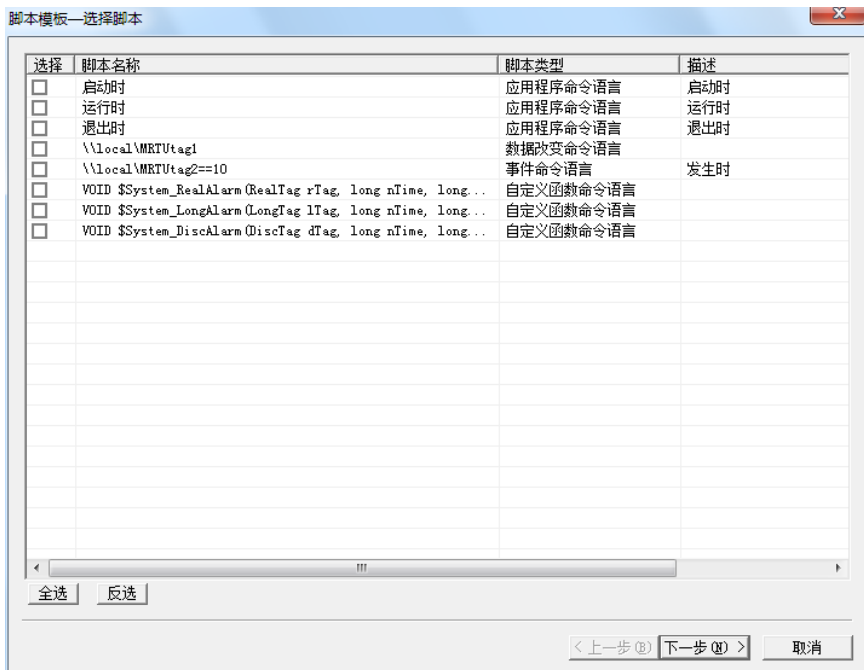


图 27.39 选择命令语言

3、选择替换变量

进入变量选择界面，列表中显示选择命令语言中用到的数据词典中的变量，在此界面中定义该变量是否允许被替换（模板生成脚本时），通过全选和撤销可选择全部变量，这里设置选择全部变量。变量选择完成后，点击“下一步”跳转到字符串选择界面。

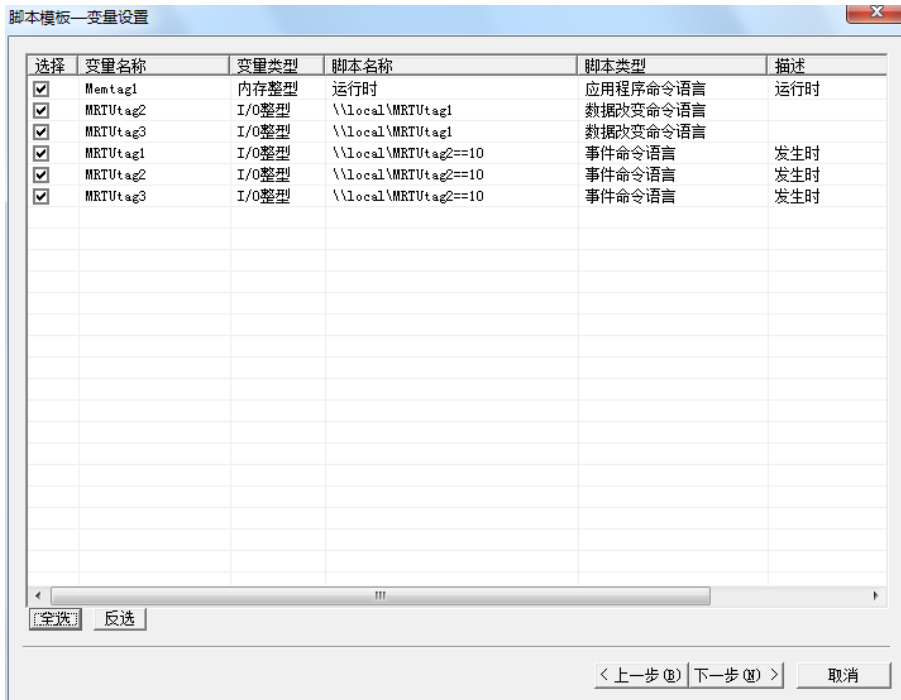


图 27.40 选择替换变量

4、选择替换字符串


进入字符串选择界面，列表中显示选择命令语言中用到的字符串，在界面中定义字符串是否允许被替换（模板生成脚本时），通过全选和撤销可选择全部字符串，这里选择全部字符串。选择完成后，点击“下一步”跳转到模板基本属性定义界面。

- ◇ **主模板：**脚本模板做为备选模板时，选择的主模板。
- ◇ **为主模板：**将脚本模板设置为主模板，可作为备选模板的选择。
- ◇ **依赖模板：**脚本模板中使用的变量依赖的采集模板。
- ◇ **创建人：**对模板作者的描述性字符串。



图 27.42 定义脚本模板基本属性

脚本模板基本属性设置完成后，点击“完成”按钮，生成脚本模板，并存储到模板管理器中选择的分组下。

 备注：设置脚本模板基本属性时，除模板名称外，其它属性均可采用默认属性，

输入完成模板名称后，可直接完成向导。

27.4.2 通过脚本模板生成命令语言

上一节中介绍如何将命令语言生成一个脚本模板，接下来介绍通过脚本模板生成命令语言，同样是以上一节中生成的脚本模板举例说明。

1、选择脚本模板

在模板管理器中选择脚本模板后，点击工具栏中的“实例”图标或右键菜单中选择“实例”项目，进入模板生成脚本向导，弹出模板选择界面，在模板列表中可以选择和取消选择脚本模板，若脚本模板存在备选模板，也可以选取备选的脚本模板，此选择模板无备选模板，采用默认选择即可。选择完成后，点击“下一步”，跳转到变量替换界面。

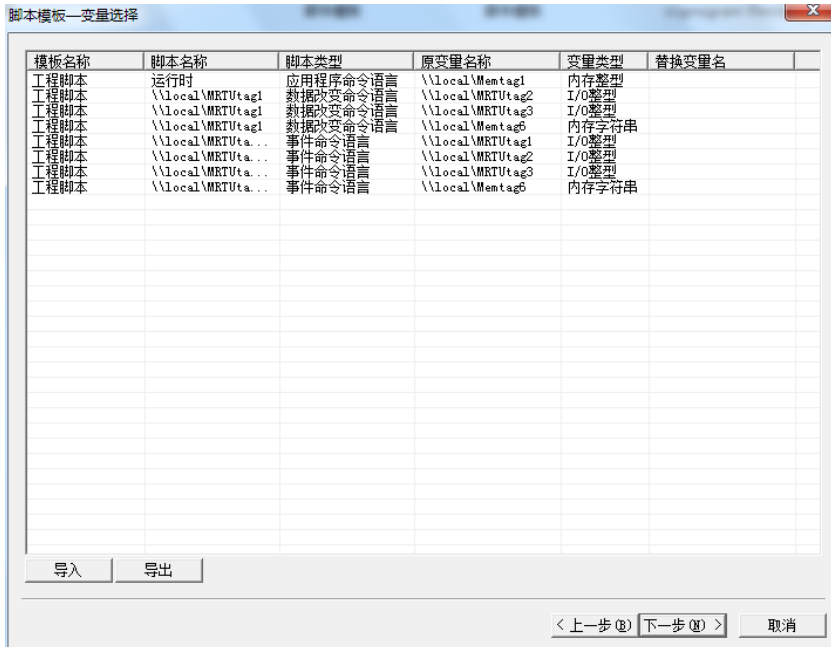


图 27.44 替换变量

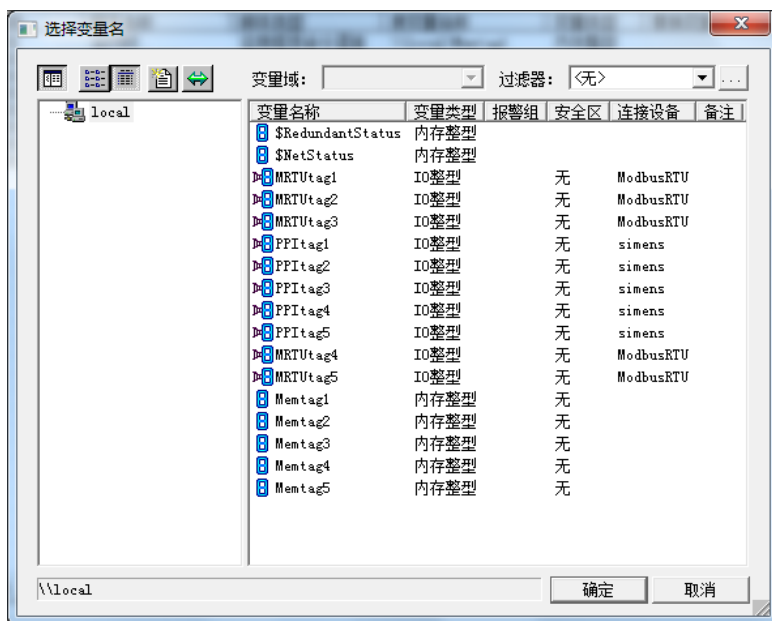


图 27.45 选择或新建变量

变量替换设置完成后，点击“下一步”，跳转到字符串替换界面，进行字符串设置。

3、替换脚本模板中字符串

进入字符串替换界面，列表中显示选择脚本模板中能够被替换的字符串，在替换字符串字段中，可输入新的替换字符串，也可以采用默认。替换字符串输入完成后，点击“完成”按钮，在当前工程中生成模板中的命令语言，并结束命令语言生成向导。

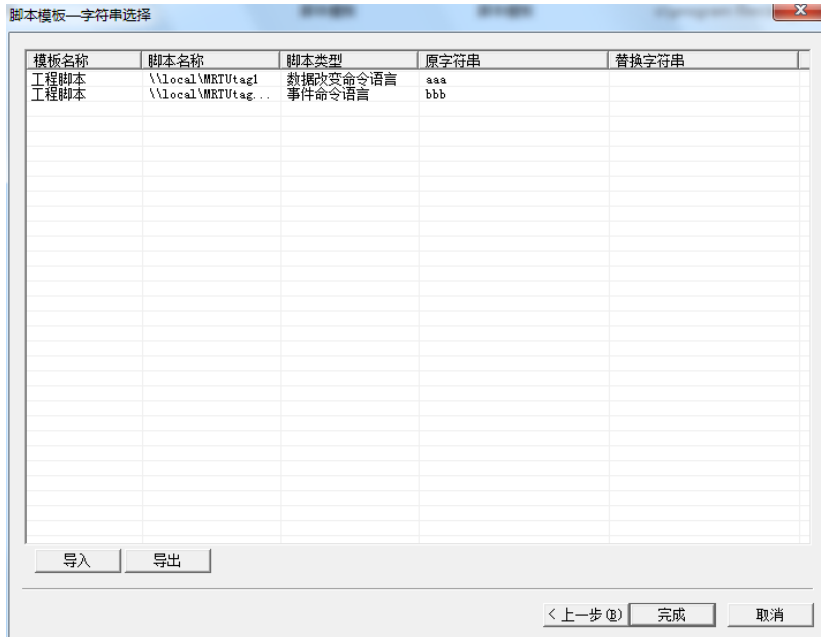


图 27.46 替换脚本中字符串

27.5 图形模板

什么是图形模板？

图形模板是一种特殊的画面模板，它与画面模板一样可以生成完整的工程画面，画面中包括精灵的布局阵列，在生成画面时布局阵列里的精图及精灵图的个数可以进行配置，在画面生成均匀分布的精灵图。

什么是布局阵列？

布局阵列式一种特殊的布局图素，它提供精灵图的均匀布局，并可修改需要布局的精灵图，

在图形画面上以标准的行列方式分布精灵图，在组态画面时，需要为其制定精灵图及布局属性。

27.5.1 如何生成图形模板

要制作图形模板，首先要在组态王画面编辑器中组态一副画面，画面中绘制一个或多个布局阵列图素，并为其设置精灵图。画面组态完成后通过向导界面，生成图形模板并存储到模板管理器中。

准备工作

组态包含阵列的画面。在画面编辑器中创建一个画面 view2，并在画面上绘制两个布局阵列，并为其添加精灵图，设置对应的动画连接属性，关联数据词典中变量，如下图所示：

布局名称：命名的首字符只能为字符，不能为数字等非法字符，名称中间不允许有空格、算术符号等非法字符存在。名称长度不能超过 31 个字符。

布局行数和列数：布局行数*布局列数不得超过 200 为原则，否则会弹出错误提示框。

内边线宽：不超过 100，设置该参数时应考虑布局阵列在画面中拖动的大小，否则会提示布局图素区域范围过小，请调整的提示。

外边线宽：不超过 100，设置该参数时应考虑布局阵列在画面中拖动的大小，否则会提示布局图素区域范围过小，请调整的提示。

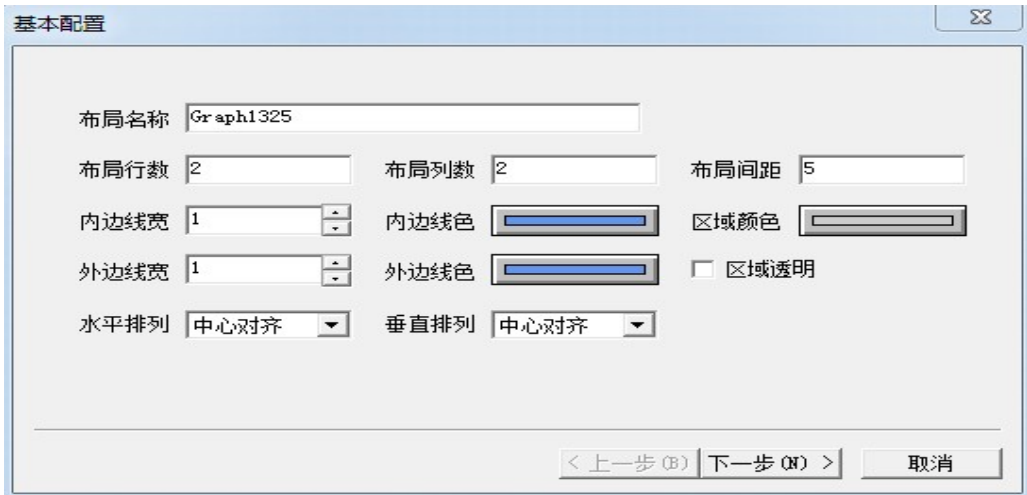


图 27.47 设置布局阵列属性



图 27.48 组态画面

生成图形模板

进入模板管理器，点击工具栏上“新建”图标，或右键菜单选择“新建”项目，打开新建模板类型界面，在界面中选择“图形模板”图标，点击“确定”跳转到图形界面选择界面。

1、选择图形画面，设置模板属性

进入选择图形画面界面，在画面列表中显示当前工程能够生成图形模板的列表中，在此实例中显示上面制作的图形画面 view2，勾选 view2，设置模板的基本属性。

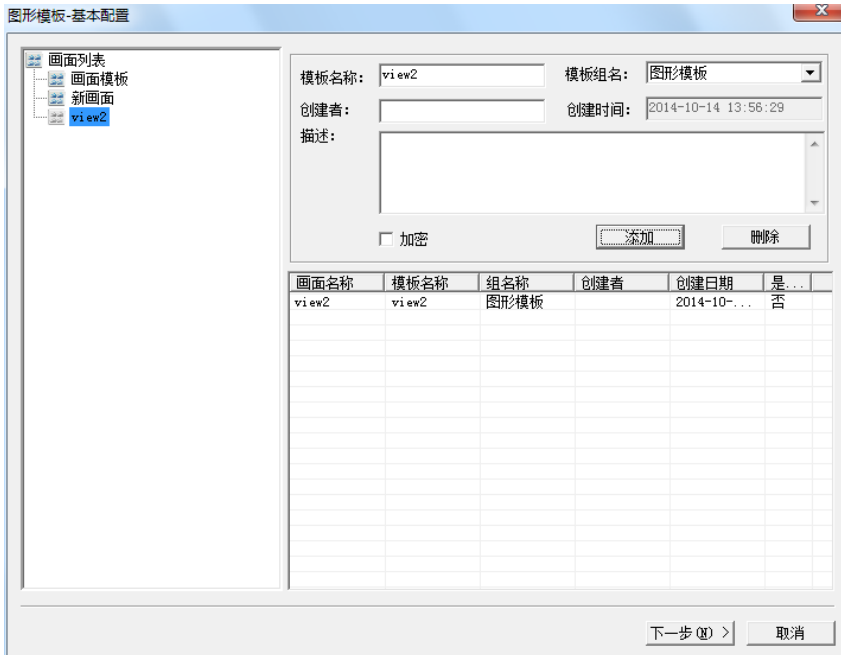


图 27.49 选择图形画面，设置模板属性

- ◇ **模板名称：**模板的标识名称，在同一模板组下不允许同名，可填写字符串信息，默认为画面名称。

- ◇ **模板组:** 模板在模板管理器中的存储位置, 属于哪个分组。
- ◇ **模板安全:** 模板安全, 是模板的操作权限设置, 它包括加密和无限制。
 - 1) 无限制, 不受权限限制, 任何人都可编辑或实例化模板;
 - 2) 加密, 受模板管理器密码保护, 打开时需要验证密码
- ◇ **创建人:** 生成模板时, 用户自行输入标识创建模板或编辑模板的用户信息。
- ◇ **模板描述:** 用户输入的关于模板的文本描述信息, 可以作为此采集模板的一些简单功能说明。
- ◇ **创建时间:** 模板的生成时间, 存储时自动生成, 编辑修改后自动更新。

图形模板基本属性设置完成后, 点击“下一步”跳转到生成模板缩略图, 可为选择的每个图形模板生成一个缩略图。

2、浏览生成图形模板的缩略图

进入模板界面缩略图浏览界面, 通过点击模板列表中显示图形模板, 可显示图形模板的缩略画面, 只有生成缩略图, 才能够模板管理器中展示。模板的缩略图生成后, 点击“下一步”跳转到画面字符串选择界面。

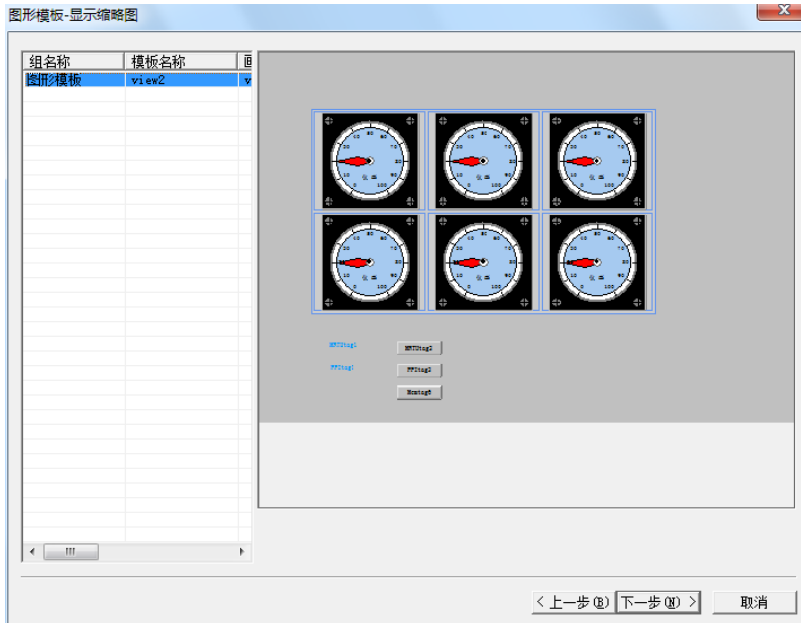


图 27.50 生成缩略图

3、生成图形模板，选择可替换变量

进入选择替换变量界面，列表中显示图形模板中所使用的变量，通过“全选/撤销”按钮可选择所有变量，也可以选择部分变量允许被替换。若变量被选择，在图形模板生成画面时，可以将此变量用其它变量进行替换。变量选择完成后，点击“下一步”跳转到画面字符串选择界面。

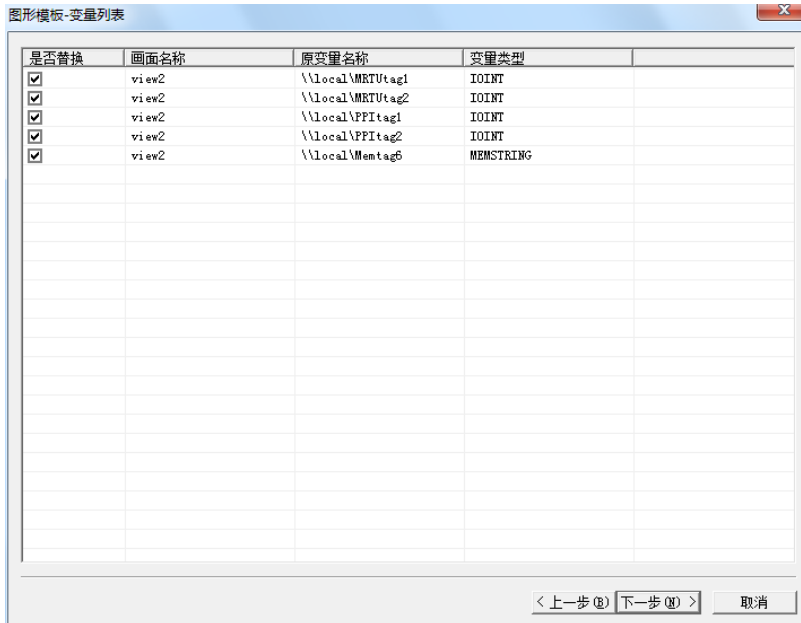


图 27.51 替换变量选择

4、生成图形模板，选择可替换字符串

进入选择替换字符串界面，列表中显示图形模板脚本中所使用的字符串，通过“全选/撤销”按钮可选择所有字符串，也可以选择部分字符串允许被替换。若字符串被选择，在图形模板生成画面时，可以将此字符串用其它字符串进行替换。字符串选择完成后，点击“完成”结束图形模板生成向导，将图形模板存储到模板管理器中。

模板名称	主模板	模板类型	模板组
dfdsf		图形模板	图形模板
view		图形模板	图形模板
view2		图形模板	图形模板

图 27.53 选择图形模板

在模板选择界面中，可重新选择图形模板，在模板列表中编辑画面的替换名称，若为时空，默认不修改模板中画面的名称。编辑画面名称完成后，点击“下一步”跳转到精灵图设置界面。

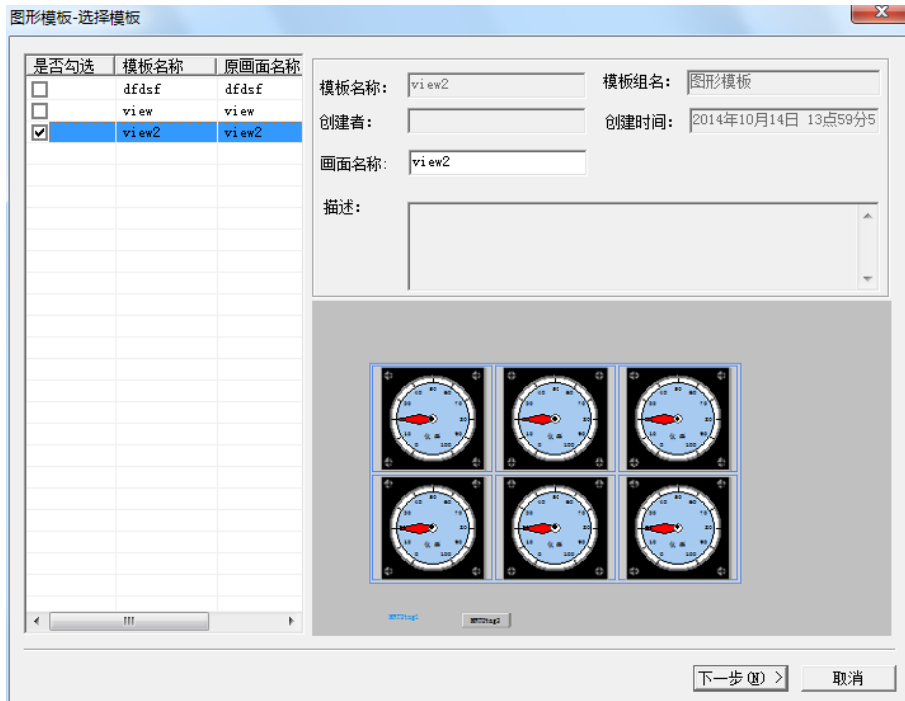


图 27.54 设置模板画面名称

2、设置画面中布局阵列属性，选择精灵图

进入画面精灵布局设置界面，通过切换精灵布局列表，可以查看每个布局的属性，预览画面中精灵的预设布局，在属性显示栏中可以设置生成精灵的个数，并可更换精灵图。

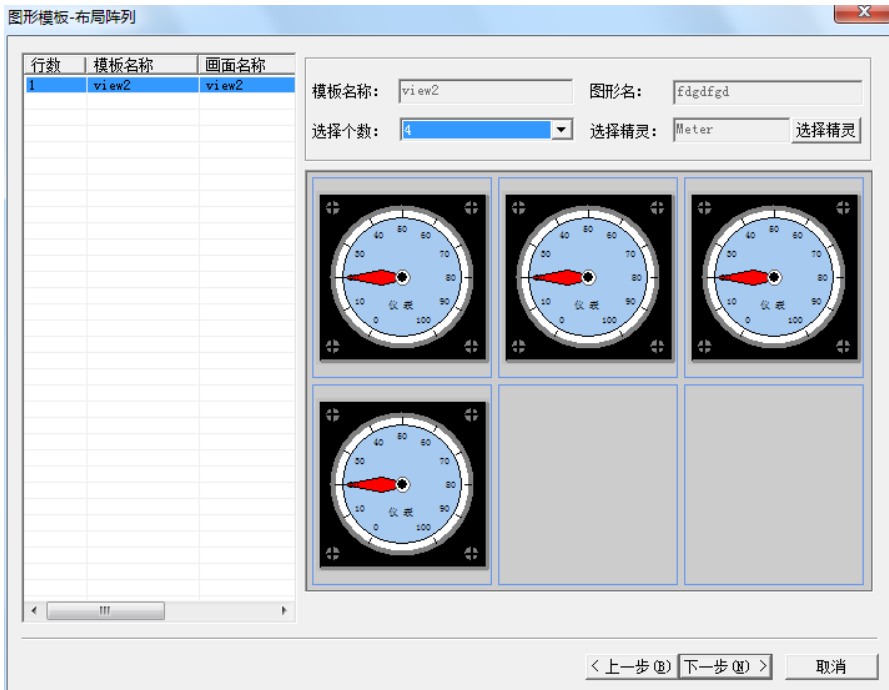


图 27.55 设置精灵图阵列

从图库中选择精灵图。

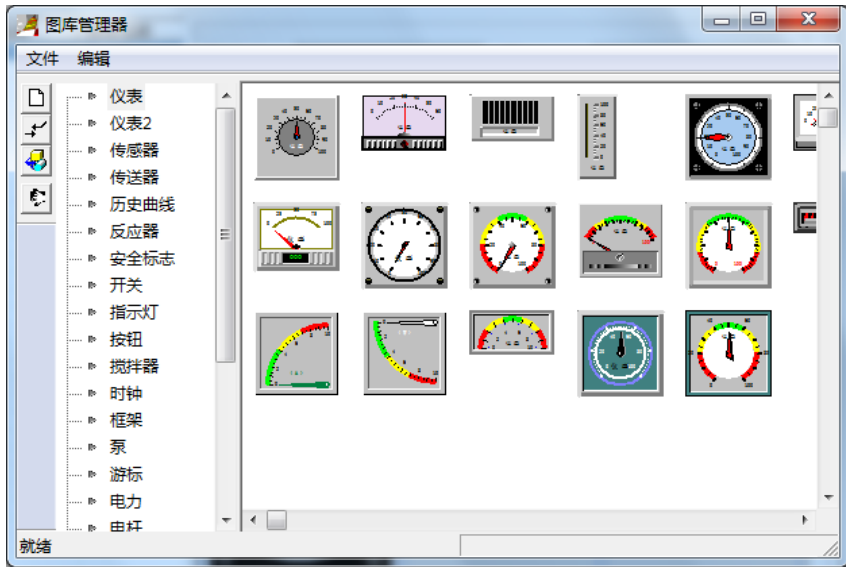


图 27.56 选择替换精灵图

设置完成精灵布局后，点击“下一步”跳转到变量替换界面。

3、替换画面变量

进入变量替换界面，变量列表中显示所有可以被替换的变量，点击替换变量字段，弹出变量选择窗口，可从当前工程中选择变量。替换变量选择完成后，点击“下一步”，跳转页面到字符串替换界面。

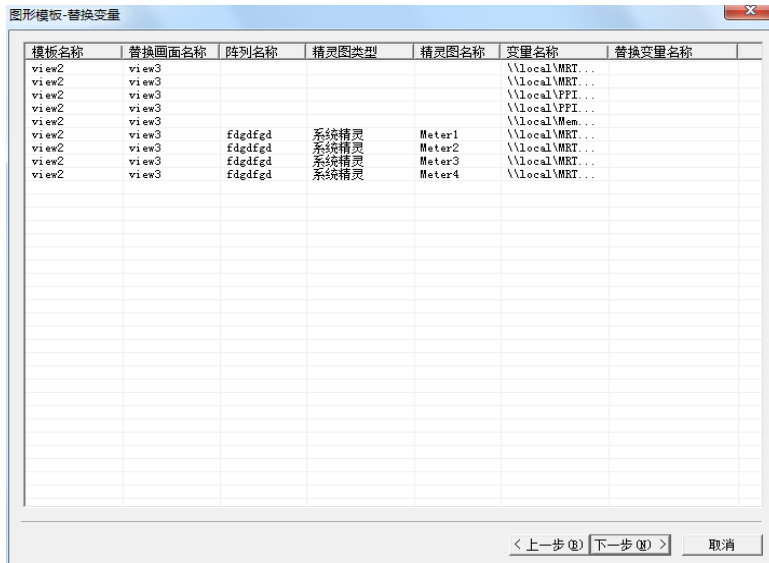


图 27.57 替换画面变量

4、替换字符串

进入字符串替换界面，字符串列表中显示画面脚本中可以被替换的字符串，点击替换字符串，可输入新的字符串信息，替换字符串也可保持为空，不进行替换。字符串替换设置完成后，点击“完成”按钮，将设置的图形模板在当前工程中生成画面。

寄存器名称	寄存器	数据类型	变量类型
模拟量输入	AD <i>dd</i> [.F <i>xLxx</i>][.G <i>xxx</i>]	SHORT	I/O 实型
模拟量输出	DA <i>dd</i>	SHORT	I/O 实型
开关量输入	DI <i>dd</i>	BIT/BYTE/ UNSHORT	I/O 离散、I/O 整型
开关量输出	DO <i>dd</i>	BIT/BYTE/ UNSHORT	I/O 离散、I/O 整型
计数器	TC <i>d</i> [.M <i>x</i>]	SHORT	I/O 整型

说明:

1 AD*dd*[.F*xLxx*][.G*xxx*]中的 *dd* 为通道号,用两位十进制数表示。取值范围请见板卡说明书。单端和伪双端时,AD 通道号取值从 0 到 15。双端时,通道号取值从 0 到 7。例如,AD01 或 AD1 均可表示第 1 通道的模拟量输入; []中的内容为可选项,例如,AD01,AD01.F3L5,AD01.F3L5.G100;

▲ F*x*,表示滤波方式,*x*取值范围为 0 ~ 3,用户选择此项可以消除 A/D 转换中的噪音。板卡的滤波方式有以下几种:

F0: 无滤波

F1: 全程平均,将当前采样值与前几次采样所得值累加,然后取平均值。

F2: 滑动平均,将当前采样值与前几次采样所得值加权累加,然后取平均值。

F3: 取舍平均,与全程类似,但在取平均值前,首先去掉最大值和最小值。

▲ L*xx*,表示平滑时所保存的前几次采样值的个数,*x*取值范围为 5 ~ 10。

▲ G*xxx*,表示板卡实际所取的增益值。

- 2 DA dd , DI dd , DO dd 中的 dd 为通道号, 用两位十进制数表示。取值范围请见板卡说明书。

- 3 TC d [.M x]中的 d , 表示板卡上的计数器的通道号。板卡上有多个计数器时, 按计数器的地址由低到高的顺序确定各个通道号。例如, 板卡上有两个计数器, 每个计数器有三个通道, 则低地址计数器三个通道的通道号为 TC0、TC1 和 TC2, 高地址计数器三个通道的通道号为 TC3、TC4 和 TC5。对于带计数功能的板卡不用在定义设备时写初始化字, 例如定义了 TC2.M2 寄存器即表示了您已选择了计数器 2, 工作方式为 2, 在命令语言的应用程序命令语言中启动时给 TC 计数器赋一个初值, 例如 $tc0=65535$; 则第一个脉冲到达后, 计数器从 65535 开始做减计数。

▲ M x , 表示计数器的工作方式, x 取值范围 $0 \sim 5$; 计数器的工作方式有以下六种:

- M0: 表示计数器结束产生中断方式
- M1: 表示可编程单次脉冲方式
- M2: 表示频率发生器
- M3: 表示方波频率发生器
- M4: 表示软件触发选通方式
- M5: 表示硬件触发选通方式

- 4 组态王中原始最大值、原始最小值是根据转换位数确定的。例如, A/D 转换分辨率为 12 位, 则原始最大值定为 4095 (十六进制数 FFF), 原始最小值 0; 最

大值、最小值按实际选择量确定。D/A 转换时，各值的确定同 A/D 转换。

更详细的内容请参见相应的组态王设备帮助。

查找最新的设备帮助请看 Kinview_driver.chm 文件，该文件在组态王安装路径下
(如: C:\Program files\Kingview\Help)

本手册中只列出了部分组态王支持的硬件设备，如果您想了解更多详细内容，请参见组态王驱动在线帮助，或访问亚控网站: www.wellintech.com 的“驱动下载”，或咨询亚控公司驱动技术支持工程师。

如果您需要开发设备驱动程序、定制产品，请与亚控技术支持或销售工程师联系。我们将为您提供及时周到的服务和优质的产品。

可编程控制器 PLC

莫迪康公司 MODBUS 系列 MODBUS PLUS 系列 MICRO37 系列

欧姆龙株式会社 OMRON 系列 (通过 HostLink 或 ControlLink 与串口连接) 欧姆龙 C1kPCI 欧姆龙 以太网

松下公司 FP 和 FP3 系列

三菱公司 MELSEC FX2, FX2n 系列 MELSEC A 和 AnA 系列 FX2 485 系列 三菱 CCLINK 三菱 中型 PLC 以太网 三菱 Q 系列以太网

西门子公司 S5 系列 S7-200 系列 S7-300MPI 系列 S7-300Profibus-FMS 系列 西门子 PROFIBUS-DP

富士公司 FLEX-PC NB 系列 FLEX-PC NJ 系列 NJ-RS2 系列 NJ-RS4 系列 NP1L-ET1

华光公司 SU-6 系列 SG-8 系列 SR-21 系列 LG 公司 Master-K 系列 Master-K-XXXS
系列 Master K30H 系列 CCM 协议 以太网

GE 公司 SERIES 90 系列 GE 系列 SNP-X 协议 GE 系列 9030 以太网

金钟-默勒 PS4-141-MM1 系列 MASTER 系列 MASTERCOM

贝加莱系列 贝加莱 PVI

东芝 EX 系列 罗克韦尔 AB-MicroLogix AB-MicroLogix-DH

罗克韦尔 AB-MicroLogix AB-MicroLogix-DH

TWINCAT TwinCAT TwinCAT 远程控制

横河 横河 F3 横河 F3 以太网

天津贝克电气有限公司 控制器 RK512

智能模块

组态王支持的硬件设备

研华公司 ADAM 4000 系列 ADAM 5000 系列 ADAM 5000(CAN)系列 ADAM 6000 系列
ADAM 7000 系列 （研华全系列模块）

威达公司 NUDAM 6000 系列 NUDAM 7000 系列

中机浦发公司 ORIENT-2000 系列

集讯系列 ISM100

IDC893 系列

Eastfar 系列

长英科技系列

ALPHA900 系列

泓格系列

研祥系列

中泰公司

凌华公司

硕人科技

智能仪表

欧陆 808, 818, 900, 590 系列

理化 REX-D, REX-C, CD901, Farex, REX-F900 系列

导电 SR73 , SR53 , SR25, FP21, MR13 系列

宇光 AI 系列

山武-霍尼韦尔 SDC30/31, SDC40B, SDC40A/40G, SRF100 系列

西屋 WEST 系列

神王 TTM-110 系列

神岗 FC 系列

亚特克 AL808 系列

霍尼韦尔 UDC6300, UMC800, UDC2300 系列

上润系列

百特系列

昆仑海岸系列

金大陆系列

德莱系列

PANTHER

TOLEDO

富士系列

昆仑天辰 XS 系列

昌晖系列 ……

板卡

威达全系列

华远全系列

磐仪全系列

研祥全系列

中泰全系列

研华全系列

康拓全系列

中达-斯米克全系列

金科航全系列

天时系列

超拓系列

众人系列

科日新系列

宏拓系列

康泰克全系列

双诺系列

同维系列

变频器

AB 系列

LENZE 系列

SAMCO 系列

SIEMENS 系列

DANFOSS 系列

DELTA 系列

FUJI 系列

日立系列

佳灵系列

Vacon 系列

明电舍

西门子

捷丰

华为

科尔摩根

科比

安川

LG

.....

组态王支持 GPRS 通讯

随着移动推出 GPRS 无线数据传输以来，GPRS 的通信速度快、通信费用低、组网灵

活等优点，越来越被广大客户看好。GPRS 数传终端，具有 TCP/IP 协议转换功能不需要用户提供 TCP/IP 的支持。可适用于所有带串口的终端设备，通过 GPRS 网络平台实现数据信息的无线和透明传输，为不具备 TCP/IP 协议处理的终端设备提供了 GPRS 通信的能力。

北京亚控公司开发的 GPRS 服务程序支持通过 GPRS 数传终端（以下简称 GPRS DTU）与亚控开发过驱动的串口设备间的通讯。

组态王 7.0 新功能简介

组态王 7.0 产品包括了以下主要新功能：

◇ 组态模板功能

在组态王 7.0 中，提供了模板管理器，包含有画面模板、采集模板、脚本模板和图形模板等几种类型模板。模板管理器统一对模板信息及模板文件进行管理。使用模板功能能够为工程实施和部署带来便利，提高工程组态内容复用程度，降低工程组态开发成本。



◇ 支持多语言操作系统

使用组态王 7.0 中文版开发的工程，可以使用组态王 7.0 英文版或其他语言版本直接打开运行，不再受语言环境影响，减少了为不同语言开发组态工程工作量，降低了工程组态成本。

◇ 开发系统组态工程 XML 格式文件导入导出

XML 语言以其良好的扩展性、规范性、可读性、兼容性和传输性等特点，被广泛应用于企业集成和应用开发。在组态王开发系统中，提供了各个组态配置项目的 XML 格式文件导入导出功能，如画面、变量、脚本、配方、非线性表、系统配置等。这些组态配置项目可分别导出到 XML 格式文件中，根据需要编辑后，可再导入到工程。方便第三方在不使用组态王开发编辑环境的情况下，完成组态配置。

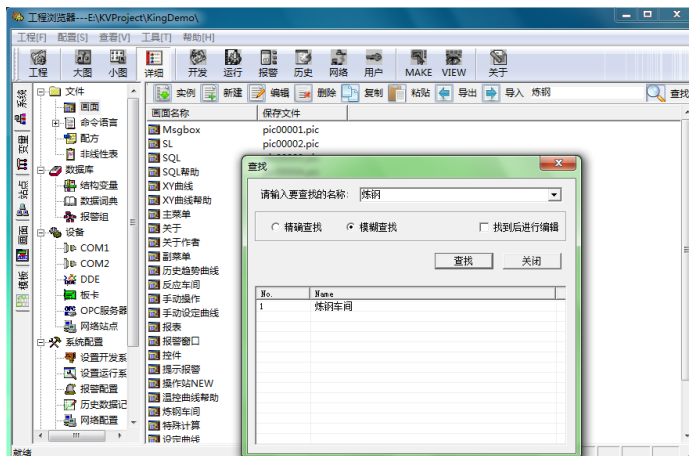
◇ 组态开发环境易用性增强

组态开发环境中，在多个方面增加或改进了功能，提升了易用性。包括：

- 1) 脚本编辑器中增加函数列表，可快速浏览函数使用方法和向编辑器中输入函数。
- 2) 在脚本编辑器中增加了脚本函数自动提示功能，当输入的字符符合函数名称字符

时，自动提示所有类似函数名称，方便快速输入函数。

- 3) 工程浏览器中，针对包括画面、脚本、变量等，增加查找功能，查找后显示符合条件的内容列表，从列表中选中项目后可直接转入编辑界面。
- 4) 在工程浏览器中，增加了快速操作工具栏，常用功能均在工具栏中，方便操作。
- 5) 在工程浏览器和变量选择器中，增加了变量分类管理功能。变量可根据类型、内存或 IO、不同设备等几种方式进行分类浏览、管理。

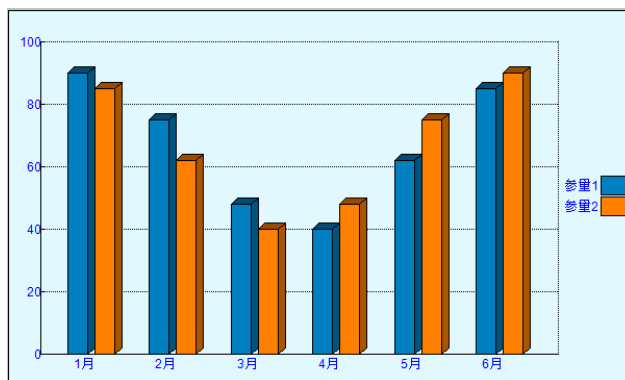


◇ 图库精灵增加

扩展了开关、指示灯、按钮等几种类型的图库精灵，内容更加丰富。

◇ 棒图控件功能扩展

对组态王原内置棒图控件改进，增加了系列功能，可在一个图表中添加多个系列的棒图数据，方便对比分析。



◇ 提供工程升级工具，新的工程升级方式

组态王 7.0 提供独立的工程升级工具，支持组态王 6.55 工程升级，将组态王旧版本工程转换为中间文件，然后再升级到组态王 7.0 中。这样可以防止组态王旧版本工程因升级造成工程文件被改变风险。

◇ 新的授权系统，使用更加方便，提供软授权方式

组态王 7.0 除兼容组态王 6.55 硬件加密锁外，新增加了一套授权体系，包括硬件授权和软授权两种，用户可根据自身需要选择。

新的硬件授权以 USB 设备形式提供。软授权是与申请授权的目标计算机绑定的，不可迁移。

新授权，无论硬件加密锁或软授权，可实现不与组态王安装在同一台计算机上的限制，方便授权的部署使用。

新的授权，均可实现远程升级。

⚠ 在使用组态王 7.0 时，需要注意以下方面：

- ✚ 目前工程升级工具只能升级组态王 655 中文版工程。
- ✚ 原组态王中内置 XY 曲线控件，在组态王 7 中暂不支持。
- ✚ 原组态王中内置棒图控件，在组态王 7 中功能增加，控件函数与原来不同，升级工程后，需要将脚本中控件函数的使用按照新函数进行编辑后方可使用。
- ✚ 组态王 Html 版本的 WEB 功能在组态王 7.0 中暂不支持。

KV7 新授权使用说明：

1、Kv7 加密锁特性

- 将启用软授权，为不方便使用硬件锁的工程应用提供便利；
- 将实现授权锁可与组态王应用不在同一计算机的方式；
- 可远程升级，降低授权成本；
- 兼容已有锁；

2、Kv 新授权分类

- 按软件保护方式分，有硬件授权和软授权；
 - 硬件授权：主要是指加密锁形式的授权，包括并口锁和 USB 锁。
 - 软授权：主要是指软件形式的授权，一般采用与计算机硬件特征绑定的电子许可证形式的加密技术，它依然需要使用加密算法作为主要的保护软件的技术手段。
- 按照加密锁功能本身还分为单机锁与网络锁：
 - 单机锁：只能够同时在一台计算机上使用。
 - 网络锁：授权用户数量，可以允许网络上面的其他计算机使用，同时使用的用户数量不能够超过锁的授权用户数量。
- 按照组态王使用时限分为试用版授权和正式版授权，其中正式版授权又分为限时授权和无限时授权。
 - 试用授权：针对亚控公司发行的软件试用版本，一次性制作完成含固定期限的授权文件（demo*.V2C），并将其放入产品安装包中提供给用户试用，产品软件在试用期间、“试用参数范围”内正常运行，有试用提示，在试用期间过后不能正常运行；
 - 限时授权：针对亚控公司发行的软件正式版本，根据用户需求制作授权文件，并将其提供给授权用户使用，产品软件在限时期间、“授权参数范围”内正常运行，有限时提示，在限时期间过后不能正常运行；
 - 无限时授权：针对亚控公司发行的软件正式版本，根据用户需求制作授权文件，并将其提供给授权用户使用，产品软件在“授权参数范围”内正常运行。

3、Kv7.0 新加密锁驱动安装

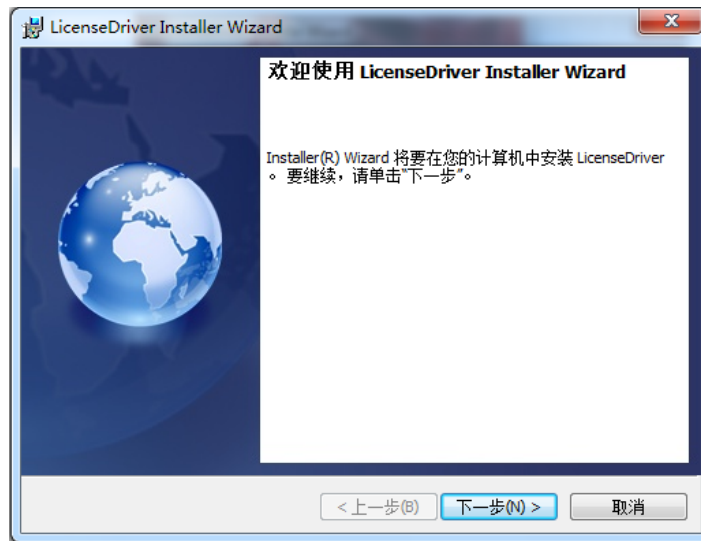
在产品安装包中针对 kv 新授权包含 LinceseDriver 文件夹，其中包含：

- “授权服务\驱动安装文件 haspdinst_OAIKD.exe”；
- “授权激活升级迁移工具文件 RUS_OAIKD.exe”；

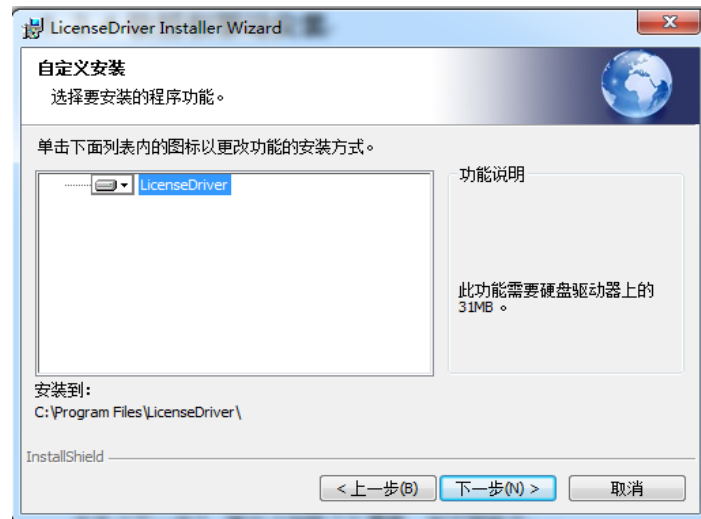
➤ “授权查看工具文件 LicenceViewer_CN.exe”；

安装过程如下：

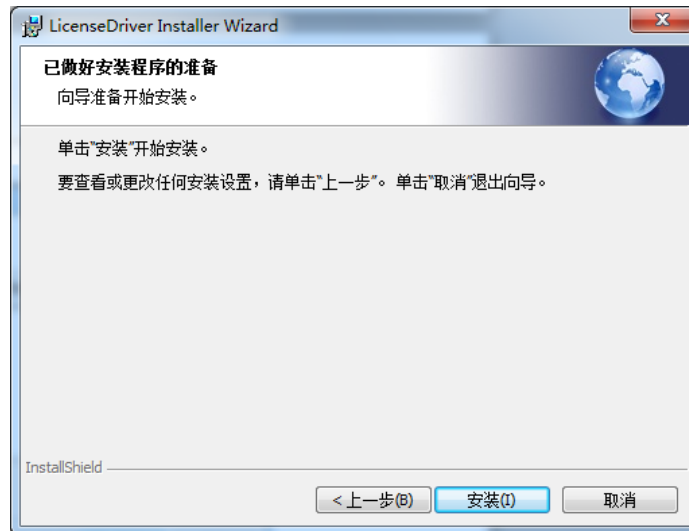
点击“KV7.0 加密锁驱动安装”项，弹出“向导 1”界面，如下图所示：



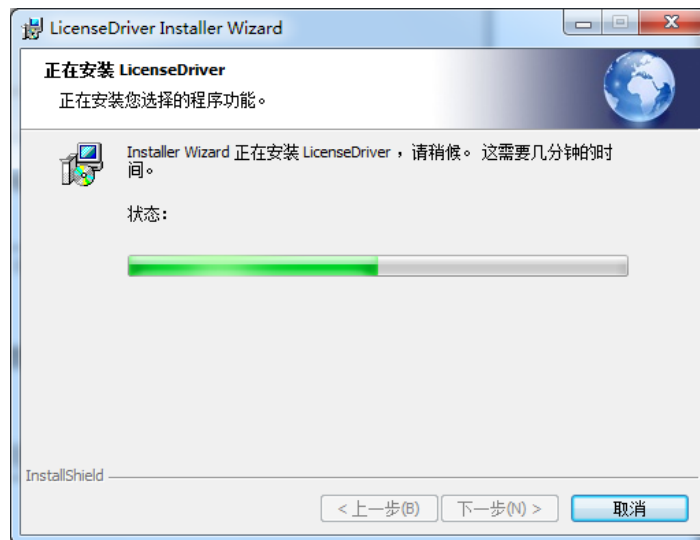
点击“下一步”按钮，弹出“向导 2”界面，如下图所示：



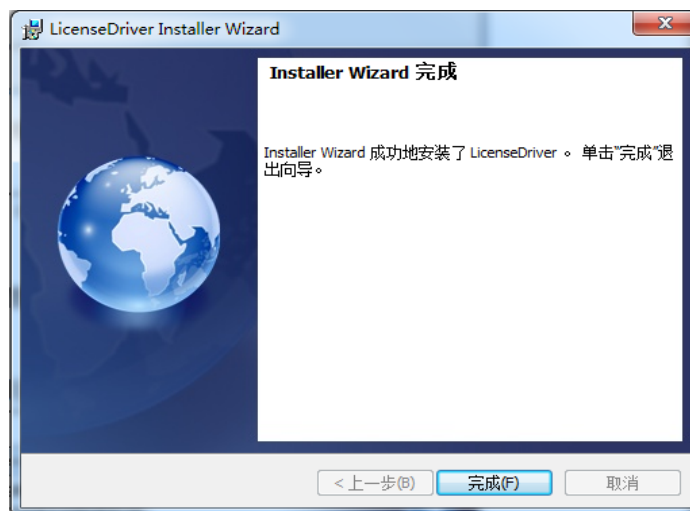
点击“下一步”按钮，弹出“向导 3”界面，如下图所示：



点击“安装”按钮，执行安装。

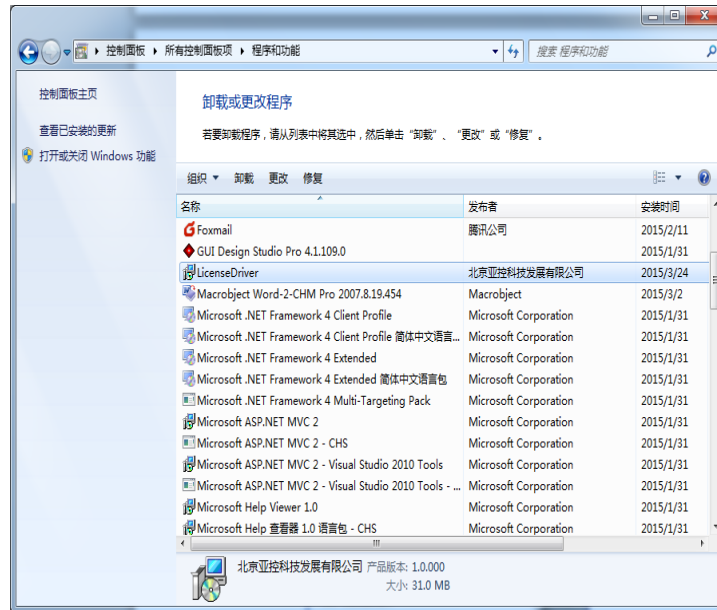


安装完成后，点击“完成”按钮，则退出当前安装。

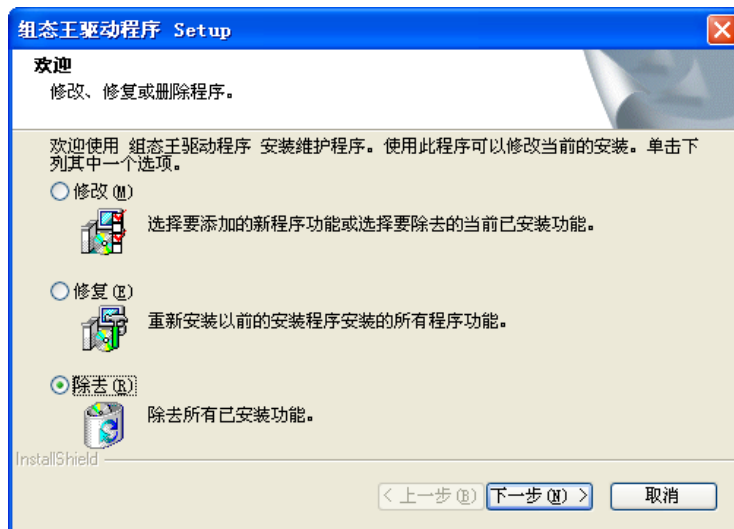


4、Kv7.0 新加密锁驱动卸载

第一步，选择 windows 的“开始菜单”，选择“设置” — “控制面板”，进入“控制面板”界面，选择“添加/删除程序”如图所示：



第二步，在上图中选择 LicenseDriver 程序，单击“更改/删除”按钮，系统弹出如图 1.28 所示对话框。选择“除去”选项，系统会自动删除安装的项目。



5、如何使用授权

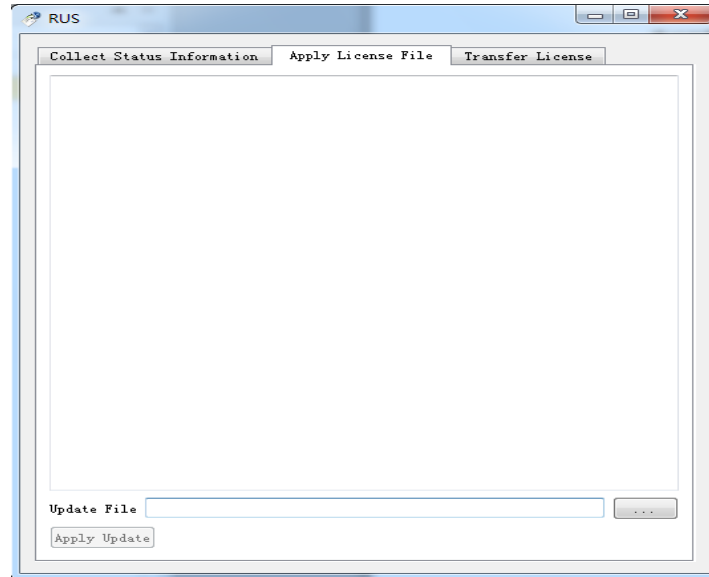
1、硬授权锁使用：

用户在需要授权的机器上安装产品安装包中“授权相关的安装”，取得硬授权锁后，将硬授权锁插在授权机器上。

2、软授权使用：

步骤 1：亚控将设置好授权参数的授权文件*.V2C 发给用户。

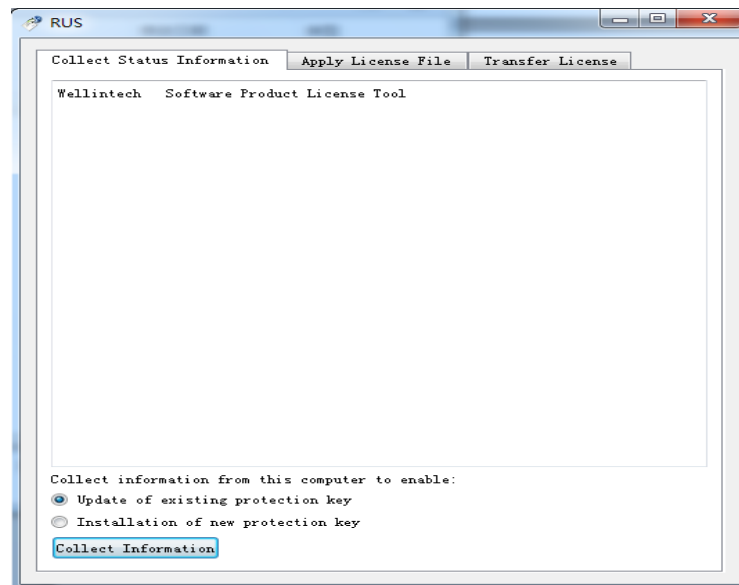
步骤 2: 用户通过安装盘中提供的“授权激活升级迁移工具文件 RUS_OAIKD.exe”将授权文件*.V2C 文件导入到授权机器上, 即完成授权; 具体做法是双击“授权激活升级迁移工具文件 RUS_OAIKD.exe”, 在“Apply License File”标签项下的“Update File”项, 将亚控提供的授权文件*.V2C 文件导入到授权机器上, 即完成授权, 如下图所示:



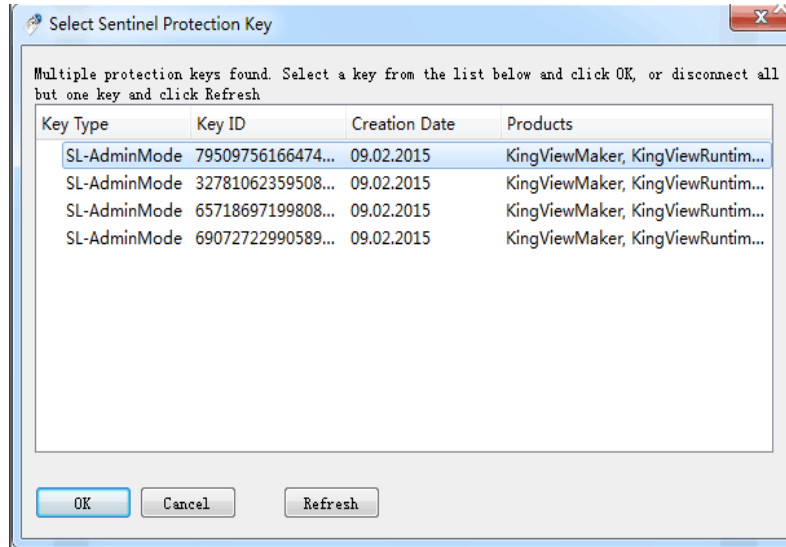
6、如何远程升级

1) 对于硬授权(正式)的升级:

步骤 1: 双击安装盘中提供的“授权激活升级迁移工具文件 RUS_OAIKD.exe”, 弹出如下界面:

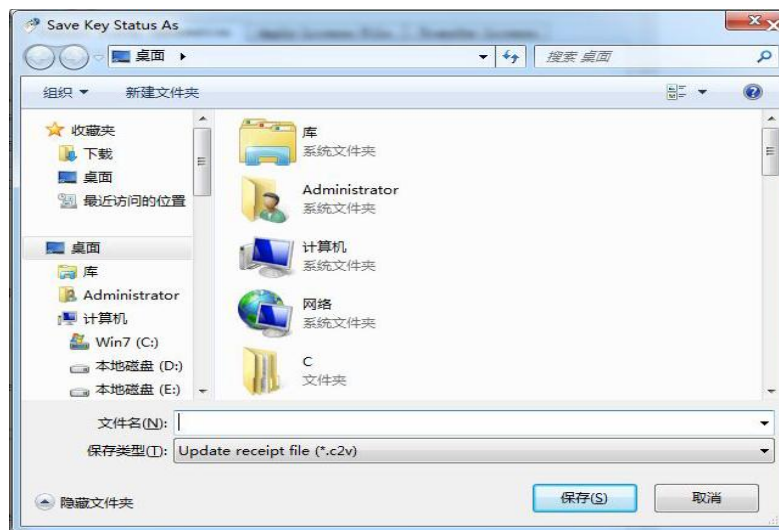


步骤 2: 在“Collect Status Information”标签项下选择“Update of existing Protection Key”, 点击 Collect Information 按钮, 弹出如下所示界面:



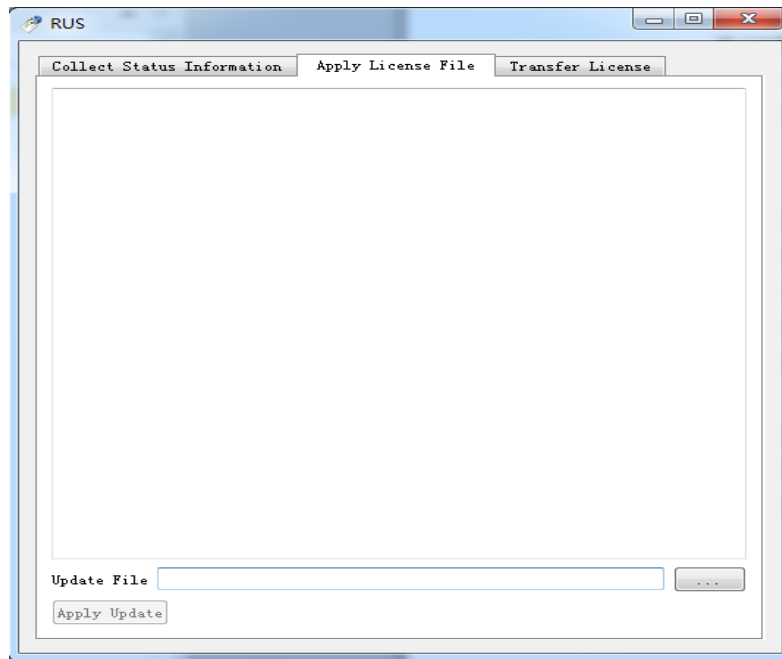
该界面显示了多个加密锁的授权信息。

步骤 3: 选择要升级的加密锁授权信息，点击“ok”按钮，弹出路径选择对话框，如下图所示：



通过该界面，将要升级的加密锁的授权信息保存成.C2V)文件，并将该文件发给亚控公司，亚控升级后会将相应的授权文件*.V2C发给客户。

步骤 4: 双击“授权激活升级迁移工具文件 RUS_OAIKD.exe”，在“Apply License File”标签项下的“Update File”项，将升级后的授权文件*.V2C文件导入到授权机器上，即完成授权升级，如下图所示：



2) 对于软授权（正式）的升级：

步骤 1：用户通过安装盘中提供的“授权激活升级迁移工具文件 RUS_OAIKD.exe”获取已授权机器的授权信息（文件.C2V）并发给亚控，亚控将升级后的授权文件*.V2C 发给客户。详细过程参见上述硬件授权。

步骤 2：用户通过安装盘中提供的“授权激活升级迁移工具文件 RUS_OAIKD.exe”将升级后的授权文件*.V2C 文件导入到授权机器上，即完成授权升级；详细过程参见上述硬件授权。

7、KV 新授权系统的产品范围

- 现行产品 KV7 的中英文通用版；
- KV 部分 OEM 版本；
- 后续 KV7 升级产品；