文件名称：HOLLiAS-C.IEC.SDD.02.变量模块设计文档

文件编号：A11-B14-000060

项目名称：石化版DCS-IEC组态软件开发

项目编号：A0903A-4

物料编码：

版 本 号：A

文件密级：秘密

文件状态：CFC

受控标识：受控

|  |  |
| --- | --- |
| 拟制： | 张居昌 王虹静 2011 年 3 月 23 日 |
| 审核： | 黄宁 2011 年 3 月 23 日 |
| 会签： |  |
|  |  |
| 批准： | 刘金龙 2011 年 3 月 25 日 |

**文件发放范围：**

**修订页**

| 序号 | 版本号 | 修订内容简述 | 拟制/日期 | 审核 | 批准 |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1 介绍 5](#_Toc289238273)

[1.1 目的 5](#_Toc289238274)

[1.2 文档概述 5](#_Toc289238275)

[1.3 术语与缩略语 5](#_Toc289238276)

[2 总体设计 6](#_Toc289238277)

[2.1 主要功能 6](#_Toc289238278)

[2.2 模块划分 6](#_Toc289238279)

[2.3 开发方式 6](#_Toc289238280)

[2.4 数据交互 6](#_Toc289238281)

[2.5 相关类设计 7](#_Toc289238282)

[2.5.1 变量类 7](#_Toc289238283)

[2.5.2 数据类型类 8](#_Toc289238284)

[2.5.3 数据容器类 8](#_Toc289238285)

[2.5.4 文档视图类 10](#_Toc289238286)

[2.5.5 显示控件类 15](#_Toc289238287)

[2.5.6 其他辅助类 15](#_Toc289238288)

[2.5.6.1 标识符检查类 15](#_Toc289238289)

[2.5.6.2 对话框类 16](#_Toc289238290)

[3 主要功能点设计 16](#_Toc289238291)

[3.1 测点变量 16](#_Toc289238292)

[3.1.1 导入测点 16](#_Toc289238293)

[3.1.2 增加测点 21](#_Toc289238294)

[3.1.2.1 在测点变量视图中增加测点 21](#_Toc289238295)

[3.1.2.2 通过增加硬件模块增加测点 24](#_Toc289238296)

[3.1.3 删除测点 26](#_Toc289238297)

[3.1.3.1 在测点变量视图中删除测点 26](#_Toc289238298)

[3.1.3.2 通过删除硬件模块删除测点 29](#_Toc289238299)

[3.1.4 修改测点 31](#_Toc289238300)

[3.1.4.1 在测点变量视图中修改 31](#_Toc289238301)

[3.1.4.2 在详细对话框中修改 33](#_Toc289238302)

[3.1.4.3 通过修改硬件模块设备地址修改 35](#_Toc289238303)

[3.1.5 转移测点 38](#_Toc289238304)

[3.1.6 保存测点 40](#_Toc289238305)

[3.1.6.1 保存过程 40](#_Toc289238306)

[3.1.6.2 保存数据格式 41](#_Toc289238307)

[3.1.7 测点名唯一性处理 43](#_Toc289238308)

[3.2 内部变量 43](#_Toc289238309)

[3.2.1 增加变量 44](#_Toc289238310)

[3.2.2 修改变量 46](#_Toc289238311)

[3.2.3 插入变量 49](#_Toc289238312)

[3.2.4 删除变量 50](#_Toc289238313)

[3.3 网络变量 52](#_Toc289238314)

[3.3.1 生成网络变量请求表 53](#_Toc289238315)

[3.3.2 生成网络变量资源表 55](#_Toc289238316)

[3.3.3 修改网络变量 57](#_Toc289238317)

[3.3.4 删除网络变量 59](#_Toc289238318)

[3.4 POU局部变量 59](#_Toc289238319)

[3.4.1 PRG POU变量 59](#_Toc289238320)

[3.4.1.1 增加变量 60](#_Toc289238321)

[3.4.1.2 插入变量 62](#_Toc289238322)

[3.4.1.3 删除变量 62](#_Toc289238323)

[3.4.1.4 修改变量属性 63](#_Toc289238324)

[3.4.2 FB POU变量 64](#_Toc289238325)

[3.4.2.1 页面初始化 64](#_Toc289238326)

[3.4.2.2 检查递归的调用环 64](#_Toc289238327)

[3.4.2.3 个数限制 66](#_Toc289238328)

[3.4.2.4 只读属性 66](#_Toc289238329)

[3.4.2.5 增加变量 66](#_Toc289238330)

[3.4.2.6 插入变量 68](#_Toc289238331)

[3.4.2.7 删除变量 68](#_Toc289238332)

[3.4.2.8 修改变量属性 69](#_Toc289238333)

[3.4.2.9 辅助编辑功能 72](#_Toc289238334)

[3.4.3 FUN POU变量 74](#_Toc289238335)

[3.5 自定义数据类型 75](#_Toc289238336)

[3.5.1 结构体 75](#_Toc289238337)

[3.5.1.1 页面初始化 75](#_Toc289238338)

[3.5.1.2 增加成员 76](#_Toc289238339)

[3.5.1.3 插入成员 77](#_Toc289238340)

[3.5.1.4 删除成员 77](#_Toc289238341)

[3.5.1.5 修改成员属性 78](#_Toc289238342)

[3.5.1.6 辅助的编辑功能 80](#_Toc289238343)

[3.5.2 枚举 82](#_Toc289238344)

[3.5.2.1 页面初始化 82](#_Toc289238345)

[3.5.2.2 增加成员 82](#_Toc289238346)

[3.5.2.3 插入成员 84](#_Toc289238347)

[3.5.2.4 删除成员 84](#_Toc289238348)

[3.5.2.5 修改成员属性 86](#_Toc289238349)

[3.5.2.6 辅助的编辑功能 87](#_Toc289238350)

[3.6 变量自动声明 88](#_Toc289238351)

[3.6.1 定义新变量 89](#_Toc289238352)

[3.6.2 修改变量属性 92](#_Toc289238353)

[3.7 类型变化及库变化 94](#_Toc289238354)

[3.8 在线部分 102](#_Toc289238355)

[3.8.1 内部变量在线 102](#_Toc289238356)

[3.8.2 测点变量在线 103](#_Toc289238357)

[3.8.3 FB POU、FUN POU变量在线 105](#_Toc289238358)

[3.8.4 PRG POU变量在线 105](#_Toc289238359)

[3.8.5 详细对话框中变量在线 105](#_Toc289238360)

[4 附录 106](#_Toc289238361)

[4.1 支持类型 106](#_Toc289238362)

[4.2 变量属性设置 106](#_Toc289238363)

[4.3 相关文件及内容 107](#_Toc289238364)

[4.4 相关表格及内容 108](#_Toc289238365)

# 介绍

## 目的

本文档对变量定义模块结构与功能划分进行了描述，对该模块与相关模块间接口进行说明，编写此文档的目的是明确变量定义模块与其他软件的主要接口，明确整个系统的功能框架和相关数据结构。

本文档是进行后续软件设计、开发与测试依据。

本文档的预期读者是开发设计人员、测试人员。

## 文档概述

该系统是石化版DCS-IEC组态软件开发项目中变量定义模块的设计文档，本文档描述内容在系统结构中的位置如深色背景所描述模块。



1. 图 ‑1：系统结构体图

## 术语与缩略语

LD: Ladder Diagram

CFC: Continuous Function Chart

ST: Structured Text

Function: 函数

Function Block : 功能块

POU: 程序组织单元

# 总体设计

## 主要功能

实现组态软件中变量定义及类型定义功能。

## 模块划分

变量定义模块按功能主要分成一下几部分，如下图所示：



图 2‑1：模块划分图

## 开发方式

采用dll的形式开发，其中界面相关类放在VarDef.dll中，数据相关类放在AppData.dll中。

## 数据交互

变量定义模块与组态软件其他模块以及工程总控存在数据交互，如下图所示。



图 2‑2：数据交互图

## 相关类设计

### 变量类

不同类型的变量定义了不同的数据结构，变量类包括：

* CBaseDB：派生于Cobject，基础变量类，定义基本类型及枚举类型变量
* CPointerDB：派生于CBaseDB，定义指针类型变量
* CArrayDB：派生于CBaseDB，定义数组类型变量
* CStructDB：派生于CBaseDB，定义结构体类型变量
* CFunctionBlockDB：派生于CBaseDB，定义功能块类型变量

变量类关系如下图所示：



图 2‑3：变量类图

### 数据类型类

数据类型类定义了不同自定义类型的数据结构，自定义的数据类型包括枚举、结构体、功能块，数据类型类主要有：

* CBaseType：派生于CObject，数据类型类的基础类
* CEnumType：派生于CBaseType，定义枚举类型
* CStructType：派生于CBaseType，定义结构体类型
* CFunctionBlockType：派生于CBaseType，定义功能块类型

数据类型类关系如下图所示：



图 2‑4：自定义数据类型类图

### 数据容器类

数据容器类包括变量容器类及类型容器类，

变量包括测点变量（全局变量）、内部变量、网络变量及局部变量。各种变量的作用域分别为：

* 测点变量（全局变量）：整个工程（所有控制站）中全局有效
* 内部变量：控制站内部全局有效
* 网络变量：控制站内部全局有效
* 局部变量：POU内部有效，包括PRG POU、FB POU及FUN POU中的局部变量

不同类型的变量存放于不同的数据容器中，变量容器类主要有：

* CGlobalVarContainer：派生于CObject，记录一组测点变量（本控制站内的）、内部变量、网络变量、PRG POU中的变量以及对这几种变量的各种操作
* CLocalVarContainer：派生于CObject，记录一个FB POU、FUN POU中的局部变量以及对局部变量的各种操作。CLocalVarContainer与对应的CPOU相关联
* CDBContainer：派生于CObject，记录CGlobalVarContainer的容器。测点变量、内部变量、网络变量、PRG POU中的变量以分组（CGlobalVarContainer）的形式记录于此容器中
* CPOU：派生于CObject，与CLocalVarContainer容器关联
* CPOUContainer：派生于CObject，POU容器，存放工程中所有的POU，由于CPOU与CLocalVarContainer一一对应，CLocalVarContainer以分组的形式记录与此容器中

工程中自定义的类型记录于自定义类型容器（CUserTypeContainer）中。

* CUserTypeContainer：派生于CObject，自定义类型容器，记录自定义的枚举、结构体、功能块等类型

数据容器类关系如下图所示：





图 2‑4：数据容器类图

### 文档视图类

文档视图类主要负责文档存储和视图显示，并在视图内实现相关的编辑操作。主要包括：全局变量文档视图、局部变量视图、网络变量文档视图、结构体类型文档视图、枚举类型文档视图、测点变量文档视图。

全局变量文档视图类如下图所示，包括：普通全局变量和库中的全局变量，它们共用一个文档类。



图2-5：普通全局变量文档视图类

CFC语言的PRG变量文档视图类如下图所示。



图2-6：CFC语言的PRG变量文档视图类

LD语言的PRG变量文档视图类如下图所示。



图2-7：LD语言的PRG变量文档视图类

ST语言的PRG变量文档视图类如下图所示。

图：图2-8：ST语言的PRG变量文档视图类

局部变量包括：函数中的变量和功能块中的变量，它们共用一个视图类。CFC语言的局部变量文档视图类如下图所示。

图2-9：CFC语言的局部变量文档视图类

LD语言的局部变量文档视图类如下图所示。

图2-10：LD语言的局部变量文档视图类

ST语言的局部变量文档视图类如下图所示。

图2-11：ST语言的局部变量文档视图类

网络变量文档视图类如下图所示。



图2-12：网络变量文档视图类

枚举类型文档视图类如下图所示。



图2-13：枚举类型文档视图类

结构体类型文档视图类如下图所示。



图2-14：结构体类型文档视图类

测点变量文档视图类如下图所示。



### 显示控件类

显示控件类（Grid控件）封装了变量定义表的编辑界面，如下图所示。



图2-15：Grid控件类

控件内部的具体关系如下图所示。



图2-16：Grid控件类内部关系图

### 其他辅助类

#### 标识符检查类

标识符检查类：CCheckIdentifierName检查变量部分的标识符是否符合标识符规则

#### 对话框类

1. 变量自动声明对话框：CDlgAssistant
2. 数组设置对话框: CDlgSetArray
3. 指针设置对话框: CDlgSetPointerDateType
4. 类型管理器: CDlgHelpManager
5. 详细对话框：CDlgVarDetail

# 主要功能点设计

## 测点变量

测点变量（全局变量）在整个工程中全局有效。测点变量在AT测点变量视图中进行编辑，同时记录该操作，在AT保存时，测点变量的操作记录，利用共享内存，发送给总控，实现数据的一致。

AT中对测点变量的编辑，包括导入测点、增加测点（通过右键增加、通过模块增加、通过变量自动声明增加）、修改测点（修改点名，修改非点名项）、插入测点、删除测点、转移测点以及剪切、复制、粘贴、撤销、恢复等辅助功能。



图 3‑1：测点功能图

### 导入测点

导入变量是AT与总控之间数据交互的一部分，完成的功能是：按照约定的数据格式解析变量的各个属性（例如：变量名、变量说明等），并按照解析出来的属性生成新的全局变量。数据库总控将每个站中的测点变量保存到相应的文件中，文件格式如下：

1. 文件格式：

表 3‑1：文件格式

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 类型个数 | char | 1 |
| 类型1信息 |  |  |
| 类型2信息 |  |  |
| …… |  |  |
| 类型n信息 |  |  |

表 3‑2：类型信息格式

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 类型名称 |  | 32 |
| 变量个数 | int | 4 |
| 变量1信息 |  |  |
| 变量2信息 |  |  |
| …… |  |  |
| 变量n信息 |  |  |

表 3‑3：变量信息格式

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 变量名长度 | char | 1 |
| 变量名 | char[] | 变量名长度 |
| 成员个数 | int | 4 |
| 成员1信息 |  |  |
| 成员2信息 |  |  |
| …… |  |  |
| 成员n信息 |  |  |

表 3‑4：成员信息格式

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 成员名长度 | char | 1 |
| 成员名 | char[] | 成员名长度 |
| 成员值长度 | char | 1 |
| 成员值 | char[] | 成员值长度 |

1. 文件名：站号.var
2. 其他的约定

* 每个类型的变量导入同一个变量组，组名即为类型名。
* PN（点名）即为变量名，因此，成员中无需再使用PN。
* 导入某一类型变量时，该类型一定存在，否则该组变量无法导入。
* 当导入过程中某一个变量或某一组变量出现异常时，导入继续进行，顺序解析下一个或下一组变量。
* 由于导入的文件为总控编译通过的文件，因此，除重名检查外，无需对标识符进行其他检查。

主要接口函数有：

1. void CDBContainer::LoadControlVar(char \*pcPath)

参数说明： pcPath 文件路径+文件名

功能说明：导入测点变量总入口

返 回 值：无

函数流程图如下图所示。



图3-2：导入测点变量总入口

1. void CDBContainer::LoadTypeInfo(CFile& VarFile, bool bControl)

参数说明：VarFile 文件指针

bControl 是否为测点变量，true：是，false：不是

功能说明：导入类型信息

返 回 值：无

函数流程图如下图所示。



图3-3：导入类型信息

1. void CDBContainer::LoadVarInfo(CFile& VarFile, CGlobalVarContainer\* &pGlobalVarContainer, CFunctionBlockType \*pFBType)

参数说明：VarFile 文件对象

pGlobalVarContainer 待添加的全局变量容器

pFBType 类型指针

功能说明：导入变量信息

返 回 值：无

函数流程图如下图所示。



图3-4：导入变量信息

1. void CDBContainer::LoadMemberInfo(CFile& VarFile,CFunctionBlockDB\* pFB,bool bIstheATvar,CBaseDB\* pATBase,CGlobalVarContainer\* pContainer)

参数说明：VarFile 文件对象

pFB 新创建的变量指针

bIstheATvar 是否与AT中的测点变量重名，true：重名，false：不重名

pATBase 重名时AT中的变量指针

pContainer 待添加的变量容器指针

功能说明：导入变量信息

返 回 值：无

函数流程图如下图所示。

图3-5：导入成员信息

### 增加测点

增加测点的方式包括在测点变量视图中增加测点、通过增加硬件模块增加测点、通过自动声明增加测点，增加的测点记录于CGlobalVarContainer中。每次增加测点时都要记录相应的操作，保存时，通知总控增加相同的数据。

#### 在测点变量视图中增加测点

在测点变量视图中增加测点的方法是：点击鼠标右键，弹出右键菜单，选择“增加变量”。不能通过增加硬件模块增加的测点都可以通过此方法增加。

这种方法增加测点时主要调用的函数如下：

* void CPinGlobalVarView::OnAddVar()

功能：在测点变量视图中增加测点变量

参数：无

返回值：无

说明：选择“增加变量”时，响应此接口，此接口先向总控请求一个新的点名，然后调用下面一个接口完成增加操作

* bool CPinGlobalVarDoc::AddInst(CString strNewName)

功能：增加测点变量记录

参数：strNewName：新的点名

返回值：true：增加成功，false：增加失败

说明：增加测点记录的总接口，此接口内调用以下两个接口完成测定的增加，然后记录增加操作

* bool CGlobalVarContainer::AddPinRecord(CFunctionBlockType \*pFBType,CString strNewName)

功能：在测点变量容器中增加新的测点记录

参数：pFBType：新的测点类型，strNewName：新的测点名

返回值：true：增加成功，false：增加失败

说明：此接口内先根据类型和测点名构造一个新的测点变量，该变量各引脚默认值在db.dat配置文件中定义，然后将该变量增加到容器的末端，具体流程如下图所示：



图 3‑6：在测点变量容器中增加新的测点记录流程

* bool CDBClassInstSet::AddInst(CGridItemSet \*pGridItemSet,CFunctionBlockDB \*pFBDB)

功能：在测点实例中增加新的测点实例，供显示时使用

参数：pGridItemSet：当前各点项的显示方式，pFBDB：新的测点变量

返回值：true：增加成功，false：增加失败

说明：根据当前的显示方式和新增加的测点变量构造新的测点实例

* void CPinGlobalVarDoc::AddPinOper(CFunctionBlockDB \*pFBDB)

功能：记录增加操作

参数： pFBDB：新的测点变量

返回值：无

说明：根据已定义的数据格式，记录增加测点操作，保存时通知总控增加该测点

* void CPinGlobalVarView::FillToLine(int iRow, CDBClassInst \*pInst,CHWTPSet \*pTPSet/\* = NULL\*/)

功能：在测点变量视图中增加一行新的测点

参数：iRow：新增加的行号，pInst：新的测点实例

返回值：无

说明：先在测点变量视图中的grid控件上新增一行，然后用pInst填充这一行

增加测点变量的流程如下图所示：



图 3‑7： 增加测点变量流程

#### 通过增加硬件模块增加测点

工程中DVI、AVI、AVO、DVO、REALIN、PUI、SOE\_DH、BITIN、REALOUT、BITOUT、WORDOUT、AVI6等类型的测点只能通过增加硬件模块增加。增加硬件模块时，硬件配置记录通道信息，并由框架通知变量增加对应的测点。

框架中定义的消息响应接口为：

char CMainFrame::OnMessageHWChannelInfo(WPARAM wParam, LPARAM lParam)

为实现增加测点操作，该接口首先解析增加硬件模块时记录的数据包，判断是否为增加操作，然后调用以下接口（按调用顺序）：

* char CGlobalVarContainer::HWAddPinRecord(CString strModelName,CString strDNValue, CString strCNValue)

功能：在容器中增加一个测点变量记录

参数：strModelName：模块名，strDNValue：新增加模块的设备地址，strCNValue：新增测点对应的通道号

返回值：0：不能正常与总控通讯，3：不能与服务器端总控通讯，-1：增加失败

说明：硬件模块的每一个通道都对应一个测点变量，该接口增加一个通道对应的的测点变量

一个通道的测点变量增加流程如下图所示：



图 3‑8： 增加一个通道对应的测点变量流程

* void CGlobalVarContainer::AddDevVar(CString strDNValue)

功能：增加诊断变量

参数： strDNValue：新增加模块的设备地址

返回值：无

说明：每增加一个硬件模块都对应增加一个诊断变量，诊断变量的命名规则为“FIO”+站号+‘\_’+设备地址，变量类型为HSGetState，诊断变量属于内部变量，构造的诊断变量记录于内部变量容器中（CGlobalVarContainer

通过硬件模块增加测点的流程如下图所示：



图 3‑9： 通过硬件模块增加测点流程

### 删除测点

删除测点的方式包括在测点变量视图中删除测点、通过删除硬件模块删除测点，每次删除测点时都要记录相应的操作，保存时，通知总控删除对应的数据。

#### 在测点变量视图中删除测点

在测点变量视图中删除测点的方法是：点击鼠标右键，弹出右键菜单，选择“删除变量”。不能通过删除硬件模块删除的测点都可以通过此方法操作。

通过这种方式删除测点主要调用的函数如下（按调用顺序）：

* void CPinGlobalVarView::OnDelVar()

功能：在测点变量视图中删除测点变量

参数：无

返回值：无

说明：选择“删除变量”时，响应此接口，先判断是否有变量处于编辑状态，然后调用下面接口完成删除操作

* bool CPinGlobalVarDoc::DelInsts(CStringArray \*pstrInsts)

功能：在测点记录中删除一组测点

参数：pstrInsts：被删除测点的名字

返回值：true：删除成功，false：删除失败

说明：删除测点记录的总接口，主要负责删除测点实例及测点容器中的记录

* bool CDBClassInstSet::DelInsts(CStringArray \*pstrInsts)

功能：在测点实例记录中删除一组测点实例

参数：pstrInsts：被删除测点的名字

返回值：true：删除成功，false：删除失败

说明：负责删除测点实例，以便更新显示数据

* bool CGlobalVarContainer::DelPinInsts(CStringArray \*pstrInsts)

功能：在测点容器中删除一组测点

参数：pstrInsts：被删除测点的名字

返回值：true：删除成功，false：删除失败

说明：删除测点容器中的一组测点数据，记录删除操作

在测点视图中删除测点的流程如下图所示：



图 3‑10：在测点视图中删除测点流程

#### 通过删除硬件模块删除测点

工程中DVI、AVI、AVO、DVO、REALIN、PUI、SOE\_DH、BITIN、REALOUT、BITOUT、WORDOUT、AVI6等类型的测点只能通过删除硬件模块删除。删除硬件模块时，硬件配置记录操作信息，并由框架通知变量删除对应的测点。

框架中定义的消息响应接口为：

char CMainFrame::OnMessageHWChannelInfo(WPARAM wParam, LPARAM lParam)

为实现删除测点操作，该接口首先解析删除硬件模块时记录的数据包，判断是否为删除操作，然后调用以下接口（按调用顺序）：

* bool CGlobalVarContainer::HWDelPinInst(CString strDNValue).

功能：删除测点变量容器中设备地址为strDNValue的测点变量

参数：strDNValue：设备地址

返回值：true：删除成功，false：删除失败

说明：每个被删除的硬件模块都对应一个设备地址，通过设备地址删除该模块对应的一组测点变量

根据设备地址删除容器中测点变量的流程如下图所示：



图 3‑11：根据设备地址删除容器中测点变量流程

* void CGlobalVarContainer::DelDevVar(CString strDNValue)

功能：根据设备地址删除对应的诊断变量

参数：strDNValue：设备地址

返回值：无

说明：从诊断变量容器中删除对应的诊断变量

通过删除硬件模块删除测点的流程如下图所示：



图 3‑12：通过删除硬件模块删除测点流程

### 修改测点

测点变量各点项值的有效性判断是由配置文件（db.dat）定义的，该文件中定义的点项之间的约束关系如下：

* SN+DN+CN 唯一
* AREANO（区号）取值范围[0,31]
* 报警限制：MD<AL3<LL<AL<AH<HH<AH3<MU
* TP（信号类型）限制：不同模块对应的TP值范围不同，由ModuleChannel.ini配置文件决定
* OT（输出上限）> OB（输出下限）
* PU（PV的量程上限）> PD（PV的量程下限）
* FN（故障信息）取值范围[0,255]
* CE（指令信息）取值范围[0,255]
* SE（状态信息）取值范围[0,255]
* ZT（运行状态）取值范围[0,255]
* SU（给定上限幅） > SD（给定下限幅）

测点变量DN项是不可编辑的，只能通过修改硬件模块的设备地址来修改对应测点变量的DN项，其它可编辑项可以在测点变量视图中修改。测点变量的修改可分为在测点变量视图中修改、在详细对话框中修改、通过修改硬件模块设备地址修改。

#### 在测点变量视图中修改

测点变量可编辑的项都可以在测点变量视图中修改，修改过程中主要调用的函数如下（按调用顺序）：

* void CPinGlobalVarView::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

功能：修改前保存旧值

参数：

返回值：无

说明：在修改点项之前保存修改前的值，用于修改失败后恢复旧值

* void CPinGlobalVarView::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

功能：修改项值后响应的总接口

参数：

返回值：无

说明：修改项值后响应此接口，该接口首先根据被修改变量的行号得到被修改变量的名字，然后调用下面的接口完成修改的有效性判断

* bool CPinGlobalVarDoc::EditItem(CString strOldValue, CString strPN, CString strItem,CString &strValue)

功能：判断项值是否有效

参数： strOldValue：旧值，strPN：被修改变量的名，strItem：被修改的点项名，strValue：修改值

返回值：true：修改成功，false：修改失败

说明：判断新值是否有效的总接口，调用以下两个接口判断实现

* bool CDBClassInst::CheckNewValueValidity(CString strItemName, CString &strNewValue)

功能：根据配置文件中定义的约束判断项值是否有效

参数： strItemName被修改的点项名，strNewValue：修改值

返回值：true：修改成功，false：修改失败

说明：修改的是PN项，则请求总控判断新的点名是否有效；修改项未在配置文件中定义约束，则判断新值是否和类型、长度匹配；修改项在配置文件中定义了约束，则根据约束检查新值的有效性

* bool CGlobalVarContainer::EditPinItem( CString strPN, CString strItem,CString strValue)

功能：修改测点变量点项的值

参数： strPN被修改测点的名字，strItem：被修改的点项名，strValue：修改值

返回值：true：修改成功，false：修改失败

说明：修改测点变量的点项值

* void CAppGlobalFunc::ModifyTestPinItemOper(CString strPN, CString strItem, CString strValue)

功能：记录修改操作

参数： strPN：被修改测点的名字，strItem：被修改的点项名，strValue：修改值

返回值：无

说明：记录修改操作，保存时用于与总控通讯

在测点变量视图中修改点项的流程如下图所示：



图 3‑13：在测点变量视图中修改点项流程

#### 在详细对话框中修改

双击测点变量，打开其详细对话框，在此对话框中可以修改点项的值。修改过程主要调用的函数如下（按调用顺序）：

* void CDlgVarDetail::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

功能：修改前保存旧值

参数：

返回值：无

说明：在修改点项之前保存修改前的值，用于修改失败后恢复旧值

* void CDlgVarDetail::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

功能：修改项值后响应的总接口

参数：

返回值：无

说明：修改项值后响应此接口，调用下面的接口完成修改的有效性判断

* bool CFunctionBlockDB::CheckTestPinValue(CString strPInName, CString &strNewValue)

功能：检查新值的有效性

参数： strPInName：被修改的点项名，strNewValue：修改值

返回值：true：修改成功，false：修改失败

说明：修改项未在配置文件中定义约束，则判断新值是否和类型匹配；修改项在配置文件中定义了约束，则根据约束检查新值的有效性

* void CFunctionBlockDB::ModifyTestPinItemOper(CString strPN, CString strItem, CString strValue)

功能：记录修改操作

参数： strPN：被修改测点的名字，strItem：被修改的点项名，strValue：修改值

返回值：无

说明：记录修改操作，保存时用于与总控通讯

* void CPinGlobalVarView::FillToLine(int iRow, CFunctionBlockDB \*pFBDB)

功能：重新填写测点视图中对应的变量各点项值

参数： iRow：被修改变量的位置，pFBDB：修改后的测点

返回值：无

说明：在详细对话框中对点项修改后，需要通知测点变量视图同步更新显示的值

在详细对话框中修改测点流程如下图所示：



图 3‑14：在详细对话框中修改测点流程

#### 通过修改硬件模块设备地址修改

DVI、AVI、AVO、DVO、REALIN、PUI、SOE\_DH、BITIN、REALOUT、BITOUT、WORDOUT、AVI6等类型的测点DN项是不可编辑的，在详细对话框中也不允许修改，但是修改模块设备地址时会修改该项的值。修改硬件模块设备地址时，硬件配置记录相应的数据，并由框架通知变量修改对应测点DN项的值。

框架中定义的消息响应接口为：

char CMainFrame::OnMessageHWChannelInfo(WPARAM wParam, LPARAM lParam)

为实现修改测点DN项操作，该接口首先解析硬件模块时记录的数据包，判断是否为修改操作，然后调用以下函数（按调用顺序）：

* bool CGlobalVarContainer::HWEditDN(CString strOldDNValue, CString strNewDNValue)

功能：修改容器中测点变量的设备地址

参数：strOldDNValue：被修改的设备地址，strNewDNValue：新的设备地址

返回值：true：修改成功，false：修改失败

说明：在容器中找设备地址为strOldDNValue的测点，修改其DN项值为strNewDNValue

修改容器中测点变量的设备地址流程如下图所示：



图 3‑15：修改容器中测点变量设备地址流程

* void CGlobalVarContainer::DelDevVar(CString strDNValue)

功能：删除旧设备地址对应的诊断变量

参数：strDNValue：旧的设备地址

返回值：无

说明：修改设备地址时，先删除旧设备地址对应的诊断变量，再利用新的设备地址生成新的诊断变量

* void CGlobalVarContainer::AddDevVar(CString strDNValue)

功能：根据新的设备地址增加新的诊断变量

参数：strDNValue：新的设备地址

返回值：无

说明：修改设备地址时，先删除旧设备地址对应的诊断变量，再利用新的设备地址生成新的诊断变量

通过修改模块设备地址修改变量DN项值的流程如下图所示：



图 3‑16：通过修改模块设备地址修改变量DN项值流程

### 转移测点

一些不同类型的测点是可以相互转移的，如AVI中的点可以全部或一部分转移到AVI6中，反之亦可以，可以发生转移的测点类型定义在配置文件（Transfer.ini）中。

实现测点类型转移的方法：在测点视图中，选中待转移的测点，点击鼠标右键，弹出右键菜单，选择“转移”，即可将选中的测点转移到对应的类型中。

转移过程中主要调用的函数如下（按调用顺序）：

* void CPinGlobalVarView::OnTransfer()

功能：转移测点变量

参数：无

返回值：无

说明：选择“转移”时，响应此接口，首先判断是否有变量是否处于编辑状态，然后收集被选中的变量的名字，再调用下面的接口实现转移

* bool CPinGlobalVarDoc::TransferInsts(CStringArray \*pstrInsts)

功能：转移测点变量

参数：pstrInsts：记录被转移测点变量的名字

返回值：true：转移成功，false：转移失败

说明：该接口首先根据被转移的类型找到测点容器，然后调用下面的接口实现转移功能

* bool CGlobalVarContainer::TransferPinInsts(CStringArray \*pstrInsts,CList<CBaseDB\*,CBaseDB\*> \*pToFBList)

功能：在变量容器中实现测点的转移

参数：pstrInsts：记录被转移测点变量的名字，pToFBList：转移后的测点记录

返回值：true：转移成功，false：转移失败

说明：首先根据当前的测点类型找到转移到的类型，然后根据测点名收集被转移的变量，修改该变量的类型、offset等属性，将其添加到转移到的类型容器中，并在当前容器中删除记录。具体流程如下：



图 3‑17：在变量容器中实现测点的转移流程

转移测点的流程如下图所示：



图 3‑18：转移测点流程

### 保存测点

为了保证AT与总控中的测点数据一致，AT中对测点增加、修改、删除、转移等操作时都会记录相应的操作数据于操作链表中，保存时，将这些操作数据放到一块共享内存，通知总控取数据、解析完成相应的操作。

#### 保存过程

保存测点时调用的接口为：

* void CAutoThinkApp::SendVarMessage()

功能：将AT中对测点的操作数据放到共享内存，通知总控读数据

参数： 无

返回值：无

说明：保存时调用此接口完成与总控的通讯

实现流程如下图所示：



图 3‑19：保存测点流程

#### 保存数据格式

由AT端对某个站的测点进行编辑，保存时将所有操作一次发到总控，使总控中的数据与AT中的数据保持一致。消息总结构如下：

4个字节的操作个数+每一个操作的数据

AT保存时向总控发送数据，数据格式如下：

表 3‑1：向总控发送数据格式

|  |  |  |
| --- | --- | --- |
| 增加测点数据格式 | | |
| 标记 | 1Byte | 值：1 |
| 点名 | 32Byte | 值：点名 |
| 点类型 | 32Byte | 值：类型名 |
| 站号 | 1Byte | 值：站号 |
| 修改点名数据格式 | | |
| 标记 | 1Byte | 值：2 |
| 原点名 | 32Byte | 值：原点名 |
| 新点名 | 32Byte | 值：新点名 |
| 修改点项（非点名项）数据格式 | | |
| 标记 | 1Byte | 值：3 |
| 点名 | 32Byte | 值：点名 |
| 项名 | 16Byte | 值：项名 |
| 项长度 | 1Byte | 值：项值的长度 |
| 项的值 | 项长度个Byte | 值：项的值（以字符串形式） |
| 删除测点数据格式 | | |
| 标记 | 1Byte | 值：4 |
| 点名 | 32Byte | 值：点名 |
| 插入测点数据格式 | | |
| 标记 | 1Byte | 值：6 |
| 点名 | 32Byte | 值：点名 |
| 点类型 | 32Byte | 值：类型名 |
| 站号 | 1Byte | 值：站号 |
| 该位置原来点名 | 32Byte | 值：点名 |
| 增加通道点（通过增加模块增加测点）数据格式 | | |
| 标记 | 1Byte | 值：5 |
| 点名 | 32Byte | 值：点名 |
| 点类型 | 32Byte | 值：类型名 |
| 站号 | 1Byte | 值：站号 |
| 设备号 | 1Byte | 值：模块地址 |
| 通道号 | 1Byte | 值：通道号 |

* AT保存时，将操作数据写入共享内存，然后向总控发送窗口消息ID\_ ID\_DATABASECONFIG（#define ID\_ DATABASECONFIG 62），通知总控接收数据
* 总控收到数据正确处理后，返回1作为正确接收标识
* AT根据返回值判断总控是否已正确接收数据，如果总控正确接收数据，AT清空操作记录，否则，AT保留原来的记录，下次保存时，将未保存的数据再次发给总控。AT工程关闭时，将操作记录序列化

### 测点名唯一性处理

在AT上对测点进行操作时，因无法保证点名在总控工程的唯一性，所以对点名有关的操作都要向总控发送判断消息进行处理来保证点名工程内唯一。

AT向总控发送的数据格式：

表 3‑2：向总控发送数据格式

|  |  |  |
| --- | --- | --- |
| 向总控请求一个点名 | | |
| 标记 | 1Byte | 值：1 |
| 站号 | 1Byte | 值：站号 |
| 类型名 | 32Byte | 值：类型名 |
| 修改点名请求总控判断新的点名是否合法 | | |
| 标记 | 1Byte | 值：2 |
| 原点名 | 32Byte | 值：原点名 |
| 新点名 | 32Byte | 值：新点名 |

* AT增加新点时，向总控发送消息VAR\_ADD\_PIN（#define VAR\_ADD\_PIN 1）,向总控请求一个新的点名
* AT修改点名时，向总控发送消息VAR\_MODIFY\_PIN（#define VAR\_MODIFY\_PIN 2），请求总控判断新点名的合法性
* AT根据总控的返回值判断是否成功，返回值MACS\_CLOSR（#define MACS\_CLOSR -1）标识“不能正常与总控通讯”；返回值SERVER\_MACS\_CLOSR（#define SERVER\_MACS\_CLOSR 3）标识“不能与服务器端总控正常通讯”；返回值PN\_RENAME（#define PN\_RENAME 2）标识“点名已被定义”；返回值PN\_VALID（#define PN\_VALID 1）标识点名有效

## 内部变量

内部变量在控制站内全局有效，在内部变量视图中对内部变量进行编辑。AT中对内部变量的编辑，包括增加变量、修改变量、插入变量、删除变量以及撤销、恢复、剪切、复制、粘贴等辅助功能。



图 3‑20：内部变量功能图

### 增加变量

在内部变量视图中点击鼠标右键，弹出右键菜单，选择“增加变量”，即可实现增加变量操作。

增加变量时，主要调用的函数如下（按调用顺序列出）：

* void CGlobalVarView::OnAddVar()

功能：增加变量

参数：无

返回值：无

说明：选择“增加变量”时，响应此接口

* bool CGlobalVarContainer::AddNormalRecord()

功能：在内部变量容器中增加变量

参数：无

返回值：true：增加变量成功，false：增加变量失败

说明：在内部变量容器中增加变量时，首先获得新的点名，然后构造默认类型（BOOL）变量，最后在容器末端追加新生成的变量，具体流程如下图所示：



图 3‑21：在内部变量容器中增加变量流程

* void CGlobalVarView::AddRecord(CBaseDB \*pBaseDB)

功能：在视图中增加一行变量记录

参数：pBaseDB：新增加的变量

返回值：无

说明：在视图中增加一行，显示新生成的变量

增加变量流程如下图所示：



图 3‑22：增加变量流程

### 修改变量

内部变量各种属性都是可以按照一定的规则进行修改的，通过双击内部变量视图中某一行变量的单元格或点击下拉按钮来输入或选择要修改的值，然后对修改的新值进行合法性检查，如果新值合法则修改成功，否则，修改失败，恢复旧值。

修改变量时，主要调用的函数如下（按调用顺序列出）：

* void CGlobalVarView::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT \*pResul)

功能：保存旧值，修改失败时用于恢复

参数：

返回值：无

说明：双击单元格或点击下拉按钮时响应此接口，用于保存修改前的值，防止修改失败后，恢复旧值

* void CGlobalVarView::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

功能：检查修改值的有效性，保存新值或恢复旧值

参数：

返回值：无

说明：输入新值失去焦点后响应此接口，判断修改的是变量的哪种属性

* bool CGlobalVarContainer::EditKey(CString strOldName, CString strNewName)

功能：检查新变量名是否合法

参数： strOldName：旧变量名，strNewName：新变量名

返回值：true：新名合法，false：新值不合法

说明：修改变量别名属性时，调用此接口，检查新名的合法性，如果合法则修改变量名为新值

* bool CGlobalVarContainer::EditRefName(CString strKey, CString strNewValue)

功能：检查新变量别名是否合法

参数： strKey：被修改变量的名字，strNewValue：新变量别名

返回值：true：新别名合法，false：新别名不合法

说明：修改变量别名属性时，调用此接口，检查新别名的合法性，如果合法则修改变量别名为新值

* bool CGlobalVarContainer::EditComment(CString strKey, CString strNewValue)

功能：检查新的变量说明是否合法

参数： strKey：被修改变量的名字，strNewValue：新变量说明

返回值：true：新变量说明合法，false：新变量说明不合法

说明：修改变量说明属性时，调用此接口，检查新值的合法性，如果合法则修改变量说明为新值

* bool CGlobalVarContainer::EditType(CString strKey, CString strNewValue)

功能：检查新的变量类型是否合法

参数： strKey：被修改变量的名字，strNewValue：新变量类型

返回值：true：新变量类型合法，false：新变量类型不合法

说明：修改变量类型属性时，调用此接口，检查新值的合法性，如果合法则修改变量类型为新值，并修改对应的初始值

* bool CGlobalVarContainer::EditInitValue(CString strKey, CString strNewValue)

功能：检查新的初始值是否合法

参数： strKey：被修改变量的名字，strNewValue：新初始值

返回值：true：新初始值合法，false：新初始值不合法

说明：修改变量初始值属性时，调用此接口，检查新值的合法性，如果合法则修改变量初始值为新值

* bool CGlobalVarContainer::EditNeedNet(CString strKey, CString strNewValue)

功能：检查新的是否上网属性值是否合法

参数： strKey：被修改变量的名字，strNewValue：新的是否上网属性值

返回值：true：新是否上网属性值合法，false：新是否上网属性值不合法

说明：修改变量是否上网属性时，调用此接口，检查新值的合法性，如果合法则修改变量是否上网属性值为新值

修改变量属性流程如下图所示：

图 3‑23：修改变量属性流程

### 插入变量

在内部变量视图中选中一行或多行，点击鼠标右键，弹出右键菜单，选择“插入变量”，即可实现插入变量操作。

插入变量时，主要调用的函数如下（按调用顺序列出）：

* void CGlobalVarView::OnInsertVar()

功能：插入变量

参数：无

返回值：无

说明：选择“插入变量”时，响应此接口

* bool CGlobalVarContainer::InsertNormalRecord(CString strKey,int &iIndex)

功能：在内部变量容器的指定位置插入一个新的变量

参数：strKey：被插入位置变量的名字，iIndex：被插入的位置（输出使用）

返回值：true：插入变量成功，false：插入变量失败

说明：在内部变量容器中增加变量时，首先获得新的点名，然后构造新的变量，计算插入位置，最后在容器的指定位置插入新生成的变量，具体流程如下图所示：



图 3‑24：内部变量容器中增加变量流程

* void CGlobalVarView::FillToLine(int iRow, CBaseDB \*pBaseDB)

功能：在内部变量视图中指定位置插入、显示一行变量

参数：iRow：被插入位置，pBaseDB：被插入的变量

返回值：无

说明：在内部变量视图中，先增加一行，iRow及以下行下移一行，在iRow行位置插入变量pBaseDB

插入变量流程如下图所示：



图 3‑25：插入变量流程

### 删除变量

在内部变量视图中选择一行或多行，点击鼠标右键，弹出右键菜单，选择“删除变量”，即可实现删除变量操作。

删除变量时，主要调用的函数如下（按调用顺序列出）：

* void CGlobalVarView::OnDelVar()

功能：删除变量

参数：无

返回值：无

说明：选择“删除变量”时，响应此接口

* bool CGlobalVarContainer::DelNormalRecords(CStringArray \*pPinNames)

功能：删除指定的一组变量

参数：pPinNames：记录被删除变量的名字

返回值：true：删除成功，false：删除失败

说明：删除变量容器中指定的一组变量

* void DeleteRow(CGridCtrl \*pGrid,CArray<int,int> \*pRowArr)

功能：删除视图中对应的变量记录

参数：pGrid：Grid空间指针，pRowArr：被删除记录的位置

返回值：空

说明：删除grid空间中一组变量记录，同时调整各行位置及序号

删除变量流程如下图示：



图 3‑26：删除变量流程

## 网络变量

网络变量是可以在站间引用的变量。对于某一特定站的AT工程来说，网络变量指被本站引用的其他站的变量。网络变量的主要功能如下图所示。



1. 图3-27 网络变量的主要功能

实现网络变量的站间引用主要依靠编译时生成的网络变量请求表和网络变量资源表。网络变量请求表是.PRG文件的一部分，记录本站引用的网络变量信息；网络变量资源表形成单独一份.NET文件，记录本站可以被其他站引用的变量信息，由此，即可可按照用户输入的域站号和变量名在网络变量资源表中查找到该变量信息进而引用该变量。

### 生成网络变量请求表

网络变量请求表的数据长度为 ulNetVarSize字节。网络变量请求信息的格式如下图所示：



1. 图3-28 网络变量信息格式

其中，StationInfoTag的结构如下图所示：



1. 图3-29 StationInfoTag的数据结构

其中，VarInfoTag的结构如下图所示：



图3-30 VarInfoTag的数据结构

接口函数：void CAppGlobalFunc::CreateNetRequest()

参数说明：无

功能说明：生成网络变量请求表

返 回 值：无

函数流程图如下图所示。



图3-31 创建网络变量请求表

### 生成网络变量资源表

网络变量资源表即NET文件包含三部分信息，分别是文件头、一级哈希表定位信息和变量信息，其中变量信息的组织形式是二级哈希表的形式。其格式如下：

图3-32 NET文件格式说明

1. 文件头：目前文件头包含两部分信息，分别是校验和和文件大小，文件大小包括文件头、一级哈希表定位信息、和全部变量信息的大小。
2. 一级哈希表定位信息：一级哈希表定位信息是一个unsigned long 型的有256的元素的数组，用于定位一级哈希表中的256个一级哈希桶在文件中的准确位置。
3. 变量信息：变量信息的组织形式是二级哈希表，当前站处理其他站发送的网络变量请求信息时，得到请求信息中每个变量的哈希值，哈希值是2个字节，读出2个字节中高字节，作为数组下标，在一级哈希表定位信息中查找到当前哈希桶的位置。

例如，请求信息中变量Var的哈希值是0x1234；那么查找其在本站的信息时，首先取得哈希值的高一个字节，在一级哈希表定位信息数组中直接找到HashLocation[18]，HashLocation[18]的值为0x54321，那么确定一级哈希桶18在文件中的位置是0x54321，直接定位到一级哈希桶18。之后得到其哈希值的低一个字节，再在二级哈希表中查找，最终找到该变量的信息。

一级哈希桶的结构如下图所示：



1. 图3-33 一级哈希桶结构图

此处，一级哈希桶中的二级哈希桶不一定有256个，且其哈希值也不一定是逐个加1，但是，一定是从前到后递增的。

二级哈希桶的结构如下图所示：



1. 图3-34 二级哈希桶结构图

变量的信息如下图所示：



1. 图3-35 本站变量信息

当二级哈希桶中变量的个数超过1的时候，通过变量名的逐个比较进行查找。

接口函数：void CAppGlobalFunc::CreateNetResource()

参数说明：无

功能说明：生成网络变量资源表

返 回 值：无

### 修改网络变量

在网络变量页面内可以修改网络变量的类型及说明，如下图所示。



图3-36：修改网络变量属性

主要接口函数有：

1. void CNetVarView::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：响应Grid控件单元格编辑结束的消息。

返 回 值：无

说 明：结束编辑时，需要调用数据容器中的函数修改变量属性，然后根据返回值判断修改是否成功。如果修改成功，函数执行结束，否则单元格内恢复修改前的值。

1. void CNetVarView::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：响应Grid控件单元格开始编辑的消息。

返 回 值：无

说 明：在此函数内保存修改前的值，便于在修改不成功时恢复修改前的值。

1. bool CNetVarContainer::EditType(CString strKey,CString strNewValue)

参数说明：strKey 变量名

strNewName 新类型名

功能说明：修改网络变量类型

返 回 值：修改成功返回TRUE，否则FALSE

说 明：在此函数内进行数据的修改，即变量对应的CBase结构指针

引用网络变量要遵守特定的格式，即“域站号@变量名”。在域站号中，截取后两位做为站号，其余部分做为域号。如果域站号只有两位，即域号为空，默认为“0”号域。为了支持域站号的自由书写格式，数据容器CNetVarContainer中的m\_netVarMap以网络变量的标准名称做为关键字存储。

### 删除网络变量

在网络变量页面支持删除变量。

接口函数：void CNetVarView::OnDelVar()

参数说明：无

功能说明：删除网络变量

返 回 值：无

## POU局部变量

### PRG POU变量

每一个PRG POU对应一组变量，该组变量以POU名为组名，组指针存放在数据容器CDBContainer中。PRG POU支持的主要功能如下图所示。



图3-37：PRG POU变量组的主要功能

整个变量组对应数据容器CGlobalVarContainer，PRG变量组中的变量可以与内部变量和测点变量重名，因此m\_globalAllVarMap不存放PRG变量组的变量名。但是，为了保证变量别名的唯一性，m\_globalAllVarRefMap中一定要存放PRG变量组的变量别名。

#### 增加变量

PRG变量组支持在页面内增加变量，主要接口函数有：

1. void CGlobalVarView::OnAddVar()

参数说明：无

功能说明：在视图内增加变量

返 回 值：无

说 明：由于PRG变量组表使用CGlobalVarView作为视图类，但与IEC语言共用一个Doc。因此，不能由CGlobalVarDoc的成员m\_pGlobalVarContainer得到数据容器指针，必须由遍历m\_groupMap才能得到。

函数流程图如下图所示：



图3-38：PRG POU增加变量

1. bool CGlobalVarContainer::AddNormalRecord()

参数说明：无

功能说明：在数据容器内增加（new）一个新变量，设置变量的默认属性，并增加到对应的m\_globalVarList和m\_globalVarMap中。次序上，为该组变量的最后一个。

返 回 值：成功返回true，否则false

1. void CAppGlobalFunc::DoAfterChangeofGlobalVar(CGlobalVarContainer\* pGVContainer, CString strVarName)

参数说明：pGVContainer 数据库容器指针

strVarName 变量名

功能说明：全局变量变化后，设置一系列的标志位

返 回 值：无

说 明：具体设置的内容包括：交叉引用表、POU变化标志、工程修改标志、工程编译标志。

#### 插入变量

PRG变量组支持在页面内插入变量，主要接口函数有：

1. void CGlobalVarView::OnInsertVar()

参数说明：无

功能说明：在视图内插入变量

返 回 值：无

说 明：与增加变量相比，插入变量只是增加的次序上有所不同，其余一致，不再赘述。

1. bool CGlobalVarContainer::InsertNormalRecord(CString strKey,int &iIndex)

参数说明：strKey 指定的变量名

Index 插入的位置

功能说明：在数据容器内插入（new）一个新变量，设置变量的默认属性，并增加到对应的m\_globalVarList和m\_globalVarMap中。次序上，由用户选择的实际位置决定。

返 回 值：成功返回true，否则false

#### 删除变量

PRG变量组支持在页面内插入变量，主要接口函数有：

1. void CGlobalVarView::OnDelVar()

参数说明：无

功能说明：在视图内删除变量

返 回 值：无

函数流程图如下图所示



图3-39：PRG POU删除变量

1. bool CGlobalVarContainer::DelNormalRecords(CStringArray \*pPinNames)

参数说明：pPinNames 待删除的全局变量组名数组

功能说明：在数据容器内插入删除选中的一组变量

返 回 值：成功返回true，否则false

#### 修改变量属性

在PRG变量组页面上可以编辑修改变量的属性，除数据容器指针的获得方式外与内部变量相似，可参照内部变量的属性修改。

### FB POU变量

每一个FB POU对应一组变量，对应的数据容器为CLocalVarContainer。该组变量是POU的一部分，通过POU指针可以找到对应的组指针。

FB变量组内的变量没有变量别名，当整个工程处于在线状态时没有在线值

FB POU支持的主要功能如下图所示。

图3-40：FB POU支持的主要功能

#### 页面初始化

当用户打开FB页面时，变量表表随之建立，接口函数为：

1. void CLocalDBView::OnInitialUpdate()

参数说明：无

功能说明：变量视图初始化

返 回 值：成环返回true，否则false

说 明：功能块页面具有四个tap页，默认显示输入页面

1. void CLocalParamDlg::InitGrid()

参数说明：无

功能说明：变量表初始化

返 回 值：无

#### 检查递归的调用环

在FB的定义过程中有可能会形成递归的调用环，如下图所示。形成调用环后，会造成工程的死循环，因此，在变量定义部分必须阻止定义形成递归的调用环。



图3-41：形成递归的调用环

接口函数：bool CFunctionBlockType::IsTtoTRecursion(CString strNewType,CString strOldType)

参数说明：strNewType 准备定义的类型

strOldType 已定义的类型

功能说明：检查是否形成递归的调用环

返 回 值：成环返回true，否则false

函数流程图如下图所示。



图3-42：检查调用环的函数流程图

#### 个数限制

功能块成员分区存储，每个区的子成员个数均有上限，分别是：输入区的上限为64，输出区的上限为32，输入输出区的上限为16，局部区的上限为128。

接口函数：bool CLocalVarContainer::CheckMaxCount(unsigned char ucType)

参数说明：ucType tap页类型

功能说明：检查功能块各区的子成员个数

返 回 值：未达到返回true，否则false

#### 只读属性

功能块输入区引脚的只读属性为true，且不可修改。

接口函数：void CFunctionBlockDB::SetAccess(bool bReadOnly)

参数说明：bReadOnly 是否为只读

功能说明：设置功能块引脚的只读属性

返 回 值：无

#### 增加变量

FB POU变量组支持在页面内增加变量的操作，通过单击右键在菜单中选择“增加变量”即可。主要接口函数有：

1. void CLocalParamDlg::OnAddVar()

参数说明：无

功能说明：在视图内增加变量

返 回 值：无

函数流程图如下图所示。



图3-43：FB POU中增加变量

1. bool CLocalVarContainer::AddRecord(unsigned char ucType)

参数说明：ucType tap页类型

功能说明：在数据容器内增加（new）一个新变量，设置变量的默认属性，并增加到对应的List和Map中。次序上，为该组本区域内变量组的最后一个。

返 回 值：成功返回true，否则false

说 明：对于功能块子成员，新增成员的默认名规则为“p+数字”（数字是从1开始直到该名称唯一），默认类型为“BOOL”类型，Retain、HMI、NeedForce、NeedNet四种属性从配置文件读取，Show属性默认为true。

1. void CAppGlobalFunc::DoAfterChangeofLocalVar(CPOU\* pPOU, char chType)

参数说明：pPOU POU指针

chType tap页类型

功能说明：局部变量变化后，设置一系列的标志位

返 回 值：无

说 明：对于功能块子成员变量来说，具体设置的内容包括：更新版本号、更新该类型引脚、更新相关变量显示引脚、设置工程修改标志、设置工程编译标志、设置POU变化标志。此函数在局部变量的重要属性发生变化时都需要调用，仅在此处做出说明。

#### 插入变量

FB POU变量组支持在页面内插入变量的操作，通过单击右键在菜单中选择“插入变量”即可，主要接口函数有：

1. void CLocalParamDlg::OnInsertVar()

参数说明：无

功能说明：在视图内插入变量

返 回 值：无

说 明：与增加变量相比，插入变量只是增加的次序上有所不同，其余一致，不再赘述。

1. bool CLocalVarContainer::InsertRecord(unsigned char ucType, CString strBeforeName,int &iIndex)

参数说明：ucType tap页类型

strBeforeName 在此引脚前插入新的变量

iIndex 返回插入的位置

功能说明：在数据容器内插入（new）一个新变量，设置变量的默认属性，并增加到对应的List和Map中。次序上，由用户选择的实际位置决定。

返 回 值：成功返回true，否则false

#### 删除变量

FB POU变量组支持在页面内插入变量的操作，通过单击右键在菜单中选择“插入变量”即可，主要接口函数有：

1. void CLocalParamDlg::OnDelVar()

参数说明：无

功能说明：在视图内删除变量

返 回 值：无

流程图如下图所示。



图3-44：FB POU中删除变量

1. bool CLocalVarContainer::DelRecord(unsigned char ucType,CStringArray &NameArray)

参数说明：ucType tap页类型

NameArray 待删除的变量名组

功能说明：在数据容器内删除一组变量

返 回 值：成功返回true，否则false

#### 修改变量属性

变量表中各种属性可以进行修改，但是修改过程中都要对新值进行合法性检查，如果新值合法则修改成功，否则，修改失败，恢复旧值。

修改变量时，主要调用的函数如下：

* void CLocalParamDlg::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：保存旧值，修改失败时用于恢复

返 回 值：无

* void CLocalParamDlg::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：修改变量属性，若新值不合法恢复旧值

返 回 值：无

* bool CLocalVarContainer::EditKey(CString strOldName,CString strNewName,unsigned char ucType)

参数说明：strOldName 原来的变量名

strNewName 修改后的变量名

ucType tap页类型

功能说明：修改变量名称

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditComment(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量说明

功能说明：修改变量说明

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditType(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量类型

功能说明：修改变量类型

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditInitValue(CString strKey, CString &strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量初始值

功能说明：修改变量初始值

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditNeedNet(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的是否上网属性

功能说明：修改变量是否上网属性

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditRetain(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的掉电保护属性

功能说明：修改变量掉电保护

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditNeedForce(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的是否可强制属性

功能说明：修改变量是否可强制属性

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditHMI(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的HMI属性

功能说明：修改变量HMI属性

返 回 值：成功返回true，否则false

* bool CLocalVarContainer::EditShow(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的是否显示属性

功能说明：修改变量是否显示属性

返 回 值：成功返回true，否则false

修改变量属性流程如下图所示：

图3-45：FB POU中修改变量属性

#### 辅助编辑功能

为了便于用户进行编辑操作，FB变量表提供一些辅助的编辑功能，主要接口函数有：

1. void CLocalVarContainer::OnSearch(CString strFind, BOOL bMatchCase, BOOL bMatchWholeWord)

参数说明：strFind 待查找的字符串

bMatchCase 是否大小写敏感

bMatchWholeWord 是否全字匹配

功能说明：查找

返 回 值：无

1. bool CLocalDBView :: OnReplace(CString strFind, CString strReplace, BOOL bWholeWord, BOOL bMatchCase, char& chType)

参数说明：strFind 待查找的字符串

strReplace 待替换的字符串

bWholeWord 是否全字匹配

bMatchCase 是否大小写敏感

chType 操作标记

功能说明：替换

返 回 值：无

1. void CLocalParamDlg::OnUndo()

参数说明：无

功能说明：撤销

返 回 值：无

1. void CLocalParamDlg::OnRedo()

参数说明：无

功能说明：恢复

返 回 值：无

1. void CLocalParamDlg::OnCutVar()

参数说明：无

功能说明：剪切

返 回 值：无

1. void CLocalParamDlg::OnCopyVar()

参数说明：无

功能说明：复制

返 回 值：无

1. void CLocalParamDlg::OnPasteVar()

参数说明：无

功能说明：粘贴

返 回 值：无

### FUN POU变量

与FB类似，每一个FUN POU对应一组变量，对应的数据容器同样为CLocalVarContainer。该组变量是POU的一部分，通过POU指针可以找到对应的组指针。FUN POU支持的主要功能如下图所示。其中，约束子成员类型主要通过约束变量表内的变量类型选项列表完成，并无特定函数控制。



图3-46：FUN POU支持的主要功能

FUN POU变量与FBPOU变量共用一个视图类和数据库容器类，不仅如此，所有的主要功能点的入口函数都一致。因此，此处不再重复叙述，仅挑选其中的不同之处进行描述。

1. void CAppGlobalFunc::DoAfterChangeofLocalVar(CPOU\* pPOU, char chType)

参数说明：pPOU POU指针

chType tap页类型

功能说明：局部变量变化后，设置一系列的标志位

返 回 值：无

说 明：对于函数子成员变量来说，具体设置的内容包括：设置调用POU的编译标志、更新相关变量的显示引脚、变量设置工程修改标志、设置工程编译标志、设置POU变化标志。此函数在局部变量的重要属性发生变化时都需要调用，仅在此处做出说明。

1. void CLocalDBView::LocateAllRow(CElementPosition\* pPos)

参数说明：pPos 定位信息

功能说明：错误定位

返 回 值：无

1. 函数没有输出页面，输入输出页面是否存在由配置文件决定。

接口函数：void CLocalDBView::OnInitialUpdate()

1. 对于函数子成员，新增成员的默认名规则为“p+数字”（数字是从1开始直到该名称唯一），默认类型为“BOOL”类型，Retain、HMI、NeedForce、NeedNet四种属性默认为false，Show属性无用。

接口函数：bool CLocalVarContainer::AddRecord(unsigned char ucType)

1. 函数成员分区存储，每个区的子成员个数均有上限，分别是：输入区的上限为64，输入输出区的上限为16，局部区的上限为128。

接口函数：bool CLocalVarContainer::CheckMaxCount(unsigned char ucType)

## 自定义数据类型

自定义数据类型包括：结构体类型和枚举类型。用户可以根据实际情况定义合适的数据结构进行组态。

### 结构体

结构体类型以CStructView和CStructDoc做为文档视图类，对应的数据库容器为CStructType，已定义好的数据库容器指针存于CUserTypeContainer中。结构体类型支持的主要功能如下图所示。



图3-47：结构体类型支持的主要功能

#### 页面初始化

当用户打开结构体页面时，结构体类型表随之建立，接口函数为：

void CStructView::InitColumnWidth()

参数说明：无

功能说明：在枚举类型对应的视图内增加成员

返 回 值：无

说 明：结构体类型表只有序号、成员名、成员类型、成员说明、初始值五列。

#### 增加成员

结构体类型支持在页面内增加成员的操作，通过单击右键在菜单中选择“增加成员”即可。主要接口函数有：

1. void CLocalParamDlg::OnAddVar()

参数说明：无

功能说明：在视图内增加变量

返 回 值：无

函数流程图如下图所示。



图3-48：结构体类型增加变量

1. bool CStructType::AddPin()

参数说明：无

功能说明：在数据容器内增加（new）一个新成员变量，设置变量的默认属性，并增加到对应的子成员链表m\_pinList中。次序上，为该组成员的最后一个。

返 回 值：成功返回true，否则false

说 明：结构体类型新增成员默认为“BYTE”类型，初始值为“0”。其余成员变量的属性在定义该类型变量时设置，定义结构体类型时不予以设置。

#### 插入成员

结构体类型支持在页面内插入成员的操作，通过单击右键在菜单中选择“插入成员”即可，主要接口函数有：

1. void CStructView::OnInsertPin()

参数说明：无

功能说明：在视图内插入成员

返 回 值：无

说 明：与增加成员相比，插入成员只是增加的次序上有所不同，其余一致，不再赘述。

1. bool CStructType::InsertPin(CString pinName)

参数说明：pinName 在此引脚前插入新的成员

功能说明：在数据容器内插入（new）一个新变量，设置变量的默认属性，并增加到对应的m\_pinList中。次序上，由用户选择的实际位置决定。

返 回 值：成功返回true，否则false

说 明：与增加成员相比，插入成员只是增加的次序上有所不同，其余一致，不再赘述。

#### 删除成员

结构体支持在页面内删除成员的操作，通过单击右键在菜单中选择“删除成员”即可，主要接口函数有：

1. void CStructView::OnDelPin()

参数说明：无

功能说明：在结构体类型视图内删除成员

返 回 值：无

流程图如下图所示。



图3-49：结构体类型中删除成员

1. bool CStructType::DelPins(CStringArray \*pPinNames)

参数说明：pPinNames 待删除的变量名组

功能说明：在对应的数据容器内删除某些变量

返 回 值：成功返回true，否则false

#### 修改成员属性

变量表中各种属性可以进行修改，但是修改过程中都要对新值进行合法性检查，如果新值合法则修改成功，否则，修改失败，恢复旧值。

修改变量时，主要调用的函数如下：

1. void CStructView::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：保存旧值，修改失败时用于恢复

返 回 值：无

1. void CStructView::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：修改变量属性，若新值不合法恢复旧值

返 回 值：无

1. bool CStructType::EditPinKey(CString strKey, CString strNewValue)

参数说明：strKey 原来的变量名

strNewName 修改后的变量名

功能说明：修改变量名称

返 回 值：成功返回true，否则false

1. bool CStructType::EditPinComment(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量说明

功能说明：修改变量说明

返 回 值：成功返回true，否则false

1. bool CStructType::EditPinType(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量类型

功能说明：修改变量类型

返 回 值：成功返回true，否则false

说 明：结构体只支持基本类型，因此修改结构体成员只能在类型中选择

1. bool CStructType::EditPinInitValue(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量初始值

功能说明：修改变量初始值

返 回 值：成功返回true，否则false

修改变量属性流程如下图所示：

图3-50：结构体类型中修改成员初始值

#### 辅助的编辑功能

为了便于用户进行编辑操作，结构体变量表提供一些辅助的编辑功能，主要接口函数有：

1. void CStructType::OnSearch(CString strFind, BOOL bMatchCase, BOOL bMatchWholeWord)

参数说明：strFind 待查找的字符串

bMatchCase 是否大小写敏感

bMatchWholeWord 是否全字匹配

功能说明：查找

返 回 值：无

1. bool CStructView :: OnReplace(CString strFind, CString strReplace, BOOL bWholeWord, BOOL bMatchCase, char& chType)

参数说明：strFind 待查找的字符串

strReplace 待替换的字符串

bWholeWord 是否全字匹配

bMatchCase 是否大小写敏感

chType 操作标记

功能说明：替换

返 回 值：无

1. void CStructView::OnUndo()

参数说明：无

功能说明：撤销

返 回 值：无

1. void CStructView::OnRedo()

参数说明：无

功能说明：恢复

返 回 值：无

1. void CStructView::OnCutVar()

参数说明：无

功能说明：剪切

返 回 值：无

1. void CStructView::OnCopyVar()

参数说明：无

功能说明：复制

返 回 值：无

1. void CStructView::OnPasteVar()

参数说明：无

功能说明：粘贴

返 回 值：无

### 枚举

枚举类型以CEnumView和CEnumDoc做为文档视图类，对应的数据库容器为CEnumType，已定义好的数据库容器指针存于CUserTypeContainer中。结构体类型支持的主要功能如下图所示。



图3-51：枚举类型支持的主要功能

#### 页面初始化

当用户打开枚举页面时，枚举类型表随之建立，接口函数为：

void CEnumView::InitColumnWidth()

参数说明：无

功能说明：在枚举类型对应的视图内增加成员

返 回 值：无

说 明：枚举类型表只有序号、成员名、成员说明、初始值四列。

#### 增加成员

枚举类型支持在页面内增加成员的操作，通过单击右键在菜单中选择“增加成员”即可。主要接口函数有：

1. void CEnumView::OnAddPin()

参数说明：无

功能说明：在枚举类型对应的视图内增加成员

返 回 值：无

函数流程图如下图所示。



图3-52：枚举类型增加成员

1. bool CEnumType::AddPin( )

参数说明：无

功能说明：在数据容器内增加（new）一个新成员变量，设置变量的默认属性，并增加到对应的子成员链表m\_pinList中。次序上，为该组成员的最后一个。

返 回 值：成功返回true，否则false

说 明：枚举类型新增成员默认为“INT”类型，初始值为“INT”型变量对应的默认初始值。其余成员变量的属性在定义该类型变量时设置，定义结构体类型时不予以设置。另外，子成员有个数上限，最多为65536个，当个数达到上限后，增加成员失败。

1. void CAppGlobalFunc::DoAfterChangeofEnum(CEnumType \*pType)

参数说明：pType 枚举类型指针

功能说明：枚举类型有成员变化导致类型实质发送变化时，调用此接口设置一系列的标志位。

返 回 值：无

说 明：在此接口中更新类型版本号、更新该类型变量引脚、设置工程修改标志、设置工程编译标志、设置POU编译标志。

#### 插入成员

枚举类型支持在页面内插入成员的操作，通过单击右键在菜单中选择“插入成员”即可，主要接口函数有：

1. void CEnumView::OnInsertPin()

参数说明：无

功能说明：在枚举类型对应的视图内插入成员

返 回 值：无

说 明：与增加成员相比，插入成员只是增加的次序上有所不同，其余一致，不再赘述。

1. bool CEnumType::InsertPin(CString strPinName)

参数说明：strPinName 在此引脚前插入新的成员

功能说明：在数据容器内插入（new）一个新变量，设置变量的默认属性，并增加到对应的m\_pinList中。次序上，由用户选择的实际位置决定。

返 回 值：成功返回true，否则false

说 明：与增加成员相比，插入成员只是增加的次序上有所不同，其余一致，不再赘述。

#### 删除成员

枚举类型支持在页面内删除成员的操作，通过单击右键在菜单中选择“删除成员”即可，主要接口函数有：

1. void CEnumView::OnDelPin()

参数说明：无

功能说明：在枚举类型对应的视图内删除成员

返 回 值：无

流程图如下图所示。



图3-53：枚举类型中删除成员

1. bool CEnumType::DelPin(CStringArray &strPinNameArray)

参数说明：strPinNameArray 待删除的变量名组

功能说明：在对应的数据容器内删除某些变量

返 回 值：成功返回true，否则false

#### 修改成员属性

变量表中各种属性可以进行修改，但是修改过程中都要对新值进行合法性检查，如果新值合法则修改成功，否则，修改失败，恢复旧值。

修改变量时，主要调用的函数如下：

1. void CEnumView::OnGridStartEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：保存旧值，修改失败时用于恢复

返 回 值：无

1. void CEnumView::OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult)

参数说明：见MSDN

功能说明：修改变量属性，若新值不合法恢复旧值

返 回 值：无

1. bool CEnumType::EditPinKey(CString strKey, CString strNewValue)

参数说明：strOldName 原来的变量名

strNewName 修改后的变量名

功能说明：修改变量名称

返 回 值：成功返回true，否则false

1. bool CEnumType::EditPinComment(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量说明

功能说明：修改变量说明

返 回 值：成功返回true，否则false

1. bool CEnumType::EditPinInitValue(CString strKey, CString strNewValue)

参数说明：strKey 变量名

strNewName 修改后的变量初始值

功能说明：修改变量初始值

返 回 值：成功返回true，否则false

修改变量属性流程如下图所示：



图3-54：枚举类型修改变量属性

#### 辅助的编辑功能

为了便于用户进行编辑操作，FB变量表提供一些辅助的编辑功能，主要接口函数有：

1. void CEnumType::OnSearch(CString strFind, BOOL bMatchCase, BOOL bMatchWholeWord)

参数说明：strFind 待查找的字符串

bMatchCase 是否大小写敏感

bMatchWholeWord 是否全字匹配

功能说明：查找

返 回 值：无

1. bool CEnumView :: OnReplace(CString strFind, CString strReplace, BOOL bWholeWord, BOOL bMatchCase, char& chType)

参数说明：strFind 待查找的字符串

strReplace 待替换的字符串

bWholeWord 是否全字匹配

bMatchCase 是否大小写敏感

chType 操作标记

功能说明：替换

返 回 值：无

1. void CEnumView::OnUndo()

参数说明：无

功能说明：撤销

返 回 值：无

1. void CEnumView::OnRedo()

参数说明：无

功能说明：恢复

返 回 值：无

1. void CEnumView::OnCutVar()

参数说明：无

功能说明：剪切

返 回 值：无

1. void CEnumView::OnCopyVar()

参数说明：无

功能说明：复制

返 回 值：无

1. void CEnumView::OnPasteVar()

参数说明：无

功能说明：粘贴

返 回 值：无

## 变量自动声明

为了方便用户组态，变量部分提供变量自动声明功能，可以声明新变量或者修改已有变量的部分属性，如下图所示。



图3-55：变量自动声明主要功能

注：在变量自动声明中新增和修改变量同样遵循相应的规则，即与变量定义表的操作保持一致的结果，例如：在变量定义表中通过菜单增加全局变量时需要设置编译标志，如果通过自动变量声明添加编译标志同样需要设置，两种操作在实质上是一致的。因此，诸多规则本节不再赘述，可操作、、参见变量定义表的规则进行。

### 定义新变量

当IEC语言编辑单元结束编辑状态或者向IEC编辑区拖入功能块时，自动为未声明的变量弹出变量自动声明。主要接口有：

1. bool CGlobalFunc::AutoVarDeclare(CString strVarName, CString strVarType, CPOU \*pPOU, CView \*pView, CBaseDB \*&pVar, CString strComment,CString strRefName)

参数说明：strVarName 变量名称

strVarType 变量类型

pPOU 所在POU的指针

pView 视图指针

pVar 变量指针

strComment 变量说明

strRefName 变量别名

功能说明：变量自动声明总入口

返 回 值：无

函数流程图如下图所示



图3-56：变量自动声明总入口

1. bool CDlgAssistant::AutoVarDeclare(CString strVarName,CString strVarType,CPOU \*pPOU,CView \*pView,CBaseDB \*&pVar,CString strComment,CString strVarRefName)

参数说明：strVarName 变量名称

strVarType 变量类型

pPOU 所在POU的指针

pView 视图指针

pVar 变量指针

strComment 变量说明

strRefName 变量别名

功能说明：非网络变量的变量自动声明总入口

返 回 值：成功返回true，否则false

函数流程图如下图所示



图3-57：非网络变量的变量自动声明总入口

1. bool CDlgNetVar::AutoVarDeclare(CString strAllName,CBaseDB \*&pVar)

参数说明：strAllName 网络变量名称

pVar 变量指针

功能说明：网络变量自动声明

返 回 值：成功返回true，否则false

函数流程图如下图所示



图3-58：网络变量的变量自动声明总入口

### 修改变量属性

选中IEC编辑单元点击F2键，如果变量已定义，可以弹出变量自动声明对话框。用户可以在对话框中修改变量的部分属性。变量声明的这一功能又称为输入助手，主要接口有：

1. bool CDlgVarShow::ShowVarInfo(CString strKey,CPOU \*pPOU,CView \*pLocalView)

参数说明：strVarName 变量名称

pPOU 所在POU的指针

pLocalView 切分窗口中属于局部变量的视图指针

功能说明：F2输入助手的总入口

返 回 值：成功返回true，否则false

函数流程图如下图所示

图3-59：输入助手总入口

1. bool CDlgNetVarShow::AutoVarShow(CString strAllName)

参数说明：strAllName 网络变量全称

功能说明：F2网络变量输入助手的总入口

返 回 值：成功返回true，否则false

函数流程图如下图所示



图3-60：网络变量输入助手总入口

## 类型变化及库变化

当自定义数据类型（当前工程或库中）发生变化时，工程中所有该类型的数据随之自动更新。以功能块为例，对于功能块“FB”类型变量g1，当FB类型增加一个成员后，变量自动更新，如下图所示。

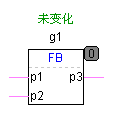
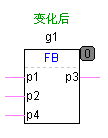
 

图3-61：类型变化示意图

为了实现这一功能，当自定义数据类型（功能块、结构体、枚举）发生变化（增、删、改、重命名）时，都要进行类型变化，调用的主要接口有。

1. void CAppGlobalFunc::UpdateTypeChange(CStringArray &changeTypeArray)

参数说明：changeTypeArray 需要变化的类型名称组

功能说明：类型变化总入口

返 回 值：无

函数流程图如下图所示

图3-62：类型变化总入口

1. void CAppGlobalFunc::UpdateAllType(CStringArray &changeTypeArray

,CStringArray &allTypeArray)

参数说明：changeTypeArray 需要变化的类型名称组

allTypeArray 引起变化的类型名

功能说明：FB变量组的类型变化总入口

返 回 值：无

函数流程图如下图所示



图3-63：FB变量组的类型变化总入口

1. void CAppGlobalFunc::UpdateAllGlobal(CStringArray& allTypeArray)

参数说明：allTypeArray 需要变化的类型名称组

功能说明：PRG变量组+内部变量组+全局变量组的类型变化总入口

返 回 值：无

函数流程图如下图所示



图3-64：PRG变量组+内部变量组+全局变量组的类型变化

1. void UpdateOneType(CString strType,CMap<CString,LPCTSTR

,CList<CString,CString>\*,CList<CString,CString>\*>& reverseMap,

CMap<CString,LPCTSTR,CString,CString>& typeMap)

参数说明：strType 类型字符串

reverseMap 类型逆序引用表

typeMap 类型表(增加的变化)

功能说明：在功能块内，更新某一类型对应的变量

返 回 值：无

函数流程图如下图所示



图3-65：在功能块内，更新某一类型对应的变量

1. void CAppGlobalFunc::UpdateAllLocal(CStringArray& allTypeArray)

参数说明：allTypeArray 需要变化的类型名称组

功能说明：FUN变量组的类型变化总入口

返 回 值：无

函数流程图如下图所示



图3-66：FUN变量组的类型变化总入口

1. void CGlobalVarContainer::UpdateGlobalVar(CStringArray& allTypeArray)

参数说明：allTypeArray 所有变化的类型

功能说明：类型变化，更新变量组中变量的总入口

返 回 值：引起该pou变化返回true，否则false

函数流程图如下图所示



图3-67：类型变化，更新变量组中变量的总入口

1. void CGlobalVarContainer::UpdateGlobalVarList(CStringArray& allTypeArray,

CList<CBaseDB\*,CBaseDB\*> \*pList,CMap<CString,LPCTSTR,

CBaseDB\*,CBaseDB\*> \*pMap)

参数说明：allTypeArray 所有变化的类型

pList 待更新的list

pMap 待更新的map

功能说明：类型变化，更新变量组中的变量

返 回 值：无

函数流程图如下图所示



图3-68：类型变化，更新变量组中变量的总入口

1. bool CLocalVarContainer::UpdateLocalVar(CStringArray& allTypeArray)

参数说明：allTypeArray 所有变化的类型

功能说明：类型变化，更新变量组中的变量

返 回 值：无

思路与全局变量类似，参考全局变量对应的函数流程图。

1. bool CLocalVarContainer::UpdateLocalVarList(CStringArray& allTypeArray

,CList<CBaseDB\*,CBaseDB\*> \*pList,CMap<CString,LPCTSTR,

CBaseDB\*,CBaseDB\*> \*pMap,unsigned char ucType)

参数说明：allTypeArray 所有变化的类型

功能说明：类型变化，更新变量组中的变量

返 回 值：无

思路与全局变量类似，参考全局变量对应的函数流程图。

## 在线部分

不同类型的变量在线时处理方式不同，下面分别介绍不同类型变量在线时的处理方式。

### 内部变量在线

所有的内部变量在线时都可以被监视，在线过程中主要调用的接口如下：

* void CGlobalVarView::CollectAllOnlineData()

功能：收集视图中可见的变量

参数：无

返回值：无

说明：通知在线或视图改变时重新收集可见的变量，只监视可见部分变量的在线值

* void CGlobalVarView::RefreshOnlineValue()

功能：刷新收集变量的在线值

参数：无

返回值：无

说明：实时响应刷新消息，更新可见变量的在线值

内部变量在线流程如下图所示：



图 3‑69：内部变量在线流程

### 测点变量在线

测点都是功能块类型的变量，测点变量视图中各列显示的是测点变量引脚的值，在线时，需要在视图中监视各引脚的在线值。在线过程中主要调用的函数如下：

* void CPinGlobalVarView::CollectAllOnlineData()

功能：收集视图中可见变量

参数：无

返回值：无

说明：测点视图中各列是测点的引脚值，因此需要收集可见测点变量在当前显示方式下可见的引脚

* void CPinGlobalVarView::RefreshOnlineValue()

功能：刷新可见测点变量各引脚的在线值

参数：无

返回值：无

说明：实时响应刷新消息，刷新可见测点变量各引脚的在线值

测点变量在线流程如下图所示：



图 3‑70：测点变量在线流程

### FB POU、FUN POU变量在线

FB POU、FUN POU内部的变量，在线时使变量不可编辑，离线时使变量可以编辑，流程如下图所示：



图 3‑71：FB POU、FUN POU内部的变量在线状态

### PRG POU变量在线

在工程中被任务调用的PRG POU与未被调用的PRG POU在线时处理方式不同，被调用的PRG POU中的变量是可以被监视的， 未被调用PRG POU中的变量不需要被监视，没有在线值。

被调用PRG POU中的变量在线过程同内部变量在线，未被调用PRG POU中的变量在线时同FB POU、FUN POU内部变量在线

### 详细对话框中变量在线

内部变量、测点变量、被调用PRG POU内部变量的详细对话框中的引脚在线时可以被监视，在线过程主要调用的函数如下：

* void CDlgVarDetail::CollectAllOnLineData()

功能：收集详细对话框中可见的引脚变量

参数：无

返回值：无

说明：详细对话框中可见的引脚变量被记录在可见列表变量容器中

* void CDlgVarDetail::RefreshOnlineValue()

功能：刷新详细对话框中可见引脚的在线值

参数：无

返回值：无

说明：实时响应刷新消息，刷新详细对话框中可见引脚的在线值

详细对话框中变量在线流程同内部变量在线流程。

# 附录

## 支持类型

AT中支持的数据类型及分类如下图所示

图4-1：AT中支持的数据类型及其分类

## 变量属性设置

对于具有子成员的复杂类型，设置变量本身的属性值会影响到子成员的属性值。这些属性值包括：Retain、NeedForce、HMI、NeedNet以及功能块子成员特有的显示属性，子成员的属性值设置遵守特定的规则，如下表所示。

表 复杂类型子成员属性值（Retain、NeedForce、HMI、NeedNet、显示）设置规则

表 4‑1：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 本身为TRUE | 本身为FALSE |
| 功能块 | 子成员属性按照类型设置 | 子成员属性均为false |
| 结构体 | 子成员属性按照类型设置 | 子成员属性均为false |
| 数组 | 子成员属性均为true | 子成员属性均为false |

## 相关文件及内容

表 4‑2：相关文件及内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 文件名 | 文件内容 | 数据结构 | 装载时机 |
| 1 | db.dat | 测点类型文件，定义测点类型、每种类型各点项的默认值及点项之间的约束关系 | CDBClassManage  CDBClassSet | 新建/打开工程时装载 |
| 2 | wg.ini | 定义了各种硬件模块对应的WG（增益）、TP（信号类型）、MU、MD之间的关系 | CDP\_WG\_Table  CDP\_WG | 新建/打开工程时装载 |
| 3 | Transfer.ini | 定义可相互转移的测点类型 | CTransferTable | 新建/打开工程时装载 |
| 4 | ModuleChannel.ini | 定义各模块不同通道的默认信号类型 | CModuleChannelTPTable  CModuleChannelTPSet | 新建/打开工程时装载 |
| 5 | HSDB文件夹中的DBAM6.ini等文件 | 定义各类型测点的显示方式 | CColGroupS  CColGroup | 打开测点文档视图时装载 |
| 6 | UT.ini | 定义量纲选项 | CUTManage | 新建/打开工程时装载 |

## 相关表格及内容

1. 总控AT导入测点格式

数据库总控将每个站中的测点变量保存到相应的文件中，文件格式如下：

文件名：站号.var

文件格式：

表 4‑3：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 类型个数 | char | 1 |
| 类型1信息 |  |  |
| 类型2信息 |  |  |
| …… |  |  |
| 类型n信息 |  |  |

类型信息格式

表 4‑4：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 类型名称 |  | 32 |
| 变量个数 | int | 4 |
| 变量1信息 |  |  |
| 变量2信息 |  |  |
| …… |  |  |
| 变量n信息 |  |  |

变量信息格式

表 4‑5：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 变量名长度 | char | 1 |
| 变量名 | char[] | 变量名长度 |
| 成员个数 | int | 4 |
| 成员1信息 |  |  |
| 成员2信息 |  |  |
| …… |  |  |
| 成员n信息 |  |  |

成员信息格式

表 4‑6：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 大小（byte） |
| 成员名长度 | char | 1 |
| 成员名 | char[] | 成员名长度 |
| 成员值长度 | char | 1 |
| 成员值 | char[] | 成员值长度 |