

# AicDataHub 用户手册

北京艾克信控科技有限公司

1	简介:1
2	安装与卸载2
	2.1 安装
	2.2 卸载
3	登录和主界面5
	3.1 登录5
	3.2 主界面7
4	对象类型(ObjectTypes)9
	4.1 基本功能9
	4.1.1 新建变量(New Variable)12
	4.1.2 新建属性(New Property)17
	4.1.3 新建方法(New Method)20
	4.1.4 新建报警(New Alarm)
	4.2 高级功能25
	4.2.1 子对象一新建对象25
	4.2.2 继承和重载一新建对象类型27
	4.2.3 解除到父对象关联
	4.2.4 新建脚本
	4.2.5 配置导出和导入31
	4.2.6 应用类型修改
5	工厂对象(PlantObjects)34
	5.1 新建对象34
	5.2 对象管理
	5.2.1 应用类型定义变化(Apply Type Definition Changes)37
6	报警(Alarms)
	6.1 报警类型

## AicDataHub 用户手册

	6.1.1	系统异常报警(SystemOffNormalAlarmType):	.39
	6.1.2	2 跳变报警(TrigAlarmType):	.45
	6.1.3	3 单一越限报警 (ExclusiveLimitAlarmType):	.45
	6.1.4	4 单一变化率报警(ExclusiveRateOfChangeAlarmType):	.49
	6.1.5	5 单一偏差报警(ExclusiveDeviationAlarmType):	.50
	6.1.6	5 非单一报警(复合报警)	.53
	6.2 报	警区域	.53
	6.2.1	报警根区域	.53
	6.2.2	2 添加新的报警区	.54
	6.2.3	3 添加已存在的对象到报警区	.55
	6.3 独	立报警	.56
7	脚本(S	cripts)	.58
	7.1 常	7用上下文	.58
	7.2 脚	」本常用函数	.59
	7.3 条	·件脚本	.60
	7.4 添	加数据变化脚本	.62
	7.5 添	加系统脚本	.64
	7.6 添	加用户功能脚本	.65
	7.7 引	用程序集(ReferenceAssemblies)	.65
8	属性(P	Property)	.67
	8.1 历	史记录设置	.68
	8.2 工	程单位量程	.69
	8.3 仪	表量程	.70
	8.4 工	程单位	.71
9	安全管理	里(Security)	.73
	9.1 用	户管理	.73
	9.1.1	创建新用户	.74

	9.1.2	删除用户	74
	9.1.3	修改密码	75
	9.1.4	重置密码	75
	9.1.5	解锁用户	75
	9.1.6	添加用户角色	75
	9.1.7	删除用户角色	76
	9.2 角1	色管理	77
	9.2.1	创建新角色	77
	9.2.2	删除角色	78
	9.2.3	为某个角色添加用户	78
	9.2.4	为某个角色删除用户	78
10	附录一常	常用接口函数	79
	10.1 地	址空间访问接口(IServerAddressSpace)	79
	10.1.1	1 FindNode	79
	10.1.2	2 GetReferences	80
	10.2 节。	点访问接口(IServerNode)	80
	10.2.1	1 FindChild	80
	10.2.2	2 FindProperty	81
	10.2.3	3 FindMethod	
	10.2.4	4 ReadAttribute	
	10.2.5	5 WriteAttribute	85
	10.2.6	6 ClearChangeMasks	87
	10.3 方法	法节点接口(IServerMethodNode)	87
	10.3.1	1 Call	87
	10.4 变	量节点接口(IServerVariableNode)	89
	10.4.1	1 GetValue	89
	10.5 重/	件访问接口(IFilterTarget)	90
	10.5 🕀	(in more ranged)	

10.5.1 IsTypeOf	90
10.5.2 GetAttributeValue	

# 1 简介:

工业实时库 AicDatahub 是 AicVision 套件的核心, 它实现了大规模实时数据 处理,包括脚本、报警、事件,以及历史数据存储,全面实现 OPC UA 标准接口,具有如下特点:

- ① 基于 OPC UA 标准的数据通信,实现了完整的 OPC UA 服务接口,包括 DA, HA 以及 AC。
- ② 大容量,高实时性:单台普通服务器能支持 100,000 点规模,每秒采集 100,000 条数据,存储 20,000 条历史数据。
- ③ 基于 OPC UA 标准实现对象模型:实现了基于 OPC UA 标准的复杂对象 建模功能,对可重用的设备运用面向对象的方法进行建模,大大提高了 组态的效率,也简化了设备对象的管理。
- ④ 支持实时报警检测:除了常用的限值报警(高、低、高高、低低)、变化 率和偏差报警外,还支持一般的条件表达式报警检测。
- ⑤ 支持多种服务器脚本:包括系统启动、终止、登录、登出、定时器、条件触发等多种类型的脚本,脚本采用C#语言编写,具有无与伦比的强大功能,避免了传统脚本语言的局限性;而且脚本是编译执行而非解释执行,大大提高了脚本执行的性能。
- ⑥ 不仅支持实时数据的订阅/发布,还支持实时报警和事件的订阅/发布。
- ⑦ 支持变量数据的历史记录和查询。
- ⑧ 支持报警和事件的历史记录和查询。
- ⑨ 在线组态:完全支持在线组态,所有的配置均能在线修改,避免了修改 配置时要求服务器重启导致的服务中断,大大提高了系统的可用性。
- ⑩ 支持基于角色的安全控制机制 (RBAC): 用户、角色(用户组)、对象实体、操作(权限)。

工业实时库 AicDatahub 包括两个组件:

- ① AicDataHub Server: 以 Windows 服务方式运行的后台服务程序。
- ② AicDataHub Configurator: 是数据库配置和建模的工具。

# 2 安装与卸载

# 2.1 安装

双击"AicDataHub"安装程序,打开安装许可协议阅读窗口(如图 2.1 所示), 勾选"我同意许可协议中的条款(I agree to the license terms and conditions)", 然 后点击"安装(Install)"按钮立即进入安装界面(如图 2.2 所示),默认将此软 件安装在 C 盘。如果想修改安装路径,则先点击"选项(Options)"按钮,自定 义好安装路径后再单击"安装(Install)"按钮即可。安装完成后会弹出安装成功 提示窗口(如图 2.3 所示)。

H AicTech DataHub Server Setup	
AicTech DataHub Server	
SOFTWARE LICENSE	AGREEMENT
This is a legal agreement between you (either an individual and AicTech Ltd. ("AicTech"). By installing the enclosed soft agreeing to be bound by the terms of this you do not agree to the terms of this Agreement, prom software and the accompanying items to the place you obtain	I or an entity) ware, you are Agreement. If otly return the ed them for a T
I agree to the license terms Options Install	and conditions

图 2-1 安装许可协议窗口



图 2-2 安装界面



图 2-3 安装成功提示窗口

# 2.2 卸载

在 Windows 系统的"开始"菜单中选择"控制面板",然后找到"程序"项, 点击"卸载程序",在列表中找到"AicTech DataHub Sever",右击选择"卸载" 即可。如图 2.4 所示。

SA	发布者	安装时间	大小	版本
🛱 AicTech DataHub Server	AicTech	2013/3/14	243 MB	1.0.0.0
AicTech IOServer	AicTech	2013/1/9	154 MB	1.0.0.0

图 2-4 卸载程序

# 3 登录和主界面

# 3.1 登录

运行 AicDataHub Configurator 软件后,首先进入的是登录界面(如图 3.1 所示)。用户必须正确的填写服务器的地址(Server Uri)、选择合适的安全模式(Security Mode)和安全策略(Security Policy)、填写正确的用户名(User Name)及密码(Password),然后点击"确定(OK)"按钮方可登录成功进入主界面。 否则将会弹出错误信息,提醒用户输入错误。点击"取消(Cancel)"按钮则会直接退出系统。

登录		- 🗆 X
	服务器地址	opc.tcp://127.0.0.1:4520/DataHubServer
	安全模式	None 🔻
	安全策略	None •
	用户名	sa
	密码	••
		确定 取消

图 3-1 登录

服务器地址的选择:

如果不确定服务器的 Uri,可以点击登录界面右侧的 … 浏览(Browse) 按钮,打开服务器地址选择窗口(如图 3.2 所示)。在 Host Name 中输入服务器 IP 地址或主机名,然后点击右侧的 … 发现(Discover) 按钮,则会在终端列 表中展示所有通信协议和安全模式的终端信息。

发现服务器	- 🗆 X
主机名 127.0.0.1	<b>•</b>
服务器列表 AicTech DataHub Server	终结点列表 opc.tcp://127.0.0.1:4520/DataHubServer - SignAndEncrypt opc.tcp://127.0.0.1:4520/DataHubServer - Sign opc.tcp://127.0.0.1:4521/DataHubServer - None http://127.0.0.1:4521/DataHubServer - SignAndEncrypt http://127.0.0.1:4521/DataHubServer/Basic256 - Sign http://127.0.0.1:4521/DataHubServer/None - None 确定 取消

#### 图 3-2 发现服务器地址

其中以"opc.tcp"开头的表示底层使用 TCP 通信协议,以"http"开头的表示运用的是 http 协议,以"https"开头的表示使用的是 https 加密协议(图中的服务器没有配置此协议)。

# 安全模式

登录		_ 🗆 X
服务器	如 opc.tcp://127.0.0.1:4520/DataHubServer	
安全	र्त None	-
安全	路 None	
用	名 Sign SignAndEncrypt	
	码	
		确定 取消

## 图 3-3 选择安全模式

- ◆ None: 表示没有任何安全模式, 安全级别最低。
- ◆ Sign: 表示"签名"的安全模型,安全级别中等。

◆ SignAndEncrypt:表示"签名和加密"的安全模式,安全级别最高。依据不同的安全级别的要求选择相应的安全模式。

安全策略

登录		-		×
服务器地址	opc.tcp://127.0.0.1:4520/DataHubServer			
安全模式	None	•		
安全策略	None	•		1
用户名	None			1
密码	Basic128Rsa15			1
	Basic256			1
		确定 耳	消	
	登录         服务器地址         安全模式         安全策略         用户名         密码	登录 服务器地址 opc.tcp://127.0.0.1:4520/DataHubServer 安全模式 None 安全策略 None 用户名 None Basic128Rsa15 Basic256	登录 服务器地址 opc.tcp://127.0.0.1:4520/DataHubServer   安全模式 None   安全策略 None   用户名 None   Basic128Rsa15 Basic256   确定 興	登录 URSS器地址 opc.tcp://127.0.0.1:4520/DataHubServer Ge全模式 None Control None Th户名 ABasic128Rsa15 Basic256 ABASIC256 ABASIC255 ABASIC255 A

#### 图 3-4 选择安全策略

Security Policy(安全策略)中,"None"表示没有运用加密算法,安全级别最低;"Basic128Rsa15"和"Basic256"表示分别使用不同的加密算法进行加密, "Basic256"安全级别中等,"Basic128Rsa15"安全级别最高。

## 3.2 主界面

登录系统成功后将进入软件主界面(如图 3.5 所示),主界面由"模型浏览器","主窗口","信息窗口"以及"错误列表"窗口组成。

		_	_				-			_	 	- <b>A</b> - X
+=								Aiclech Da			-	
		X	Da h	0		2	~	80				
保存保存所有关闭	关闭所有	館切	复制 粘贴	模型浏览器	信息窗口	错误列表	检查脚本	登录 注销				
文件			剪贴板		窗口视图		脚本	脸证				
模型浏览器	ı	а х										
▶ 📴 工厂对象												
▶ 🧰 对象类型												
▶ □ 报答												
▼ <u></u> x±												
			错误列表									ů x
			て文	<b>‡</b> τ	<b>π</b> τ	列丁	错误代码	て 描述				τ
			信息索口 世	に見刻事					 		 	
<u></u>				「大クリスを								

图 3-5 主界面

# 4 对象类型(ObjectTypes)

对象类型是指从同类或相似的实例中抽象出来的一类模型,这类模型具有这些实例的公共的特征。建立好一个模型之后,就可以以此模型为模板来轻松构建 无数个同类的实例。

## 4.1 基本功能

右击主界面的对象类型(ObjectTypes)文件夹选择"新建对象类型(New ObjectType)"(如图 4.1),即可出现"新建对象类型"对话框(如图 4.2)。填入 合适的"浏览名称(BrowseName)"和"描述(Description)"后点击"确定(OK)" 按钮即可出现新建对象类型的编辑窗口(如图 4.3)。

注意:对于所有节点的浏览名称(BrowseName),只有在用户新建的时候才能输入,一旦创建以后不能再修改。此外,根据 OPC UA 标准,浏览名称应使用非本地化字符(即英文),且禁止使用以下特殊字符/&:.<#!>\%\*="',;?。强烈建议用户仅使用英文大小写字母、数字以及下划线的组合,并且以字母开头。



图 4-1 新建对象类型

新建对象类型		
浏览名称	Tank	
描述	罐体	
	确定取消	

图 4-2 新建对象类型

Tank	
◇ 父对象信息	
父节点	对象类型
引用类型	HasSubtype
◇ 节点属性	
节点分类	ObjectType
节点ID	ns=3;i=1
浏览名称	2:Tank
显示名称	▼ Tank
描述	<ul> <li>▼ </li> <li></li></ul>
可写标志	1048672
抽象类型	

#### 图 4-3 对象类型编辑窗

对象类型编辑窗口包含"父对象信息(Parent Information)"、"节点属性(Node Attributes)"两部分。

## 父对象信息(Parent Information)

包含"父节点(Parent Node)"和"引用类型(Reference Type)"信息。其中"父节点(Parent Node)"表示该项目上一级的节点名称,本例中父节点是"对象类型";"引用类型(Reference Type)"表示父节点相对本变量的引用关系,引用类型遵循 OPC UA 标准。在本例中,引用类型是"HasSubType"。

# 节点属性(Node Attributes)

节点属性包含以下信息:

节点分类(NodeClass):表明节点的类型,节点类型定义遵循 OPC UA 的标准,这里是 ObjectType。

节点 ID(NodeId):表明 OPC UA 地址空间的 ID,包括 ns(命名空间)和 id(标识)两部分组成,系统自动分配。

浏览名称(BrowseName): OPC UA 地址空间的可读节点名,系统根据用户用户输入的浏览名称再附加上命名空间编号所构成。

显示名称 (DisplayName): 表示该节点显示的名称,可以自己定义。

描述 (Description): 用于添加对新建的节点的解释标注。

可写标志(WriteMask):标识哪些属性是可更改的,哪些是不可更改的。

抽象类型(IsAbstract): 指明是否为抽象类型,抽象类型不能有实例,只能 被子类型继承(继承将在后续章节介绍)。默认都是非抽象类型。

右键点击新建的对象类型,弹出编辑菜单:



图 4-4 对象类型编辑菜单

本小节中将介绍基本功能,包括:新建变量、属性、方法、报警。下一节"高级功能"中将介绍新建对象、新建对象类型、新建脚本和导入导出等功能。

# 4.1.1 新建变量(New Variable)

可以在类型下面建成员变量,右键点击某对象类型,选择"新建变量"弹出"新建变量"对话框(如图 4.5 所示)。

新建变量	×
浏览名称	Level
描述	液位
数据类型	Double
维度	Scalar 🔹
保存历史数据	
- 美型定义	DataItemType
建模规则	Mandatory 🔹
启用批量模式	
	确定取消

#### 图 4-5 新建变量对话框

填入相应的"浏览名称 (BrowseName)"和"描述 (Description)"。

**数据类型(DataType)**:单击右侧的 ---- 按钮弹出数据类型选择对话框(如 图 4.6 所示)。数据类型是 OPC UA 标准的数据类型(具体请参考 OPC UA 标准 的相关文档),选择相应的数据类型后单击"确定(OK)"完成。



图 4-6 选择数据类型

**维度(ValueRank)**:标识值是否是一个数组,如果它是一个数组,它允许指定数组的维度。单击右侧小三角则展开数据维度(ValueRank)的所有选项,选择相应的选项即可(如图 4.7 所示)。

新建变量	- 🗆 X
浏览名称	Level
描述	液位
数据类型	Double
维度	Scalar 🔹
保存历史数据	ScalarOrOneDimension
类型定义	Any
2212121	Scalar
建模规则	OneOrMoreDimensions
启用批量模式	OneDimension
	TwoDimensions
	ThreeDimensions
	FourDimensions

图 4-7 变量维度

保存历史数据(Historizing):指示服务器目前是否收集 Value 的历史。勾选则表示服务器目前收集 Value 的历史,否则表示不收集。系统有默认的历史记录 配置,如果需要修改默认配置,需要增加 HitorizingConfiguration 属性。具体介 绍请参考后续的增加属性章节。

**类型定义(Type Definition)**:单击右侧的 按钮,则弹出"变量类型选择"对话框。变量类型是 OPC UA 定义的标准变量类型 DataItemType 及其子类型,默认的变量类型为 DataItemType。选择相应的类型后单击"确定"完成选择。

选择VariableType	-		х
⊿ 🖗 DataItemType			
— 🐼 AnalogItemType			
🕨 🖗 ArrayItemType			
▶ 🖗 DiscreteItemType			
确定	取	肖	

图 4-8 选择类型定义

建模规则(Modeling Rule): "建模规则(Modeling Rule)"表示在构建一 个模型的实例时其中的变量是如何继承或者共享的。"建模规则(Modeling Rule)" 有四种类型: "无规则(None)"、"强制规则(Mandatory)"、"强制共享规则 (MandatoryShared)"和"可选规则(Optional)"。

新建变量	_ 🗆 X
浏览名称	Level
描述	液位
数据类型	Double
维度	Scalar 🔹
保存历史数据	
类型定义	DataItemType
建模规则	Mandatory 🔹
启用批量模式	None
	Mandatory
	Optional
	MandatoryShared

图 4-9 建模规则

**无规则**(None):表示该项是该对象类型的说明(比如作者或版本等),不属于该对象类型的变量或者方法等,不会被该对象类型的实例所继承。

**强制规则**(Mandatory):表示该项是该对象模型的所有对象实例中必须有的内容,所有基于该对象类型构建的实例均会继承该项。

**强制共享**(MandatoryShared): 表示该项是该对象模型的所有对象实例的公 共部分,所有基于该对象模型构建的实例均共享该对象模型中的此变量,语义上 类似于高级编程语言中的静态变量。

**可选**(Optional): 表示该项是该对象模型的实例中可有可无的项, 基于该对 象模型构建的实例默认不继承此项, 如果需要在实例中继承该项可以手动在实例 中添加。

**启用批量模式**:可以批量建立变量,新建变量的尾号将从指定的开始值编号, 到结束值终止。注意: 启用批量模式之后,浏览名称中必须使用 C#语言支持数 字类型的格式化模板,如图 4.10 中浏览名称中的"{0}"将被相应的数字编号所 替代。如果希望在数字前面补 0,可以使用类似"{0:D3}"这样的格式化模板。 此外,描述字段中也可使用这样的格式化模板。

新建变量	_   ×
浏览名称	Level(0)
描述	
数据类型	Double
维度	Scalar 👻
保存历史数据	
类型定义	DataItemType
建模规则	Mandatory 💌
启用批量模式	
批量起始值	1 🔺
批量结束值	III -
批量步进值	1 🔺
	确定 取消
	Tank Tank Evel1 Evel2 Evel3 Evel4

图 4-10 批量模式

将以上所有项目均选好后,在新建变量窗口中单击"确定"则完成一个或一 组变量的创建。

## 4.1.2 新建属性(New Property)

相对于变量,属性一般保存配置信息。属性可以是系统预置的属性,也可以 是用户自定义的属性。

单击"新建属性"后弹出对话框(如图 4.11 所示)。如果要定义用户自己定 义的属性,需要填入相应的浏览名称(BrowseName)、描述(Description),选 择合适的数据类型(DataType)、数据维度(ValueRank)、建模规则(Modeling Rule) 后单击"确定(OK)"按钮完成属性新建,弹出属性编辑窗口(如图 4.12 所示)。

## AicDataHub 用户手册

New Property	_ 🗆 X
BrowseName	
Description	
DataType	String
ValueRank	Scalar 🔹
Type Definition	PropertyType
Modelling Rule	MandatoryShared 🔹
	OK Cancel

图 4-11 新建属性界面

NewProperty-1		
Parent Inform	ation	
Parent Node	2:NewObject_1	
Reference Type	HasProperty	
Node Attribut	tes	
NodeClass	Variable	
NodeId	ns=3;i=2168	
BrowseName	2:NewProperty-1	
DisplayName	NewProperty-1	
Description	•	
WriteMask	1577587	
DataType	String	
ValueRank	Scalar 🔹	
Value	(null)	
AccessLevel	3	
<ul> <li>Instance Deck</li> </ul>	aration	
Type Definition	PropertyType	
Modelling Rule	MandatoryShared 🔹	

图 4-12 属性编辑界面

AicDataHub 用户手册

Edit AccessLevel		X	ζ.
Access Level	3		
Access Level 3 CurrentRead CurrentWrite HistoryRead HistoryWrite			
	ОК	Cancel	

图 4-13 访问级别选择窗口

在新建属性时,点击"浏览名称(BrowseName)"后面的选择按钮…,,会弹出系统预定义的属性。

系统预定义的类属性包括:

- 节点版本 (NodeVersion)
- 图标 (Icon)

系统预定义的对象属性包括:

- 节点版本 (NodeVersion)
- 图标 (Icon)
- 命名规则(NamingRule)

系统预定义的变量属性包括:

- 定义 (Definition)
- 值精度(ValuePrecision)
- 节点版本 (NodeVersion)
- 本地时间(LocalTime)
- 允许为空(AllowNulls)
- 工程单位(EngineeringUnits)
- 工程单位量程(EURange)
- 仪表量程(InstrumentRange)
- 历史记录设置(HistorizingConfiguration)

这些预定义的属性将在后续的属性章节中具体介绍。

# 4.1.3 新建方法(New Method)

单击后弹出对话框(如图 4.14 所示)。填入相应的"浏览名称(BrowseName)" 和"描述(Description)",选择合适的"建模规则(ModelingRule)"后单击"确 定(OK)"按钮即可完成方法的新建,进入方法的编辑窗口(如图 4.15 所示)。

新建方法	_ = ×
浏览名称	IncreaseLevel
描述	提升液位
建模规则	MandatoryShared 🔹
	确定 取消

图 4-14 新建方法

默认的编辑界面如下图,

```
MethodScript
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Diagnostics;
6 using Opc.Ua;
7 using Opc.Ua.Server;
8 using AicTech.DataHub.Common;
9 using AicTech.DataHub.Model;
10 using AicTech.DataHub.Utilities;
11
12 namespace AicTech.DataHub.Server.Generated
13 {
14
      /// <summary>
15
      /// MethodScripts class.
16
      /// </summary>
      public static partial class MethodScripts
17
18
      ł
19
          /// <summary>
          /// Method script.
20
          /// </summary>
21
          /// <remarks>
22
          /// * Output arguments must be 'out' parameters.
23
          /// * Input Arguments and Output Arguments must be with Da
24
25
          /// </remarks>
          public static Opc.Ua.ServiceResult OnCallMethod(
26
              Opc.Ua.Server.IScriptExecuteContext scriptContext,
27
28
              Opc.Ua.IOperationContext
                                                   operationContext,
                                                   objectId
29
              Opc.Ua.NodeId
              // TODO: INPUT_ARGUMENTS
30
31
              // TODO: OUTPUT ARGUMENTS
32
              )
33
           {
```

#### 图 4-15 方法编辑界面

在编辑界面修改方法,把Level值加1,修改后的方法脚本如下:

```
public static Opc.Ua.ServiceResult OnCallMethod(
   Opc.Ua.Server.IScriptExecuteContext scriptContext,
Opc.Ua.IOperationContext operationContex
                                     operationContext.
   Opc.Ua.NodeId
                                     objectId
{
   var curNode = scriptContext.AddressSpace.FindNode<IServerNode>(objectId);
   var systemContext = scriptContext.SystemContext;
   try
   {
       //得到液位节点
                                                          true,null,false);
       //得到液位值
       var levelValue = (double)levelNode.Value;
       //増加液位值
       levelValue = levelValue + 1;
       //回写液位值
       var aResult = levelNode.WriteAttribute(systemContext,
                                          Attributes.Value,
                                          NumericRange.Empty,
                                          new DataValue(new Variant(levelValue),StatusCodes.Good,DateTime.UtcNow));
       if(ServiceResult.IsGood(aResult))
           curNode.ClearChangeMasks(systemContext.true);
   catch(Exception ex)
       Utils.Trace(ex, "IncreaseLevel");
   return Opc.Ua.StatusCodes.Good;
```

#### 图 4-16 修改后的方法

此方法可以由对象继承,并通过对象调用。

#### 4.1.4 新建报警(New Alarm)

右键单击对象类型,选择新建报警,弹出新建报警对话框(如图 4.17 所示), 填入相应的"浏览名称(BrowseName)"、"描述(Descrip)",选择合适的"事件 通知(EventNotifier)"、"类型定义(Type Definition)"和"建模规则(Modeling Rule)",然后点击"确定(OK)"按钮即可出现报警的编辑窗口(如图 4.19 所示)。

Í	新建对象		х
10	浏览名称	LevelAlarm	
	描述	液位报警	
j	事件通知标志	1	
5	类型定义	ExclusiveLevelAlarmType	
	建模规则	Mandatory	•
-	启用批量模式		
1		海宁 即巡	
		MALE AXIA	

#### 图 4-17 新建报警

点击"类型定义"后面的选择按钮…,弹出报警类型选择窗口,

选择ObjectType	_ 🗆 X
<ul> <li>AlarmConditionType</li> </ul>	
► 🗠 DiscreteAlarmType	
🖌 🗠 LimitAlarmType	
ExclusiveLimitAlarmType	
<ul> <li>ExclusiveDeviationAlarmType</li> </ul>	
ExclusiveLevelAlarmType	
ExclusiveRateOfChangeAlarmType	
► 🗠 NonExclusiveLimitAlarmType	
确定	取消
MUAL	-Mina

图 4-18 选择报警类型

默认选择单一限值报警,确认后,进入报警编辑界面,下图显示单一限值报 警的编辑界面:

报警信息 ————		
名称	LevelAlarm	
源		
报警激活延时	0	毫秒
报警解除延时	0	毫秒
报警解除消息		
死区	0.00	
启用		
报警解除要求确认		

图 4-19 报警编辑界面

点击源质	后面的选择按钮, 弹出报警源设置窗口			
修改源		-		Х
绝对节点	相对节点表达式			
🔺 🀼 Tan	k			
- 🔧	LevelAlarm			
_ <u>_</u>	Level			
- 7	Temperature			
- =0	IncreaseLevel			
属性	Value			Ŧ
节点路径	/2:Level			_
and a state of the last		_	-	

图 4-20 选择报警源

在相对节点中,选择对象类型 Tank 下的 Level 作为报警源。然后设置高高、高、低、低低的限值及报警信息,示例如下:

- ŠŠ	
启用	
要求确认	
比較运算符	>
限值	95.00
报警激活消息	·····································
<b>灰白</b> 灰白/月月日	
, =u	1~u
- 高	
启用	
要求确认	
比较运算符	> •
限值	90.00
报警激活消息	液位高
严重性	中等
_ 低	
启用	
要求确认	
比较运算符	< *
限值	10.00
报警激活消息	液位低
严重性	中等

图 4-21 设置报警条件

关于报警的更多内容会在后边"报警管理(Alarms)"一章中有详细的介绍。

# 4.2 高级功能

# 4.2.1 子对象一新建对象

类型下面可以包含子对象,通过子对象可以构建层次化的对象模型。比如可 以创建一个发酵车间类型,其中包含两个罐对象,每个罐对象都是之前建立的罐 类型。

新建发酵车间类型:

新建对象类型		_ 🗆 X
浏览名称	FermentWorkshop	
描述	发酵车间	
	确定	取消

图 4-22 新建子类型

右键点击发酵车间,点击新建对象:

新建对象	- 🗆 ×
浏览名称	Tank1
描述	1#讙
事件通知标志	1
2 类型定义	Tank
建模规则	Mandatory 🔻
启用批量模式	
	确定取消

图 4-23 新建对象

在这里,"类型定义(Type Definition)"可以选择已经建立的 Tank 类型。 建好后的发酵车间对象类型如下所示:



## 4.2.2 继承和重载一新建对象类型

对象类型可以新建子类型,子类型会继承父类型的所有特性,同时也可以扩展和重载父类型。例如,我可以在之前建立的 Tank 类型下,新建子类型 FreezingTank (冷冻罐)。右键点击对象类型 Tank,在弹出的右键菜单中选择新 建对象类型,

新建对象类型		_ 🗆 X
浏览名称 描述	FreezingTank 冷冻罐	
	确定	取消

图 4-25 新建子类型

然后,给新建的子类型 FreezingTank 扩展变量 ColdWaterFlow。新建之后的 类型如下图所示:



图 4-26 类型层次

然后,在FermentWorkshop类型中,新建子对象FreezingTank3,

新建对象		Х
浏览名称	FreezingTank3	
描述		
事件通知标志	1	]
类型定义	FreezingTank	
建模规则	Mandatory	•
启用批量模式		
	· 御完 · 取消	
	WOAE -54143	

图 4-27 新建子类型对象

可以看到, FreezingTank3 除了 Tank 类型定义的内容之外,多了新定义的 ColdWaterFlow 变量。



图 4-28 子类型对象

#### 4.2.3 解除到父对象关联

在OPC UA 的地址空间中,一个节点可以有多个父节点。比如新建对象 Tank1 后,它的父节点是"工厂对象"。随后可以把 Tank1 添加到报警区,此时,Tank1 就有第二个父对象一报警区根节点。

解除到父对象关联,就是解除不再需要的某个父对象关联。

删除对象时,将自动解除和所有父对象的关联。

## 4.2.4 新建脚本

和方法被动调用不同,脚本可以设定执行条件,当条件满足时自动执行。可 以在对象类型下建立脚本,包括条件脚本、数据变化脚本和定时器脚本。

比如我们要计量 Tank 类型的容积(Volume),但没有直接的计量装置。只能 靠液位来换算。为了实现这一功能,我们给 Tank 类型增加一个数据变化脚本。

右键点击 Tank 类型,选择新建数据变化脚本,

新建脚本	_ = ×
浏览名称	OnLevelChangeScript
描述	液位变化脚本
	确定    取消
L	

## 图 4-29 新建脚本

点击"源"后面的选择按钮,选择相对节点的 Level:

•

## 图 4-30 选择脚本源

脚本内容如下:

AicDataHub 用户手册

```
源 /2:Level
                                                  ...
            /// </summary>
21
22
            public static Opc.Ua.ServiceResult OnDataChanged(
23
                Opc.Ua.Server.IScriptExecuteContext scriptContext,
24
                object
                                                        newValue,
25
                DateTime
                                                        newTime,
                Opc.Ua.StatusCode
                                                        newCode)
26
27
            {
28
                var systemContext = scriptContext.SystemContext;
                var curNode = scriptContext.ContextNode;
29
30
31
                trv
32
                {
33
                     //get level value
                    var qnLevel = new QualifiedName("Level",2);
34
                    var nodeLevel = curNode.FindChild<IServerVariableNode>(systemContext,
35
36
                                                                                 gnLevel,
37
                                                                                 ReferenceTypes.Aggregates,
38
                                                                                 true.
                                                                                 null.
39
40
                                                                                 false);
                    var vLevel = (double)nodeLevel.Value;
41
42
43
                    //calculate volume
44
                    double vVolume = 0.0;
                    if(vLevel < 20)</pre>
45
                         vVolume = 10*vLevel;
46
47
                    else
48
                         vVolume = 10*20+ 20*(vLevel-20);
49
50
                   //set volume value
51
                   var qnVolume = new QualifiedName("Volume",2);
                   var nodeVolume = curNode.FindChild<IServerVariableNode>(systemContext,
52
53
                                                                             qnVolume,
                                                                            ReferenceTypes.Aggregates,
54
55
                                                                            true,
56
                                                                            null.
57
                                                                             false);
58
                   var aResult = nodeVolume.WriteAttribute(systemContext,
59
                                                            Attributes.Value,
                                                            NumericRange.Empty,
new DataValue(new Variant(vVolume),
60
61
                                                                          StatusCodes.Good,DateTime.UtcNow));
62
63
                   if(ServiceResult.IsGood(aResult))
64
                   {
65
                       nodeVolume.ClearChangeMasks(systemContext,true);
                   }
66
67
68
                   var scirptNode = scriptContext.ScriptNode;
69
70
               }
71
               catch(Exception ex)
72
73
               {
                   Utils.Trace(ex,"OnLevelChangeScript:{0}",ex.Message);
74
                   return Opc.Ua.StatusCodes.BadUnexpectedError;
75
               }
76
77
               return Opc.Ua.StatusCodes.Good;
78
          }
79
      }
80 }
81
```

#### 图 4-31 编辑脚本

脚本的更详细解释请见后续第7章"脚本"。

#### 4.2.5 配置导出和导入

#### 导出文件

此功能用于将所选对象或文件夹所包含的变量导出到外部 Excel 文档,单击 弹出对话框(如图 4.33 所示)。
导出		_ 🗆 X
文件路径		
完成时打开文件		
导出对象(递归)		
	确定	取消

图 4-32 导出

选择合适的路径后点击"确定(OK)"按钮即可完成导出。

"完成时打开文件(OpenFileWhenCompleted)":此选项默认为选中,当导出操作时会自动打开导出文档。

"导出对象(递归)":将递归导出所有子对象的变量。

导入文件

此功能用于从外部 Excel 文档批量导入变量。

导入	_ 🗆 X
文件路径	[
首列为选中标志	
仅导入首个工作表	
	确定 取消

#### 图 4-33 导入

导入的 EXCEL 文档格式要按照如下格式来写(如图 4.34),第一行写变量 的各种属性,对应的属性列下边写各自具体的值,变量的"浏览名称 (BrowseName)"和"数据类型(DataType)"在 Excel 表格中必须指定,否则会 导入失败。对于其它属性指定与否均可,若没有指定,则导入 DataHub 后这些 属性会被赋予默认值。注意:目前的导入/导出功能有限,仅能处理变量的导入/ 导出,并且无法处理多级的树形结构,我们将在后续的版本升级中予以改进。

	B78 🔻 💽	<i>f</i> ∝ Double	
	A	В	
1	BrowseName	DataType	
2	耗油时间	Int32	
3	耗油重量	Int32	
4	油杯总重	Int32	
5	fujil	Int32	
6	fuji3	Int32	
7	fuji5	Int32	
8	fuji7	Int32	
9	fuji9	Int32	
10	温度1	Double	
11	温度2	Double	

图 4-34 导入 Excel 文档的格式

# 4.2.6 应用类型修改

应用类型定义变化(Apply TypeDefinition Changes)选项用于把对象类型的 变化更新到所有该类型的对象实例上。

注意:此功能仅适用于对象类型新增强制或强制共享规则(Mandatory/ MandatoryShared)的成员时。当对象类型的成员增加时执行此功能,会将所有 对象实例的成员也会相应的增加;如果对象类型的成员减少时执行此功能,则所 选对象的相应成员不会自动删除(即不发生变化)。

# 5 工厂对象 (PlantObjects)

工厂对象是类型的具体实例。

# 5.1 新建对象

在主界面上右击工厂对象文件夹(PlantObject)选择新建对象(NewObject) 或新建文件夹(New Folder),弹出新建对象对话窗口(如图 5.1 所示)。

新建对象	_ 🗆 X
浏览名称	FermentWorkshop1
描述	1#发酵车间
事件通知标志	1
类型定义	FermentWorkshop
建模规则	None 🔻
启用批量模式	
	<b>确</b> 完 取消
	WOAE *NIH

图 5-1 新建对象

点击类型后面的选择按钮,弹出类型选择对话框:

选择ObjectType	-		x
<ul> <li>CustomBaseObjectType</li> </ul>			
- 🖾 FermentWorkshop			
🕨 🌌 Tank			
确定	取	Ϊ	

图 5-2 选择类型

可以选择之前建立的对象模型。

启用批量模式:

可以选择批量建立对象,如下图所示:

新建对象	_ 🗆 X
浏览名称	FermentWorkshop (0)
描述	(0)#发酵车间
事件通知标志	1
类型定义	FermentWorkshop
建模规则	None
启用批量模式	
批量起始值	1
批量结束值	2
批量步进值	1 🔺
	确定取消

图 5-3 批量模式

选择批量起始值,结束值,步进值。批量建立的对象结果如下:



# 5.2 对象管理

右键点击对象,弹出对象管理的功能菜单。

AicDataHub 用户手册



图 5-4 对象管理菜单

可以看出,对象的管理功能和类型的功能类似。也就是说,可以在对象上扩 展新的变量、方法、属性、报警以及脚本等功能。具体操作可以参考上一章类型 章节,此处不再重复。

# 5.2.1 应用类型定义变化(Apply Type Definition Changes)

此选项用于把类型的修改应用到本对象,更常见的做法是在对象类型上应用 此功能,将类型变化更新到所有类型实例上。

注意:如果对象类型的成员增加时选择此项会将所选文件夹或对象的成员也 会相应的增加。但是如果父类型的成员减少时选择此项,则所选文件夹或对象的 成员不会减少。 AicDataHub 用户手册

# 6 报警(Alarms)

报警可以建在类型上,由对象继承;也可以建在对象上;还可以单独建立(全局报警)。在类型上和对象上建立报警见前面章节。本章中将详细介绍各种报警 类型、报警区以及全局报警。

#### 6.1 报警类型

报警类型分离散报警(数字量)和限值报警(模拟量),具体如下图所示:



#### 图 6-1 报警类型

数字量报警分两种类型:系统异常报警、跳变报警。模拟量报警分三种类型: 越限报警、变化率报警和偏差报警。每种模拟量类型又分单一和非单一报警。

### 6.1.1 系统异常报警(SystemOffNormalAlarmType):

系统异常报警是在变量或表达式出现异常时产生的报警。

┌ 报警信息 ────		
名称	EmergencyStopAlarm	
源	/2:EmergencyStopSignal	
报警激活延时	0	鼋秒
报警解除延时	0	毫秒
报警解除消息		
启用		
报警解除要求确认		
离散报警		_
比较运算符	•	
比较目标值		
报警激活消息	急停信号为真	
严重性	中等 •	
触发类型	真 •	
要求确认		
1		

图 6-2 系统异常报警

报警源:报警源可以是某个变量,也可以是一个表达式。点击源后面的选择按钮, 弹出报警源选择窗口。

报警源是一个变量:

修改源		-		х	
绝对节	点 相对节点 表达式				
i 🖌 🖂	RollMachine				
►	쓚 EmergencyStopAlarm				
	TemergencyStopSignal				
	🚰 Speed				
	Temperature				
1					
-					
2					
5					
属性	Value			•	
节点路	径 /2:EmergencyStopSignal				
	确定	取消	í		

图 6-3 报警源

报警源是一个表达式:

修改源	_ 🗆 X
绝对节点表达式	
<pre>{{np::/Objects/2:System/2:PlantObjects/2:FermentWorkshop1/2:Tank1/2:Level}} &amp;&amp;{{np::/ Objects/2:System/2:PlantObjects/2:FermentWorkshop1/2:Tank1/2:Temperature}}&gt;</pre>	<ul> <li>·5 ▲ 检查表达式</li> <li>·6 插入节点属性</li> </ul>

#### 图 6-4 报警源是一个表达式

**报警延时**:单位为秒。如果设置延时,当满足/解除报警条件后,并不马上产生/ 解除报警,而是开始计时,当计时时间等于或超过所定义的延时,如果报警还存 在,则显示报警信息。如果在这段时间内,产生了新类型的报警,则计时重新开始。报警解除延时类似。

**启用:**只有勾选启用,报警设置才起作用,新建报警默认不启用。

**比较运算符:**比较报警源和目标值的运算符。如果报警源本身是一个 BOOL 表达式,则不需要设置运算符和目标值。

比较运算符	解释
=	报警源等于目标值
!=	报警源不等于目标值
>	报警源大于目标值
<	报警源小于目标值
>=	报警源大于等于目标值
<=	报警源小于等于目标值
IsNull	报警源为空(无值)
Match	报警源(字符串)和目标值(字符串或正则表达式)匹
	酉己

比较目标值:目标值。

触发类型:

触发类型	解释
真	报警条件结果为真
假	报警条件结果为假
变化	报警条件结果变化

**报警激活消息**:报警激活的消息。可以是一个字符串,也可以是一个变量或报警 对象的某个属性。

报警消息是一个字符串:见前例,消息为固定字符串"急停信号为真"。 报警消息是变量或某个报警属性:点击报警激活消息后面的选择按钮,弹出报警 消息选择对话框:

急停信号为真	编辑报警消息		_ 🗆 X
▲ 取消	急停信号为真		重 山 人 插入节点属性 插入报警参数
<b>W</b> 02 0013		•	取消

#### 图 6-5 编辑报警消息

选择插入节点,可以选择某个变量。

选择插入报警参数,弹出对话框选择报警参数:

选择报警参数	_ ×
名称 て	描述で
ALARM_VALUE	报警值
ALARM_TIME	时间戳
ALARM_CODE	质量戳
ALARM_TYPE	报警类型
ALARM_SEVERITY	严重性
TARGET_VALUE	目标值(仅适用偏差报警)
SOURCE_VALUE	源值
SOURCE_NAME	源对象名称
LIMIT_VALUE	限值(仅适用限值报警)
DELAY_IN	报警激活延时
DELAY_OUT	报警解除延时
DEADBAND	死区
CONDITION_NAME	报警条件名称
CURRENT_TIME	当前时间戳
	确定 取消

### 图 6-6 选择报警参数

**要求确认:**指明此报警需要人工确认。当此选项启用之后,如果没有人工对报警进行确认,即使报警条件消除,客户端仍能查看到此报警消息。

# 6.1.2 跳变报警(TrigAlarmType):

┌ 报警信息 ────		
名称	TrigAlarm1	
源		
报警激活延时	0	室秒
报警解除延时	0	室秒
报警解除消息		
启用		
报警解除要求确认		
离散报警 ————		
比较运算符	•	
比较目标值		
报警激活消息		
严重性	中等	
触发类型	真 •	
要求确认		

图 6-7 跳变报警

触发器报警类似于系统异常报警,具体请参考前文。

# 6.1.3 单一越限报警 (ExclusiveLimitAlarmType):

限值报警是模拟量的值在跨越报警限时产生的报警。

AicDataHub	用	户手册
------------	---	-----

┌ 报警信息 ────		
名称	LevelAlarm	
源	/2:Level	
报警激活延时	0	毫秒
报警解除延时	0	毫秒
报警解除消息		
死区	0.00	
启用		
报警解除要求确认		
- 尚尚		
启用		
要求确认		
比较运算符	> •	
限值	95.00	
报警激活消息	液位极高	
严重性	<b>南</b> •	

AicDataHub 用户手册

ä		
启用		
要求确认		
比较运算符	>	•
限值	90.00	* *
报警激活消息	液位高	
严重性	中等	•
低		
启用		
要求确认		
比较运算符	<	•
限值	10.00	* *
报警激活消息	液位低	
严重性	中等	•
低低		
启用		
要求确认		
比较运算符	<	•
限值	5.00	*
报警激活消息	液位极低	
严重性	高	*

图 6-8 单一越限报警

越限报警的报警限(类型)有四个:低低限、低限、高限、高高限。这些值在 变量的最大值和最小值之间,它们的大小关系排列依次为高高限、高限、低限、 低低限。在变量的值发生变化时,如果跨越某一个限值,立即发生越限报警。

单一越限报警在某个时刻,只产生一种越限报警,例如:如果变量的值超过 高高限,就会先复位清除高限报警,再产生高高限报警,而不会同时产生两种报 警。另外,如果连续发生越限,就得看连续两次越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,小于等于低低限时报警

大于低限时恢复,小于等于低限时报警 大于等于高限时报警,小于高限时恢复 大于等于高高限时报警,小于高高限时恢复



#### 图 6-9 报警限

例如:设定某锅炉控制温度的高高报警值=150,高报警值=130,低报警值= 50,低低报警值=30,则当锅炉温度变化时会出现以下控制情况:

锅炉温度>=150 高高报警

130=<锅炉温度<150 高报警

50<锅炉温度<130 正常工作状态

30<锅炉温度<=50 低报警

锅炉温度<=30 低低报警

**越限死区:**是指如果变量的值在次报警限加减死区值的范围内,就不会产生新的报警,如果变量的值超过此范围内,则先恢复原来的报警,再产生新报警。 越限死区的作用是避免变量值在限值附近时,由于信号扰动频繁产生报警。

例如: 变量的高限=80,高高限=90,死区值为 2。则当变量值为 81 时并 不产生报警,当变量值为 82 时产生高报警;变量值为 92 时,复位高限报警, 产生高高限报警。当变量减小到 87 时,才复位高高限报警,产生高限报警。当

变量减小到77时,才复位高限报警。

# 6.1.4 单一变化率报警(ExclusiveRateOfChangeAlarmType):

变化率报警是模拟量值在单位时间内的变化速度超过一定量时产生的报警,即变量变化太快时产生的报警。当模拟量的值发生变化时,就计算变化率以决定 是否报警。

┌ 报警信息 ────		
名称	TempChangeRateAlarm	
源	/2:Temperature	
报警激活延时	0	毫秒
报警解除延时	0	毫秒
报警解除消息		
时间类型	分 •	
死区	0.00	
启用		
报警解除要求确认		
使用绝对值		
大变化 ————		
启用		
要求确认		
比较运算符	> •	
限值	3.00	
报警激活消息	温度变化极快	
严重性	高 •	
_ 小变化		
启用		
要求确认		
比较运算符	> •	
限值	1.00	
报警激活消息	温度变化快	
严重性	中等	

图 6-10 单一变化率报警

变化率的时间单位有三种:秒、分和时。变化率报警利用如下公式计算:

((变量的当前值 - 变量上一次的值)\*100)/((本次产生值的时间

-上次产生值的时间)\* (最大值 - 最小值)\* 单位对应的值

(注:如果是秒,为1;如果是分,为60,如果是时,为3600))

注意:上图中选中了"使用绝对值",因此上述计算值取绝对值作为计算结果,若计算结果大于定义的变化率的值,则出现报警。如果不选中"使用绝对值",则直接使用上述计算值作为计算结果与设定值进行比较,并且会多出两个报警项:"负的小变化"和"负的大变化",类似于限值报警中的"低限"和"低低限"。

# 6.1.5 单一偏差报警 (ExclusiveDeviationAlarmType):

偏差报警就是模拟量的值相对偏差目标值上下波动超过一定量时产生的报警。

报警信息 ————		
名称	TemperatureDeviationAlarm	]
源	/2:Temperature	
报警激活延时	0	室秒
报警解除延时	0	室秒
报警解除消息		
死区	0.00	}
目标值	25	
启用		
报警解除要求确认		
使用绝对值		
大偏差		
启用		
要求确认		
比较运算符	> •	]
限值	2.00	}
报警激活消息	温度偏差太大	
严重性	Ē. ▼	]

AicDataHub 用户手册

大偏差		
启用		
要求确认		
比较运算符	> •	
限值	2.00	
报警激活消息	温度偏差太大	
严重性	高 🔹	
小偏差		
启用		
要求确认		
比较运算符	> •	
限值	1.00	
报警激活消息	温度偏差大	

图 6-11 单一偏差报警

目标值: 目标值可以是一个绝对变量、相对变量, 或者是一个常量。

修改目标		-		х
绝对节点 相对节点 常量值				
数据类型	Double 🔹			
值	25			

#### 图 6-12 目标值

在选中"使用绝对值"选项的情况下,偏差报警分为大偏差和小偏差两种。 当波动的值的范围大于大偏差时产生大偏差报警;小于大偏差、大于小偏差时, 产生小偏差报警。示意图如下:

AicDataHub 用户手册



图 6-13 偏差报警限

注:图中实线表示报警,虚线表示报警恢复

偏差报警限的计算公式为:小偏差报警限=偏差目标值±小偏差报警设定值 大偏差报警限=偏差目标值±大偏差报警设定值

由于偏差有正负,在偏差范围内相对目标值(基准值)上下波动的模拟量最 小分界值称为最小当前值,相对目标值(基准值)上下波动的模拟量最大分界值 称为最大当前值,例如:

某变量的最小值=-100,最大值=100,设定其小偏差=50,大偏差=70,目标 值=0,则可计算出小偏差报警和大偏差报警的条件如下:

(1) 小偏差报警:

小偏差报警上限=0+50=50

小偏差报警下限=0-50=-50

则:模拟变量值>=50 或模拟变量值<=-50 时,出现小偏差报警。

(2) 大偏差报警:

大偏差报警上限=0+70=70

大偏差报警下限=0-70=-70

则: 模拟变量值>=70 或模拟变量值=<-70 时,出现大偏差报警,同时恢复 小偏差报警。

注意: 若不选中"使用绝对值", 将直接使用源值与目标值求差结果与定义

限值进行比较,此时会多两个报警选项:"负的小偏差"和"负的大偏差",类似 于限值报警中的"低限"和"低低限"。

## 6.1.6 非单一报警(复合报警)

偏差报警、液位报警和变化率报警除了有单一类型的报警之外还有非单一 (复合)类型的报警。所谓复合类型报警就是当出现更为严重的报警时原先的低 一级的报警不会消失,而是同时存在。比如一个变量出现高高报警时其原有的高 报警不会消失,而是高高报警和高报警同时存在。

# 6.2 报警区域

报警对象按区域管理。各级对象本身可以作为报警区,比如一个车间,一个 加工中心或一台设备作为一个报警区。除了对象外,还可以建立逻辑报警区。

报警区可以分层,层级没有限制。报警将自下而上,逐级通知给自己的各级 报警区。

一个对象可以同时属于不同的报警区。

# 6.2.1 报警根区域

系统会默认建立报警根区域,点击报警区域文件夹左边的小三角会展开一个 根区域(RootArea)子目录(如图 6.14 所示)。



图 6-14 报警区展开界面

右击选择编辑(Edit)会打开根区域的信息显示窗口(如图 6.15 所示),这 个根区域的信息是系统设定的,不可更改。

▲ 父对象信息	
父节点	2:Areas
引用类型	Organizes
◇ 节点属性	
节点分类	Object
节点ID	ns=2;i=63
浏览名称	2:RootArea
显示名称	▼ RootArea
描述	•
可写标志	0
事件通知标志	13
◇ 实例声明	
类型定义	BaseObjectType
建模规则	None *

图 6-15 报警根区编辑界面

### 6.2.2 添加新的报警区

在根区域图标上右击选择"新报警区域(New Area)"(如图 6.16 所示),弹 出新建报警区域对话框(如图 6.17 所示)。填入相应的"浏览名称(BrowseName)"、 "描述(Description)",选择合适的"事件通知(EventNotifer)"、"类型定义(Type Definition)"和"建模规则(Modeling Rule)",然后点击"确定(OK)"按钮即 可弹出新建的报警区域编辑窗口(如图 6.18 所示)。

如果点击"取消(Cancel)"按钮则会退出新建报警区域对话框。

对于已有的报警区域的编辑和内容的添加可以参照根报警区域的编辑和为根报警区域添加内容的操作步骤。

▶ 🗀工厂对象				
▶ 🚞 对象类型				
⊿ 🧰 报警				
🖌 🚞 Area	s			
- 🕂 R		复制	Ctrl+C	
🕨 📜 Conc	v	100.00	culiv	
▶ 🚞脚本	đ	努切	Ctrl+X	
▶ 🚞安全	<b>E</b>	粘贴	Ctrl+V	
	2	刷新		
		编辑		
	G.	新建报警团	Σ	
	6	添加已存在	E对象到报警区	

图 6-16 报警区菜单

新建报警区:	:
--------	---

新建对象	- 🗆 X
浏览名称	Plant1
描述	1分厂
事件通知标志	1
类型定义	CustomBaseObjectType
建模规则	None 🔻
启用批量模式	
	确定取消
L	

#### 图 6-17 添加新报警区

# 6.2.3 添加已存在的对象到报警区

在根报警区域图标上右击选择"添加已存在对象到报警区(Add Existing Object Area)"(如图 6.16 所示),弹出"选择项目(Select Item)"窗口(如图 6.18 所示),窗口中列出了所有"设备对象(PlantObjects)"中已有的对象实例,选择好相应的对象区域后单击"确定(OK)"按钮则完成添加。

ĺ	选择项		х	)
11H	- 节点树			
	A 🗁 System			
	► 🚞 Alarms			
	- 🔧 GlobalMethods			2
t	🖉 🗁 PlantObjects			
	FermentWorkshop1			
	► 🔧 RM1			
1	► 🔚 Scripts			
1	► 🔚 Security			
1				
F				
F				
ŧ				
×				
1				
ş				
ī				
	确完 取深	ť		
		-		

### 图 6-18 添加已有的对象到报警区

## 6.3 独立报警

右键点击报警树下的报警条件,选择"新建报警(NewAlarm)"(如图 6.19 所示),则会弹出新建独立/全局报警对话框(如图 6.20 所示)。

模型浏览器					ņ	×
▶ 🗀工厂对象						
▶ 🧀 对象类型						
⊿ 🧰 报警						
🖌 🚞 Areas						
🖌 🔧 RootA	rea					
📙 🦾 🕂 Fe	rmer	ntWorksho	p1			
Condition		-		]		
▶ □ 脚本	43	复制	Ctrl+C			
▶ □□安全	*	剪切	Ctrl+X			
	<u></u>	粘贴	Ctrl+V			
	2	刷新				
	<b>€</b> €	新建报警				

#### 图 6-19 新建报警

新建对象	_ 🗆 X
浏览名称	
描述	
事件通知标志	1
类型定义	ExclusiveLevelAlarmType
建模规则	None 🔻
启用批量模式	
-	确定 取消

图 6-20 新建报警

独立报警的报警源必须使用绝对节点,详细过程见前面章。此外,独立报警 的报警对象也必须添加到报警区中,相应的报警才能被客户端正确的接收。

# 7 脚本 (Scripts)

脚本可以建立在模型上,由对象继承;也可以直接建立在对象上;还可以建 立全局的脚本。前两者既能操作相对节点,又能操作绝对节点;全局脚本只能使 用绝对节点。本章以全局脚本为主,间或以类型脚本为例,不再分别介绍这三种 情况。

左击模型浏览器中"脚本 (Scripts)"文件夹,则展开其子目录 (如图 7.1 所示),分别为"条件脚本 (ConditionalScripts)"、"用户功能脚本 (CustomFunctions)"、"数据改变脚本 (DataChangeScripts)"、"系统脚本 (SystemScripts)"和"引用集合 (ReferenceAssemblies)"。



图 7-1 脚本展开

# 7.1 常用上下文

所有的脚本都会有一个类型为 Opc.Ua.Server.IScriptExecuteContext 的上下文 参数 scriptContext,此参数对象给脚本提供了访问 DataHub 服务器内部数据的 API 接口,以及脚本执行时的上下文信息。此上下文对象中包含如下内容:

- ♦ AddressSpace: 类型为 Opc.Ua.Server.IServerAddressSpace, 是服务器的 地址空间访问接口。
- ◆ ServerObject: 类型为 Opc.Ua.Server.IServerObject, 是服务器内部对象访问接口。

- ◆ ScriptNode: 类型为 Opc.Ua.Server.IServerVariableNode, 是存储脚本代码的变量节点,即脚本节点自身。
- ◆ ContextNode: 类型为 Opc.Ua.Server.IServerNode,是脚本的上下文节点。 如果是对象上的脚本(包括继承自对象类型的脚本),上下文节点即为 对象节点;如果是全局脚本,上下文节点为相应脚本类型文件夹对象, 例如 ConditionalScripts 文件夹。
- ♦ SystemContext: 类型为 Opc.Ua.Server.ServerSystemContext, 是系统上下 文对象, 在某些服务器内部 API 中需要此参数。
- ◆ ScriptData: 类型为 object,是供脚本内部使用的自定义数据,可以使用 此参数在多次的脚本执行间传递/共享数据,此参数只由脚本设置和读 取。

除了上述的 scriptContext 参数之外,对于某些特定类型的脚本,还会传递额外的参数作为上下文:

- ◆ session: 类型为 Opc.Ua.Server.Session,它提供了会话相关信息,仅用于 "会话激活"和"会话关闭"系统脚本。
- ◆ eventData: 类型为 Opc.Ua.IFilterTarget, 它提供了报警和事件的相关信息, 仅用于"事件"系统脚本。
- ◆ operationContext: 类型为 Opc.Ua.IOperationContext, 它提供了 OPC UA 服务调用的上下文信息, 仅用于"方法"脚本。

### 7.2 脚本常用函数

在脚本中,常用到的需要访问地址空间和操作节点对象的函数,主要有以下 几个(关于这些函数的详细描述参见本文结尾的附录部分):

- FindNode: IServerAddressSpace 接口函数,根据节点 ID 在地址空间中查 找指定的节点。
- ② FindChild: IServerNode 接口函数,根据节点浏览名称查找对象的子节点。
- ③ ReadAttribute: IServerNode 接口函数,读取节点的特性(Attribute)值。
- ④ WriteAttribute: IServerNode 接口函数,修改节点的特性(Attribute)值。

⑤ Call: IServerMethodNode 接口函数,执行方法节点所定义的方法。

# 7.3 条件脚本

条件脚本用于构建当各变量满足一定条件时才会执行的脚本。

假定 RollMachine(轧机)对象类型包含 RollForce(轧制力)和 Gap(辊缝) 变量。我们需要建立一个条件脚本,当发现轧制力过大时,辊缝抬高 5mm。

单击弹出对话框(如图 7.2 所示)。

新建脚本	_ = ×
浏览名称	RollForceProtectScript
描述	轧制力包含脚本
	确定 取消
	, 

### 图 7-2 新建条件脚本

可以在源中加入脚本触发条件,触发条件可以是一个节点,或者表达式。点击"源"后面的选择按钮,弹出修改源对话框,选择表达式,并插入相对节点和 表达式,如下:

修改源	_ = ×
绝对节点 相对节点 表达式	
{{nr::/2:RollForce}}>10000	▲ 检查表达式
	插入节点属性
	•
	福宁 町空火
	1974年 秋月

图 7-3 脚本源

条件脚本中包含三个脚本: OnConditionBecomeTrue 当条件变成真时执行; WhileConditionIsTrue 当条件保持真时执行(按设定间隔执行); OnConditionBecomeFalse 当条件变假时执行。

```
源 {{nr::/2:RollForce}}>10000
                                                 ...
间隔
                                        60,000 ≑ 室秒
11
12 namespace AicTech.DataHub.Server.Generated
13 {
14
        /// <summary>
       /// ConditionalScripts class.
15
       /// </summary>
16
       public static partial class ConditionalScripts
17
18
19
            /// <summary>
20
           /// Script executed when condition become true.
21
           /// </summary>
           /// <remarks>If you don't need such conditional script,just comment it out.</remarks>
public static Opc.Ua.ServiceResult OnConditionBecomeTrue(
22
23
24
               Opc.Ua.Server.IScriptExecuteContext scriptContext)
25
           {
26
                var systemContext = scriptContext.SystemContext;
                var curNode = scriptContext.ContextNode;
27
28
29
                try
30
                -{
31
                    //get gap node
32
                    var qnGap = new QualifiedName("Gap",2);
                    var nodeGap = curNode.FindChild<IServerVariableNode>(systemContext,
33
                                                                             qnGap,
34
35
                                                                              ReferenceTypes.Aggregates,
36
                                                                              true,
37
                                                                              null,
38
                                                                              false);
39
                    double newValue = (double)nodeGap.Value + 5;
40
41
                    //write gap value
42
                    var aResult = nodeGap.WriteAttribute(systemContext,
43
                                                            Attributes.Value,
44
                                                            NumericRange.Empty,
45
                                                            new DataValue(new Variant(newValue),
46
                                                                           StatusCodes.Good,DateTime.UtcNow));
                    if(ServiceResult.IsGood(aResult))
47
48
49
                        nodeGap.ClearChangeMasks(systemContext,true);
50
                    }
51
52
                catch(Exception ex)
53
54
                {
                    Utils.Trace(ex,"RollForceProtectScript");
55
56
                    return Opc.Ua.StatusCodes.BadUnexpectedError;
57
                }
58
59
                return Opc.Ua.StatusCodes.Good:
60
           }
61
```

#### 图 7-4 条件脚本编辑界面

#### 7.4 添加数据变化脚本

比如我们要计量 Tank 类型的容积(Volume),但没有直接的计量装置。只能 靠液位来换算。为了实现这一功能,我们给 Tank 类型增加一个数据变化脚本。

右键点击 Tank 类型,选择新建数据变化脚本,

新建脚本	_ = ×
浏览名称	OnLevelChangeScript
描述	液位变化脚本
	确定取消

图 7-5 新建数据变化脚本

点击"源"后面的选择按钮,选择相对节点的 Level:

修改源		-	x	
绝对节点	相对节点表达式			
🔺 🋂 Tan	k			
- 🕨 🔶	LevelAlarm			ľ
🔧	TempChangeRateAlarm			
▶ 🔧	TemperatureDeviationAlarm		 	
	Level			
	OnLevelChangeScript			
	Temperature			
	Volume			
	IncreaseLevel			
. ► 👒	FreezingTank			
/				
1				
属性	Value		•	
节点路径	/2:Level			
		-		
	确定	取消		

# 图 7-6 数据变化脚本源

脚本内容如下:

AicDataHub 用户手册

```
源 /2:Level
                                                  ...
            /// </summary>
Ζ⊥
22
            public static Opc.Ua.ServiceResult OnDataChanged(
23
                Opc.Ua.Server.IScriptExecuteContext scriptContext,
24
                object
                                                         newValue,
25
                DateTime
                                                         newTime,
26
                Opc.Ua.StatusCode
                                                        newCode)
27
            {
28
                var systemContext = scriptContext.SystemContext;
29
                var curNode = scriptContext.ContextNode;
30
31
                try
32
                {
                     //get level value
33
                     var qnLevel = new QualifiedName("Level",2);
34
                     var nodeLevel = curNode.FindChild<IServerVariableNode>(systemContext,
35
36
                                                                                  gnLevel,
37
                                                                                  ReferenceTypes.Aggregates,
38
                                                                                  true.
                                                                                  null.
39
40
                                                                                  false);
                     var vLevel = (double)nodeLevel.Value;
41
42
43
                     //calculate volume
44
                     double vVolume = 0.0;
                     if(vLevel < 20)</pre>
45
                         vVolume = 10*vLevel;
46
47
                     else
48
                         vVolume = 10*20+ 20*(vLevel-20);
49
50
                   //set volume value
51
                   var qnVolume = new QualifiedName("Volume",2);
52
                   var nodeVolume = curNode.FindChild<IServerVariableNode>(systemContext,
53
                                                                              qnVolume,
54
                                                                             ReferenceTypes.Aggregates,
55
                                                                             true,
56
                                                                             null,
57
                                                                             false);
58
                   var aResult = nodeVolume.WriteAttribute(systemContext;
59
                                                             Attributes.Value
                                                             NumericRange.Empty,
60
                                                             new DataValue(new Variant(vVolume),
61
                                                                           StatusCodes.Good,DateTime.UtcNow));
62
63
                   if(ServiceResult.IsGood(aResult))
64
65
                       nodeVolume.ClearChangeMasks(systemContext,true);
66
                   3
67
68
               }
69
               catch(Exception ex)
70
               {
                   Utils.Trace(ex,"OnLevelChangeScript:{0}",ex.Message);
return Opc.Ua.StatusCodes.BadUnexpectedError;
71
72
73
               }
74
75
               return Opc.Ua.StatusCodes.Good;
76
           }
77
      }
78 }
79
```

#### 图 7-7 数据变化脚本编辑界面

#### 7.5 添加系统脚本

右键点击系统脚本文件夹则会弹出新建系统脚本菜单:

AicDataHub 用户手册



图 7-8 脚本选择界面

脚本类型	解释
应用程序启动脚本	AicDatahub 启动时执行
应用程序关闭脚本	AicDatahub 退出时执行
会话激活脚本	新建会话时执行, 传入 Session (session)参数
会话关闭脚本	会话关闭时执行, 传入 Session (session)参数
事件脚本	新产生事件时执行, 传入 IFilterTarget (eventData)参数
定时器脚本	按固定间隔定时执行

# 7.6 添加用户功能脚本

可以在这里定义一些共用的类和函数。可以在其它脚本中调用。

# 7.7 引用程序集(ReferenceAssemblies)

引用程序集可以扩展 AicDatahub 的脚本功能,通过引用程序集,可以使用 第三方,或者独立开发的.NET 动态库。

右键点击"引用程序集",选择编辑(Edit),则弹出引用集合编辑窗口:

31		Sar 11 47 56 S		- 0	J X
	程序集名称 で	程序集版本	程序集位置		τ
>	WMS.Scheduling	1.0.0.0	C:\WCS\BIN\WMS.Scheduling.dll		
			添加	取消	
					_

# 图 7-9 引用程序集

可以添加、删除程序集。

# 8 属性(Property)

相对于变量,属性一般保存配置信息。属性可以是系统预置的属性,也可以 是用户自定义的属性。本章将具体介绍系统预定义的属性。

在新建属性时,点击"浏览名称(BrowseName)"后面的选择按钮…,,会弹出系统预定义的属性。

新建属性	_ — ×
浏览名称	
描述	
数据类型	String
维度	Scalar 🔹
类型定义	PropertyType
建模规则	MandatoryShared 🔹
	确定 取消

图 8-1 新建属性

系统预定义的类属性包括:

- 节点版本 (NodeVersion)
- 图标 (Icon)

系统预定义的对象属性包括:

- 节点版本 (NodeVersion)
- 图标 (Icon)
- 命名规则(NamingRule)

系统预定义的变量属性包括:

- 定义 (Definition)
- 值精度(ValuePrecision)
- 节点版本 (NodeVersion)
- 本地时间(LocalTime)
- 允许为空(AllowNulls)
- 工程单位(EngineeringUnits)
- 工程单位量程(EURange)
- 仪表量程(InstrumentRange)
- 历史记录设置(HistorizingConfiguration)

本章只介绍以上需要特殊设置的属性。

#### 8.1 历史记录设置

当选择历史记录设置属性后,将弹出历史记录设置界面。

- 保存历史数据配置			
步进			
最大超时	0	*	室秒
最小超时	0	*	室秒
异常偏差格式	绝对值	•	
异常偏差	1.00	*	
- 统计配置			_
使用服务器默认值			
将不确定的数据视为坏数据			
使用外推插值			
坏数据百分比	100	*	
好数据百分比	100	*	

图 8-2 历史记录设置

**步进:**选择步进的变量是那些不连续跳变的变量(阶梯状);否则,则是连续变化的变量。

**最大超时**:上次记录之后的最大超时时间内,即使没有产生足够的偏差,也 会触发记录一次数据。0表示忽略此设置。

**最小超时**:上次记录之后的最小超时时间内,即使产生了新的偏差,也不会 触发记录。而是等到最小超时之后。0表示忽略此设置。

**异常偏差格式:**绝对值或相对值(工程单位范围的百分比)。

**异常偏差:**产生异常偏差的阈值。

统计配置是为了计算变量区间历史数据的统计值时使用,系统支持 OPC UA 标准定义的所有统计函数,如 Interpolative、Average、TimeAverage、Total、Count、

StandardDeviationPopulation等。

将不确定的数据视为坏数据(TreatUncertainAsBad):指示服务器在进行统计计算时,如何对待质量戳为"不确定(Uncertain)"的数据。当该值为 False时,服务器将质量戳为 Uncertain 的数据当成好(Good)数据。当该值为 True时,服务器将质量戳为 Uncertain 的数据当成坏(Bad)数据。该值默认为 True。

使用外推插值(UseSlopedExtrapolation):指示服务器在缺少右边界值的情况下,如何对数据进行插值。当该值为False的时候,服务器将不使用外推插值, 而是使用最后一个已知的值作为插值结果。当该值为True的时候,服务器将使 用外推插值,即使用过去的值外推/预测未来的值。该值默认为False。

**坏数据百分比(PercentDataBad)**:指示服务器在进行统计计算时,坏数据 所占的百分比必须达到多少时,统计计算结果的质量戳才能设为 Bad。该值默认 为 100。

**好数据百分比(PercentDataGood)**:指示服务器在进行统计计算时,好数 据所占的百分比必须达到多少时,统计计算结果的质量戳才能设为 Good。该值 默认为 100。

**注意:**这里需要注意的是,上述的两个百分比参数必须满足如下不等式,否则服务器将返回 Bad\_AggregateInvalidInputs 错误码指明输入错误(如果是等式成立,则优先使用 PercentDataGood 条件):

PercentDataBad+ PercentDataGood >= 100

#### 8.2 工程单位量程

当选择值工程单位量程(EURange)属性后,将弹出量程设置界面。

◇ 父对象信息	
父节点	2:Level
引用类型	HasProperty
▲ 节点属性	
节点分类	Variable
节点ID	ns=3;i=1197
浏览名称	EURange
显示名称	▼ EURange
描述	▼
可写标志	1577587
数据类型	Range
维度	Scalar 🔹
值	{0   200}
访问级别	3
▲ 实例声明	
类型定义	PropertyType
建模规则	MandatoryShared 🔻

图 8-3 工程单位量程

量程设置值类型是 Range,包括高值和低值,点击值后面的设置按钮,弹出 值设置界面:

修	3改值		- 1	⊐ X
	Low		0.0	D 🌲
•	High		200.00	<b>)</b>
		 确定	取消	

图 8-4 量程值

## 8.3 仪表量程

当选择仪表量程(InstrumentRange)属性后,将弹出量程设置界面。

◇ 父对象信息		
父节点	2:Level	
引用类型	HasProperty	
◇ 节点属性		
节点分类	Variable	
节点ID	ns=3;i=1201	
浏览名称	InstrumentRange	
显示名称	▼ InstrumentRange	
描述	•	
可写标志	1577587	
数据类型	Range	
维度	Scalar 🔻	
值	{0   20}	
访问级别	3	
◆ 实例声明		
类型定义	PropertyType	
建模规则	MandatoryShared 🔹	

图 8-5 仪表量程

量程设置值类型是 Range,包括高值和低值,点击值后面的设置按钮,弹出 值设置界面:

修改值	- 5	ı x
Low	0.00	-
High	20.00	•
	确定取消	

图 8-6 量程值

## 8.4 工程单位

当选择工程单位(EngineeringUnits)属性后,将弹出工程单位设置界面。

◇ 父对象信息		
父节点	2:Level	
引用类型	HasProperty	
▲ 节点属性		
节点分类	Variable	
节点ID	ns=3;i=1199	
浏览名称	EngineeringUnits	
显示名称	EngineeringUnits	]
描述	•	]
可写标志	1577587	.]
数据类型	EUInformation	
维度	Scalar 🔹	
值	(null)	
访问级别	3	.]
▲ 实例声明		
类型定义	PropertyType	
建模规则	MandatoryShared 🔹	

图 8-7 工程单位

工程单位数据类型是 EUInformation,点击后面的选择按钮,弹出工程单位 设置对话框:

修改值	_ 🗆 X
NamespaceUri	http://www.opcfoundation.org/UA/units/un/cefact
UnitId	0
DisplayName	• /m
Description	▼ degree per metre
	确定取消

图 8-8 修改工程单位值

# 9 安全管理(Security)

双击主界面上的安全文件夹(Security)或者左击主界面上安全文件夹 (Security)左侧的小三角,则展开安全文件夹的子菜单,分别为用户管理(User Management)和角色管理(Role Management)(如图 9.1 所示)。

## 9.1 用户管理

双击用户管理图标或者右击后选择编辑(Edit)则会弹出用户管理窗口(如 图 9.2 所示)。

用户管理窗口中又包含三个子窗口,从左到右依次为用户列表(User List)、 角色列表(Role List)和用户信息(User Information)。

其中用户列表(UserList)列出了现有的所有用户名。其右半侧有相应的新 建用户(Create User)、删除用户(Delete User)、修改密码(Change Password) 和重置密码(Reset Password)的功能按钮,单击按照提示即可完成对所选用户 的相应的操作。

角色列表列出了当前所选用户所拥有的角色。其右半侧有相应的增加用户角色(Add User 's Role)和删除用户角色(Delete User's Role)的功能按钮。单击按照提示即可完成对所选用户的增加角色或删除角色的操作。



图 9-1 安全文件夹

用户列表		角色列表		一用户信息 ————————————————————————————————————	
sa	创建用户	SystemAdministrator	添加角色	用户名	sa
operator	删除用户		删除角色	电子邮件	sa@www.aic-tech.com
viewer	修改密码			3Coll	
@anonymous@	重置密码			UTAH	
	解锁用户			锁定标志	

图 9-2 用户管理

用户信息窗口则列出了用户名(User Name)、用户邮箱(Email)、备注信息(Comment)和用户是否被锁定(IsLockedOut)这些信息。

#### 9.1.1 创建新用户

单击用户列表(User List)一栏中右半部分的新建用户(Create User)按钮,则弹出新建用户对话框(如图 9.3 所示)。正确填入新用户的信息后点击"确定"按钮即可创建一个新用户。创建完成后新创建的用户将会显示在左侧的用户列表中。

新建用户	-		х
用户名			
密码			
确认密码			
电子邮件			
说明			
	确定 取消	Í	

图 9-3 新建用户

#### 9.1.2 删除用户

在用户列表(User List)一栏中左半部分的用户列表选中想要删除的用户, 然后点击"删除用户"(Delete User)按钮即可删除选中的用户。注意:系统预 定义用户不能删除。

#### 9.1.3 修改密码

选择想要修改密码的用户,然后点击"修改密码"按钮(Change Password), 弹出对话框(如图 9.4 所示),正确填写旧密码以及新密码后点击"确定"按钮 即可完成密码的修改。

修改密码			-		х
用户旧密码					
用户新密码					
确认密码					
		确定	取消	Í	

图 9-4 修改密码

#### 9.1.4 重置密码

选择想要重置密码的用户,然后点击"重置密码"按钮(Reset Password), 系统就会自动为选中的用户生成一个新的密码,并以对话窗口的方式显示给用 户。如图 9.5 所示:

DataHub	<b>33</b> 五月	X
1	重置用户'NewUser'密码成功,新密码为: w-A{>OnO。	
	确定	

图 9-5 重置密码

#### 9.1.5 解锁用户

若想解锁被锁定的用户,首先选中该用户,然后点击"解锁用户"按钮(Unlock User),即可完成对该用户的解锁。

#### 9.1.6 添加用户角色

选中用户后,点击"添加用户角色"按钮(Add User's Role),弹出对话框(如图 9.6 所示),选择想要添加的角色然后点击"确定"按钮,即可为该用户添加一个新的角色。

选择角色	х
SystemAdministrator	
ModelAdministrator	
Reader	
Writer	
HistoryReader	
HistoryWriter	
MethodExecutor	
ViewBrowser	
确定 取消	

图 9-6 为用户添加角色

目前系统实现的权限控制仅依赖于系统预定义角色,不同的角色具有不同的 操作权限,当用户属于某个预定义角色,就具有这个角色所代表的操作权限。系 统预定义的角色如下:

◆SystemAdministrator: 系统管理员,可以执行任何操作。

- ◆ModelAdministrator:模型管理员,可以修改地址空间(如创建/删除节点) 及读写数据等所有常规操作,唯独不能执行跟安全管理相关的操作(如 创建新用户)。
- ◆Reader: 可以读取节点属性,如读取/订阅变量实时数据。
- ◇Writer: 可以修改节点属性, 如更新变量实时数据。
- ◆HistoryReader: 可以读取变量历史数据以及报警/事件历史记录。
- ◆HistoryWriter: 可以修改变量历史数据以及报警/事件历史记录。
- ◆ MethodExecutor: 可以调用除安全相关之外的其他方法。
- ◆ViewBrowser:可以浏览地址空间。

#### 9.1.7 删除用户角色

选中"用户所属角色列表"中的角色后,点击"删除用户角色"按钮,即可 删除该用户的这一用户角色。注意:系统预定义角色不能删除。

#### 9.2 角色管理

双击角色管理(Role Management)图标或者右击后选择编辑(Edit)则会弹 出角色管理窗口(如图 9.7 所示)。

角色管理(Role Management)窗口中又包含两个子窗口,从左到右依次为 角色列表(Role List)、用户列表(User List)。

角色列表(Role List)列出了现有的所有角色成员,其右半侧有新建角色(Create Role)和删除角色(Delete Role)两个功能按钮,用于创建新角色和删除选中的角色。

用户列表(User List)列出了所有拥有当前所选角色的用户,其右半侧有添加角色用户(Add Role's User)和删除角色用户(Delete Role's User)功能按钮。用于为当前所选中的角色添加或删除用户。



图 9-7 角色管理

#### 9.2.1 创建新角色

点击"创建新角色"按钮,弹出对话框(如图 9.8 所示),填入新角色的名称后点击"确定"按钮(OK)即可创建一个新角色。创建完成后新创建的角色将 会显示在左侧的角色户列表中。

创建角色			х
?	请输入角色名:		
		确定	取消

#### 图 9-8 创建新角色

**注意**: 创建自定义角色是为未来的变量级权限控制所保留的功能,目前系统 实现的权限控制仅依赖于系统预定义角色,因此目前并无创建自定义角色的必 要。

#### 9.2.2 删除角色

在左侧的列表中选择想要删除的角色,然后点击"删除角色"按钮(Delete Role) 即可删除选中的角色。

## 9.2.3 为某个角色添加用户

选中特定的角色后,点击"添加用户"按钮(Add Role's User),弹出对话框(如图 9.9 所示),对话框中显示当前系统中所有不具有该角色的用户,选择想要添加的用户后点击"确定"(OK)即可将此用户添加到这一角色下。

选择用户		Х
sa		
viewer		
@anonymous@		
NewUser		
	确定	取消

#### 图 9-9 为某个角色添加用户

#### 9.2.4 为某个角色删除用户

在"角色所含用户列表"中选中想要删除的用户,然后点击"删除角色用户" (Delete Role's User)按钮后即可将该用户从该角色下删除。

# 10 附录一 常用接口函数

本附录列出了在脚本编写过程中最常用的一些接口函数。

## 10.1 地址空间访问接口(IServerAddressSpace)

### 10.1.1 FindNode

功能	根据节点 ID(NodeId)查找指定的节点。					
原型	IServerNode Fi	ndNode(N	ode	ld nodeld, bool initia	lizeChildren = true)	
	T FindNode <t>(NodeId nodeId,bool initializeChildren = true) where T : class,IServerNode</t>					
参数	参数名称	参数类型	٩ ١	参数说明		
	nodeId					
	initializeChild	子对象。该参数为 false 的情况				
	ren		仅用于系统初始化时,因而在脚本中调用时,此 数应始终为 true (或者不填即默认为 true)。			
返回值	返回值为 IServ	erNode 或	其派	《生接口。IServerNod	le 是所有类型节点的基接口,根	
	据节点分类不同,其实现了不同的派生接口。					
	节点分类 接口类型					
	ObjectType IS			[ServerObjectTypeNode		
	DataType		ISe	erverDataTypeNode		
	ReferenceType		ISe	erverReferenceTypeNo	ode	
	VariableType		ISe	erverVariableTypeNod	e	
	Object		ISe	erverObjectNode		
	Variable		ISe	erverVariableNode	IServerPropertyNode	
					IServerDataVariableNode	
	Method		ISe	erverMethodNode		
	View		ISe	erverViewNode		
示例	var nodeldElev var nodeElevat	ator = nev or = script n	v No Con ode	odeld(31365,3); text.AddressSpace.Fi IdElevator);	ndNode <iserverobjectnode>(</iserverobjectnode>	

功能	获取节点通过引用	月关系连接的模板节	与点信息。		
原型	IList <ireference> GetReferences(</ireference>				
	Exp	andedNodeId sour	celd,		
	Noc	leId referenceType	,		
	boo	l includeSubTypes,			
	Bro	wseDirection brow	seDirection)		
参数	参数名称	参数类型	参数说明		
	sourceId	sourceId ExpandedNodeId 源节点 ID			
	referenceType	nceType NodeId 引用关系			
	includeSubTypes	bool 是否包含 referenceType 参数指定的引用			
			系的派生引用关系所建立的引用		
	browseDirection	BrowseDirection	指明引用关系的方向,是正向 (Forward)、		
			反向(Inverse)还是都包括(Both)		
返回值	符合条件的引用节	节点信息列表。			
示例	var references = scriptContext.AddressSpace.GetReferences(				
		new Nodeld(3136	5,3),		
		Opc.Ua.Reference	TypeIds.HasComponent,		
		true,			
	BrowseDirection.Forward);				

# 10.1.2 GetReferences

# **10.2** 节点访问接口(IServerNode)

## 10.2.1 FindChild

功能	查找节点的子节点。				
原型	T FindChild <t>(</t>				
// <b>··</b> ·	ServerSystemCo	ntext context,			
	QualifiedName k	prowseName,			
	Nodeld referenceTypeld,				
	bool includeSubTypes,				
	Nodeld typeDefinitionId,				
	bool isMandatoryChild)				
	where T : class,IServerNode				
参数	参数名称	参数类型	参数说明		

	context	ServerSystemContext	系统上下文参数		
	browseName	QualifiedName	节点的浏览名称		
	referenceTypeId	NodeId	与子节点之间的引用关系		
	includeSubTypes	bool	是否包含 referenceTypeId 参数指		
			定的引用关系的派生引用关系所		
			连接的子节点		
	typeDefinitionId	NodeId	节点的类型定义		
	isMandatoryChild	bool	指明子节点是否必须存在。如果		
			该参数为 true,并且没有找到符		
			合条件的子节点,此函数将抛出		
			异常。如果该参数为 false,并且		
			没有找到符合条件的子节点,则		
			此函数将返回 null。		
返回值	符合条件的子节点或	null(没有找到符合条件	的子节点并且 isMandatoryChild		
	参数为 false 时)。				
示例	var nodeLevel= curN	ode.FindChild <iservervaria< th=""><th>bleNode&gt;(</th></iservervaria<>	bleNode>(		
	scriptContext.SystemContext,				
	new QualifiedName("Level",2),				
	Opc.Ua	a.Reference Typelds. Has Con	nponent,		
	true,				
	null,				
	false);				

## **10.2.2 FindProperty**

功能	查找节点的属性节	方点。			
原型	IServerPropertyNode FindProperty( ServerSystemContext context, QualifiedName propertyName, bool isMandatory)				
参数	参数名称   参数类型    参数说明				
	context	ServerSystemContext	系统上下文		

	propertyName	QualifiedName	属性的浏览名称	
	isMandatory	bool	指明属性节点是否必须存在。	
返回值	符合条件的属性节点或 null。			
示例	<pre>var nodeEURange = curNode.FindProperty(     scriptContext.SystemContext,     new QualifiedName("EURange"),     false):</pre>			

#### 10.2.3 FindMethod

功能	查找定义在节点上的方法。					
原型	IServerMethodNode FindMethod(					
~~··	ServerSystem	Context context,				
	QualifiedName	e methodName,				
	bool isMandat	ory)				
参数	参数名称 参数类型 参数说明					
	context ServerSystemContext 系统上下文					
	methodName	methodName QualifiedName 方法节点的浏览名称				
	isMandatory	isMandatory bool 指明方法节点是否必须存在。				
返回值	符合条件的方法节	点或 null。				
示例	var nodeMethod = curNode.FindMethod(					
• • •	scriptContext.SystemContext,					
	new Qualified	Name("Start",2),				
	false);					

### 10.2.4 ReadAttribute

功能	读取变量的特性(A	Attribute)值。			
原型	ServiceResult Read	Attribute(			
	ServerSystemC	Context context,			
	uint attributelo	ł,			
	NumericRange indexRange,				
	QualifiedName dataEncoding,				
	DataValue value)				
参数	参数名称	参数类型	参数说明		
	context	ServerSystemContext	系统上下文		

AicDataHub 用户手册

attributeId	uint	特性 ID, 由 OPC U	A 标准所定义
		的 22 种特性之一。	
		NodeId	1
		NodeClass	2
		BrowseName	3
		DisplayName	4
		Description	5
		WriteMask	6
		UserWriteMask	7
		IsAbstract	8(仅用于类
			型节点)
		Symmetric	9(仅用于引
			用 类 型 节
			点)
		InverseName	10 (仅用于
			引用类型节
			点)
		ContainsNoLoops	11 (仅用于
			视图节点)
		EventNotifier	12 (仅用于
			对象和视图
			节点)
		Value	13 (仅用于
			变量和变量
			类型节点)
		DataType	14 (仅用于
			变量和变量
			类型节点)

		ValueRank	15 (仅用于
			变量和变量
			类型节点)
		ArrayDimensions	16 (仅用于
			变量和变量
			类型节点)
		AccessLevel	17 (仅用于
			变量节点)
		UserAccessLevel	18 (仅用于
			变量节点)
		MinimumSampling	19(仅用于
		Interval	变量节点)
		Historizing	20 (仅用于
			变量节点)
		Executable	21 (仅用于
			方法节点)
		UserExecutable	22(仅用于
			方法节点)
indexRange	NumericRange	用于读取数组类型	型的值时使用
		(仅适用于 Value?	持性),可以只
		读取数组的一部分	,此参数指明
		数组的索引范围。	不是读取数组
		类型的 Value 属性	时,此参数置
		为 NumericRange.E	mpty 即可。
dataEncoding	QualifiedName	数据编码方式,此参	参数置为 null
		即可。	
value	DataValue	用于保存返回的特	性值。此参数
		必须是预先分派好	的,不能传入

## AicDataHub 用户手册

			null 值, 否则函数会返回错误。
返回值	错误码,指明读取成功或失败。		
示例	<pre>var aValue = new D var aResult = curNo scriptContext.S Opc.Ua.Attribu NumericRange null, aValue); if(ServiceResult.IsG {     // 读取成功的 } else {     // 读取失败的 }</pre>	ataValue(); de.ReadAttribute( SystemContext, ites.Value, .Empty, ood(aResult)) 句处理逻辑	

# 10.2.5 WriteAttribute

功能	修改节点的特性值。		
原型	ServiceResult WriteAttribute( ServerSystemContext context, uint attributeId, NumericRange indexRange, DataValue value)		
参数	参数名称	参数类型	参数说明
	context	ServerSystemContext	系统上下文
	attributeId	uint	特性 ID, 由 OPC UA 标准所定义的
			22 种特性之一。
	indexRange	NumericRange	用于修改数组类型的值时使用(仅
			适用于 Value 特性),可以只修改数
			组的一部分,此参数指明数组的索
			引范围。不是修改数组类型的 Value
			属性时, 置为 NumericRange.Empty

			即可。	
	value Data	DataValue	需要修改的特性值,此值不能为	
			null.	
			attributeId	注意事项
			Value(13)	value.ServerTimesta mp 必须为 DateTime.MinValue (即没有值)
			非 Value	<ul> <li>◇ value.ServerTi mestamp 必须 为</li> <li>DateTime.Min</li> <li>Value</li> <li>◇ value.SourceTi mestamp 必须 为</li> <li>DateTime.Min</li> <li>Value</li> <li>◇ value.StatusCo de 必须为</li> <li>Good</li> <li>◇ indexRange 必 须为</li> <li>NumericRange.</li> </ul>
返回值	错误码,指明写)			Empty
示例	<pre>var aResult = curN scriptContext Opc.Ua.Attril NumericRang new DataVall if(ServiceResult.Is { // 写入成功 } else { // 写入失败</pre>	lode.WriteAttribute( t.SystemContext, outes.Value, ge.Empty, ue(new Variant(11.0f),Statu Good(aResult)) 的处理逻辑 的处理逻辑	usCodes.Good,D	ateTime.UtcNow));

## 10.2.6 ClearChangeMasks

功能	清除节点的修改标志并发布更新通知。此函数将触发向订阅了该节点数据变化通		
	知的客户端发送数据变化消息的动作,从而客户端的数据也将自动得到更新。因		
	此在脚本中如果修改了节点特性,必须调用此函数以客户端能够得到通知。		
原型	void ClearChangeMasks( ServerSystemContext context, bool includeChildren)		
参数	参数名称	参数类型	参数说明
	context	ServerSystemContext	系统上下文
	includeChildren	bool	是否递归地在节点的子节点上也
			调用此函数。
返回值	无		
示例	<pre>无 此函数通常在修改节点的一个或多个特性之后调用。 var aResult = curNode.WriteAttribute(     scriptContext.SystemContext,     Opc.Ua.Attributes.Value,     NumericRange.Empty,     new DataValue(new Variant(11.0f),StatusCodes.Good,DateTime.UtcNow)); if(ServiceResult.IsGood(aResult)) {     // 写入成功的处理逻辑     curNode.ClearChangeMasks(scriptContext.SystemContext,false); } else {     // 写入失败的处理逻辑 </pre>		

# **10.3** 方法节点接口(IServerMethodNode)

#### 10.3.1 Call

功能 执行方法。
----------

原型	ServiceResult Call( ServerSystemCon Nodeld IList <variant> IList<serviceresu IList<variant></variant></serviceresu </variant>	text context, objectId, inputArguments, lt> argumentErrors, outputArguments)	
参数	参数名称	参数类型	参数说明
	context	ServerSystemContext	系统上下文
	objectId	NodeId	目标对象 ID,即在哪个对象上
			执行该方法(方法可以定义在
			对象类型上,因此调用时必须
			指明目标对象)
	inputArguments	IList <variant></variant>	输入参数列表
	argumentErrors	IList <serviceresult></serviceresult>	用于保存输入参数错误
	outputArguments	IList <variant></variant>	用于保存输出参数
返回值	错误码,指明方法调用	用是否存在异常(非参数错	持误异常)。需要注意的是,此函
	数不能仅凭错误码确定方法调用是否完全成功,还必须检查 argumentErrors 中是		
	否返回了输入参数错误。		
示例	<pre>var nodeMethod = curNode.FindMethod(     scriptContext.SystemContext,     new QualifiedName("Start",2),     false);</pre>		
	var inputArguments =	new VariantCollection();	
	var argumentErrors = i var outputArguments	<pre>new List<serviceresult>(); = new VariantCollection();</serviceresult></pre>	
	var aResult = nodeMet	hod.Call(	
	scriptContext.Syst	temContext,	
	inputArguments,		
	argumentErrors,		
	outputArguments	;); aResult))	
	{		
	// 调用失败的处	理	
	throw new Service	eResultException(aResult);	



# **10.4** 变量节点接口(IServerVariableNode)

### 10.4.1 GetValue

功能	获取变量的当前值并转换为指定的类型。		
原型	T GetValue <t>(ServerSystemContext context, bool throwOnError)</t>		
参数	参数名称	参数类型	参数说明
	context	ServerSystemContext	系统上下文
	throwOnError	bool	指明类型转换失败时,是否抛出异
			常(此参数为 true 时)还是返回
			默认值(此参数为 false 时)
返回值	变量当前值或默认值 (发生类型转换错误时)		
示例	<pre>变量当前值或默认值(发生类型转换错误时) var nodeEURange = curNode.FindProperty(     scriptContext.SystemContext,     new QualifiedName("EURange"),     false); if(nodeEURange != null) {     var valueRange = nodeEURange.GetValue<opc.ua.range>(         scriptContext.SystemContext,         false);     if(valueRange != null)     { </opc.ua.range></pre>		

}	
}	

# **10.5** 事件访问接口(IFilterTarget)

# 10.5.1 IsTypeOf

功能	检查事件的类型是否为某个特定的类型。		
原型	bool IsTypeOf( FilterContext context, NodeId typeDefinitionId)		
参数	参数名称	参数类型	参数说明
	context	FilterContext	系统上下文
	typeDefinitionId	NodeId	事件类型 ID
返回值	如果事件类型是指定的	的类型,则返回 true	e, 否则返回 false。
示例	public static Opc.Ua.Service Opc.Ua.Server.IScriptI Opc.Ua.IFilterTarget { var filterContext = new if(eventData.IsTypeOf { // 单一液位报 } else if(eventData.IsType { // 单一偏差报 } else if(eventData.IsTypeOf(filter { // 单一变化率 } return Opc.Ua.StatusO }	Result OnEvent( ExecuteContext scriptCo even w ServerFilterContext(so (filterContext,Opc.Ua.C) 擎 peOf(filterContext,Opc. 擎 Context,Opc.Ua.Object 根警	ontext, tData) criptContext.SystemContext); ObjectTypeIds.ExclusiveLeveIAlarmType)) Ua.ObjectTypeIds.ExclusiveDeviationAlarmType))

功能	获取事件记录的某个字段的值。		
原型	object GetAttributeValue(FilterContextcontext,NodeldtypeDefinitionId,IList <qualifiedname>relativePath,uintattributeId,NumericRangeindexRange)</qualifiedname>		
参数	参数名称	参数类型	参数说明
	context	FilterContext	系统上下文
	typeDefinitionId	NodeId	事件类型 ID
	relativePath	IList <qualifiedname></qualifiedname>	字段相对于事件类型的访问
			路径
	attributeId	uint	特性 ID
	indexRange	NumericRange	用于访问数组类型的字段
			时,可指定数组的索引范围,
			只读取数组的一部分内容。
			对于非数组类型字段,置为
			NumericRange.Empty即可。
返回值	字段的值或 null。		
示例	public static Opc.Ua.ServiceResult OnEvent( Opc.Ua.Server.IScriptExecuteContext scriptContext, Opc.Ua.IFilterTarget eventData) { var filterContext = new ServerFilterContext(scriptContext.SystemContext); if(eventData.IsTypeOf(filterContext,Opc.Ua.ObjectTypeIds.ExclusiveLevelAlarmType)) { // 单一液位报警,获取 Message 字段的值 var val = eventData.GetAttributeValue( filterContext, Opc.Ua.ObjectTypeIds.ExclusiveLevelAlarmType, new QualifiedName[] { new QualifiedName("Message")}, Attributes.Value, NumericRange.Empty);		

## 10.5.2 GetAttributeValue

Opc.Ua.Utils.TraceError("ExclusiveLevelAlarmType:{0}",val);
}
return Opc.Ua.StatusCodes.Good;
}