通道驱动开发

# 通道驱动构成

一种通道驱动由两个文件构成：dll代码文件和XML描述文件，dll中包含了驱动通道工作的用于IO开发和IO运行的代码，XML文件描述了使用该通道驱动的信息。

XML文件结构：

<ChannelDriver Name="" Description="" Assembly="" DevClass="" RunClass="">

</ChannelDriver>

# 命名

## 代码命名规则

命名空间：ControlEase.IoDrive

开发类命名规则：XXXChannelDev（XXX代表设备型号）

运行类命名规则：XXXChannel

程序集命名规则：XXXChannel.dll

## Xml命名规则

文件名称规则：. XXXChannel.cfg

# 功能实现

具体通道驱动实现主要根据通信端口类型和IO架构规范，完成用于IO系统中的运行的功能和开发的功能。

## 2.1 运行功能实现

主要是实现是准备通道、使用通道发送数据、从通道接收数据、检测通道的状态以及使用完毕后释放通道资源等功能。

实现这些功能时，需要从通道驱动运行基类Channel派生，实现基类中规定的各种功能。

### 描述通道的基本特性

* **通道的可配置参数（ConfigData）**

一种通道驱动可驱动多个同类的通信口为IO系统工作，不同的物理通信端口其配置不同，故具体通道驱动要指定有哪些参数需要配置。

* **通道中可供接收的的数据字节数(AvailableDataCount)**

通道驱动在运行中，要提供通道中可供接收的字节数，以便IO架构及时处理这些数据。

### 2.1.2 实现功能

##### 准备通道

准备工作依通道介质或通信方式的不同而不同，一般情况下的准备工作就是初始化通道所需要的参数，并使能该Channel。如串口通信时，串口准备工作就是初始化串口的参数，并打开所在的COM口即可。实现的方法是：

Protected virtual ExecutionResult Open ( )；

##### 发送数据

完成使用通道发送数据的功能，实现的方法是：

public virtual ExecutionResult Send ( byte[] data )；

##### 接收数据

完成从通道中接收指定字节长度的数据的功能，实现的方法是：

public virtual int Receive ( byte[] buffer, int offset, int size )

##### 通道状态

具体通道驱动实现中，要在通道发生故障时，使用基类提供的功能通知IO处理故障，调用的方法是：

Protected void OnFailured ( EventArgs e )；

##### 释放通道资源

完成关闭或释放通道的逻辑，实现的方法是：

protected virtual ExecutionResult Close ( )；

## 2.2 开发功能实现

具体通道驱动实现开发类时从ChannelDev类派生，主要提供如下功能：

需要用户配置的内容ConfigData，以及编辑配置内容的WPF控件。

# 术语

|  |  |  |
| --- | --- | --- |
| **序号** | **名词** | **名词定义** |
| 1 | 通道  （Channel） | 代表一种物理媒介，如串口接口及电缆，即IO系统和Device进行数据通信时所使用的通信媒介。 |
| 2 | 通道驱动  （Channel Driver） | IO系统中的一种软件程序，对Channel的抽象，实现对通信媒介的操控及通信数据的收/发功能，它从物理端口上按照所连接的通信网络类型接收和发送数据，并且获得物理通道的状态。  不同的通道驱动与通信端口类型相关，如通信端口类型为串口、以太网口、USB口等。 |
| 3 |  |  |