

# 目 录

<b>第 1 章 Apac 开发环境介绍</b>	<b>6</b>
1.1 程序集	6
1.2 APP.CONFIG 连接配置	7
1.3 数据契约（实体）	8
1.4 数据契约（枚举）	13
<b>第 2 章 接口调用方法</b>	<b>15</b>
2.1 构造相关方法	15
2.2 点标签数据相关方法	16
2.2.1 AddNewPoint	16
2.2.2 Contains	18
2.2.3 EditPoint	18
2.2.4 GetPointInfo	19
2.2.5 DeletePoint	19
2.3 实时数据访问相关方法	20
2.3.1 SetRealData	20
2.3.2 GetPointValue	23
2.3.3 FindPointState	24
2.3.4 FindPointName	26
2.3.5 GetSystemTime	27
2.3.6 GetSystemUtcTime	27
2.4 历史数据访问相关方法	28
2.4.1 GetHisBufferData	28
2.4.2 GetHisData	29
2.4.3 GetHisData(重写)	30
2.5 计算组件相关方法	31
2.6 告警组件相关方法	32
<b>第 3 章 高级接口</b>	<b>32</b>

3.1DDM 服务(Apace 分布式管理中心).....	32
3.1.1 Login .....	32
3.1.2 UnLogin .....	34
3.1.3 ServiceAddress .....	35
3.1.4 UpdateDDMConfig .....	36
3.1.5 GetDDMConfig .....	37
3.1.6 GetAllConnectinfo .....	38
3.1.7 GetAllForbidConnectinfo .....	39
3.1.8 GetServiceAllDir .....	41
3.1.9 ServiceState .....	42
3.1.10 ServiceKeyState .....	43
3.1.11 StopService .....	45
3.1.12 RunService .....	46
3.1.13 KillService .....	47
3.1.14 GetLogInfo .....	48
3.1.15 ClearLog .....	50
3.1.16 ReBootDDM .....	51
3.1.17 GetLocalHostKey .....	52
3.1.18 RegisterLocalHostlicence .....	53
3.1.19 GetOverdueTime .....	54
3.1.20 GetIsRegister .....	55
3.1.21 GetServiceVersion .....	56
3.1.22 GetAuthorizationMaxPointCount .....	57
3.2 PCM 服务(Apace 点标签信息管理).....	59
3.2.1 AddPointInfo .....	59
3.2.2 AddPointInfoStream .....	59
3.2.3 DeletePointInfo .....	59
3.2.4 DeletePointInfos .....	60
3.2.5 UpdatePointInfo .....	60
3.2.6 GetAllPointCollocate .....	60
3.2.7 GetPointInfo .....	61
3.2.8 GetAllPointStream .....	61
3.3 实时服务(Apace 高速缓存区管理).....	62
3.3.1 GetAllPoint .....	62
3.3.2 GetAllPointStream .....	63
3.3.3 GetPoint .....	64
3.3.4 GetPointsStream .....	65
3.3.5 GetBufferData .....	67
3.3.6 GetBufferDataStream .....	68
3.3.7 SelectPointName .....	70
3.3.8 SelectPointNameStream .....	71
3.3.9 SelectPointDescribe .....	73

3.3.10	GetHisServicePath.....	74
3.3.11	Contains.....	75
3.3.12	SelectPointDescribeStream .....	77
3.3.13	SelectPointState.....	78
3.3.14	SelectPointStateStream .....	80
3.3.15	SelectPointNameRegex.....	81
3.3.16	SelectPointNameRegexStream.....	83
3.3.17	SelectOperationType .....	84
3.3.18	SelectOperationTypeStream.....	85
3.3.19	SelectOperationCode.....	87
3.3.20	SelectOperationCodeStream .....	88
3.3.21	SelectOperationCodeAndType .....	90
3.3.22	SelectOperationCodeAndTypeStream .....	91
3.3.23	SelectPointDescribeRegex .....	93
3.3.24	SelectPointDescribeRegexStream.....	94
3.3.25	AddNewPoint .....	96
3.3.26	AddNewPointStream .....	98
3.3.27	DeletePoint .....	98
3.3.28	GetPointInfo.....	99
3.3.29	GetAllPointInfoStream .....	101
3.3.30	GetAllPointInfo .....	102
3.3.31	EditPoint .....	103
3.3.32	NewRealData.....	105
3.3.33	NewRealDatas .....	106
3.3.34	NewRealDataValue.....	107
3.3.35	NewRealDataObjcet .....	108
3.3.36	GetSystemTime.....	110
3.3.37	GetSystemUtcTime.....	111
3.3.38	GetConvertTime.....	112
3.3.39	GetPointsGuid .....	114
3.3.40	GetPointsGuidStream .....	115
3.3.41	RemovePointGuid .....	117
3.3.42	GetSpacePointStream.....	118
3.3.43	GetPointInfos .....	119
3.3.44	GetPointInfosStream .....	121
3.3.45	GetSpacePointInfoStream .....	122
3.3.46	GetPointCount .....	124
3.3.47	Math_Point_Sum .....	125
3.3.48	Math_Point_Avg .....	126
3.3.49	Math_Point_Wtd .....	128
3.4	历史服务(Apace 时序历史数据管理).....	130
3.4.1	AddNewRows.....	130
3.4.2	AddNewRowStream .....	131
3.4.3	GetHisData.....	132

3.4.4 GetBufferData .....	133
3.4.5 GetStepHisData .....	135
3.4.6 GetStepHisDataProxy .....	136
3.4.7 GetNextHisData .....	138
3.4.8 GetPreHisData .....	138
3.4.9 UpdateHisData .....	139
3.4.10 GetHisPointMoment .....	139
3.4.11 Math_Avg .....	139
3.4.12 Math_Min .....	141
3.4.13 Math_Max .....	143
3.4.14 Math_MaxMinAvg .....	144
3.4.15 Math_Wtd .....	146
3.4.16 Math_WtdAndCheck .....	147
3.4.17 Math_Correlation .....	149
3.5 计算服务(Apace 计算管理) .....	150
3.5.1 AddAigorithm .....	151
3.5.2 DeleteAigorithm .....	151
3.5.3 DeleteAigorithmElement .....	151
3.5.4 Contains .....	152
3.5.5 UpdateAigorithm .....	152
3.5.6 GetAllAigorithm .....	152
3.5.7 AddFiledll .....	153
3.5.8 UpdateFiledll .....	153
3.5.9 GetIntervalAigorithm .....	154
3.5.10 GetAllLocalConfig .....	154
3.5.11 UpdateAllLocalConfig .....	155
3.5.12 UpdateIntervalConfig .....	157
3.5.13 GetIntervalConfig .....	157
3.5.14 GetIntervalPublicCode .....	158
3.5.15 UpdataIntervalPublicCode .....	158
3.6 告警服务(Apace 告警检测管理) .....	158
3.6.1 AddAlarm .....	159
3.6.2 DeleteAlarm .....	159
3.6.3 DeleteAlarmElement .....	159
3.6.4 Contains .....	160
3.6.5 UpdateAlarm .....	160
3.6.6 GetAllAlarm .....	160
3.6.7 GetIntervalAlarm .....	161
3.6.8 GetAllAlarmMessage .....	161
3.6.9 GetAllLocalConfig .....	162
3.6.10 UpdateAllLocalConfig .....	162
3.6.11 GetAlarmMessage .....	163
3.6.12 GetAlarmMessageForAlarmCode .....	164

---

3.6.13 GetAlarmMessagesForAlarmCode .....	错误！未定义书签。
---	-----------

# 第 1 章

## Apace 开发环境介绍

Apace 作为新一代实时数据库产品完全支持 .Net Framework, 并且有非常高效的访问性能, Apace 实时数据产品系列中提供了支持 .Net Framework SDK 开发包, 通过 SDK, 开发人员能快速的掌握并了解 Apace 实时数据库产品优秀的分布式架构和丰富对外的访问接口。

说明: 该开发手册主要针对 .Net Framework 3.0 版本以上的应用解决方案, 并所用的实例代码为 C# 语言编写。

### 1.1 程序集



- Apace.Client.dll  
Apace 客服端访问程序集
- Apace.Common.dll  
Apace 客户端公共实例程序集
- Apace.DataContract.dll  
Apace 客户端数据契约程序集
- Apace.Window.Compress.dll  
Apace 客户端组件程序集

需要添加的 .Net Framework 公共程序集

System

System.Core

System.Runtime.Serialization

System.ServiceModel

System.Xml

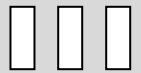
System.Xml.Linq

引用完成以后如下图所示 (以 VS2010 为例):

## 引用

- > Apace.Client
- > Apace.Common
- > Apace.DataContract
- > Apace.Window.Compress
- > Microsoft.CSharp
- > System
- > System.Core
- > System.Data
- > System.Data.DataSetExtensions
- > System.Deployment
- > System.Drawing
- > System.Runtime.Serialization
- > System.ServiceModel
- > System.Windows.Forms
- > System.Xml
- > System.Xml.Linq

## 1.2 APP.CONFIG 连接配置



Apace 客户端连接配置适合所有 .Net Framework 解决方案包括 Web 项目、Form 项目、WPF 等。

配置文件（WEB 项目是 web.config 桌面应用项目为 app.config）：

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="Apace.Client" type="Apace.Client.ApaceNetConfigSection, Apace.Client" />
  </configSections>
  <Apace.Client ServiceNodeName="apace">
    <ServiceNode Name="apace" DDMIPAddress="127.0.0.1" Port="2006" UserName="apace" Password="apace"
    NetType="NetTcp" IsBuffer="true"></ServiceNode>
  </Apace.Client>
</configuration>
```

配置属性说明：

**ServiceNodeName**：配置服务节点别名默认使用“apace”，无需修改

**ServiceNode 的 Name**：服务节点别名默认使用“apace”，无需修改

**DDMIPAddress**：Apace 服务器地址

**Port**：连接端口（NetTcp 方式默认为 2006，NetHttp 方式默认为 80）

**UserName**：登录用户名

**Password**：登录用户密码

**NetType**：连接目标服务器所使用的网络协议，可选值：NetTcp、NetHttp

**IsBuffer**：是否开始本地缓存

## 1.3 标签点数据对象




标签点数据对象用于描述 Apace 数据的各种形态，标签点数据对象发布于服务元数据中，服务元数据允许客户端将与平台、技术无关的数据类型表示形式转换为客户端本地的表示形式。

### 实时数据：

- 1 程序集： Apace.DataContract;
- 2 命名空间： Apace.DataContract.File;
- 3 契约： McbData;
- 4 形态

### 构造函数

	名称	说明
	McbData	初始化 McbData 类的新实例。

### 方法

	名称	说明
	GetHashCode	返回此实例的哈希代码。（继承自 Attribute。）
	GetType	获取当前实例的 Type。（继承自 Object。）
	MemberwiseClone	创建当前 Object 的浅表副本。（继承自 Object。）
	ToString	返回表示当前 Object 的 String。（继承自 Object。）


### 属性

	名称	说明
	PointName	获取或设数据点标签名称格式:[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含'.'字符。
	PointValue	获取或设置一个值，表示点标签的数据值，类型为 double。
	PointTime	获取或设置一个值，表示点标签时间戳，类型为 int。
	Time	获取一个值，表示点标签时间戳，类型为 DateTime。
	BindingName	获取绑定格式点名，也可理解为短点名：[PointName]。
	State	获取点标签状态，枚举类型 PointValueState。
	PointState	获取或设置一个值，表示点标签值状态，类型为 Byte。
	PointDescribe	获取或设置点标签描述文本，总长度不能超过 20 个字节。
	PointUnit	获取或设置点标签单位文本，总长度不能超过 8 个字节。
	Position	获取或设置点标签在数据流中的位置。
	ServiceDirName	获取点标签名称所对应的服务目录名：[ServiceName].[DirName]



**点标签基本信息:**

- 1 程序集: Apac.DataContract;
  - 2 命名空间: Apac.DataContract.File;
  - 3 契约: PcmData;
  - 4 形态
- 构造函数



	名称	说明
	PcmData	初始化 PcmData 类的新实例。

**方法**

	名称	说明
	GetHashCode	返回此实例的哈希代码。 (继承自 Attribute。)
	GetType	获取当前实例的 Type。 (继承自 Object。)
	MemberwiseClone	创建当前 Object 的浅表副本。 (继承自 Object。)
	ToString	返回表示当前 Object 的 String。 (继承自 Object。)

**属性**

	名称	说明
	PointName	获取或设数据点标签名称格式:[ServiceName].[DirName].[PointName], 总长度为 20 个字节其中包含'.'字符。
	HighAlarm	获取或设置一个值, 表示点标签的发生高报的界限值。
	BosomAlarm	获取或设置一个值, 表示点标签的发生中报的界限值。
	LowAlarm	获取或设置一个值, 表示点标签的发生低报的界限值。
	AllowMaxValue	获取或设置一个值, 表示点标签的有效值上限, 当与 AllowMinValue 相等时界限值将不起作用
	AllowMinValue	获取或设置一个值, 表示点标签的有效值下限, 当与 AllowMaxValue 相等时界限值将不起作用
	OperationType	获取或设置点标签的业务类型, 长度 20 个字节, 如: 锅炉专业点或气机专业点。
	OperationCode	获取或设置点标签的业务代码, 长度 20 个字节。
	DataShape	获取或设置一个值, 表示点标签的数据形态, 0: 动态 1: 静态 2: 自适应
	StaticValue	获取或设置一个值, 表示点标签静态时的值。
	Precision	获取或设置一个值, 表示点标签最大允许的精度值, 255 表示不做控制
	HisStartTiem	获取或设置一个值, 表示存放标签点历史数据的开始时间。
	BindingName	获取绑定格式点名, 也可理解为短点名: [PointName]。
	PointDescribe	获取或设置点标签描述文本, 总长度不能超过 20 个字节。

	PointUnit	获取或设置点标签单位文本，总长度不能超过 8 个字节。
	Position	获取或设置点标签在数据流中的位置。

## 历史数据：

- 1 程序集： Apave.DataContract;
- 2 命名空间： Apave.DataContract.File;
- 3 契约： HisRowData;
- 4 形态







## 构造函数

	名称	说明
	HisRowData	初始化 HisRowData 类的新实例。

## 方法

	名称	说明
	GetHashCode	返回此实例的哈希代码。（继承自 Attribute。）
	GetType	获取当前实例的 Type。（继承自 Object。）
	MemberwiseClone	创建当前 Object 的浅表副本。（继承自 Object。）
	ToString	返回表示当前 Object 的 String。（继承自 Object。）

## 属性

	名称	说明
	PointName	获取或设数据点标签名称格式:[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含'.'字符。
	PointValue	获取或设置一个值，表示点标签的数据值，类型为双精度。
	PointTime	获取或设置一个值，表示点标签时间戳，类型为整形。
	Time	获取，表示点标签时间戳，类型为 DateTime。
	State	获取点标签状态，枚举类型 PointValueState。
	PointState	获取或设置一个值，表示点标签值状态，类型为 Byte。

## 告警消息：

- 1 程序集： Apave.DataContract;
- 2 命名空间： Apave.DataContract.File;
- 3 契约： AlarmMessage;
- 4 形态

## 构造函数

	名称	说明
--	----	----

	AlarmMessage	初始化 AlarmMessage 类的新实例。
---	--------------	-------------------------

## 方法


	名称	说明
	GetHashCode	返回此实例的哈希代码。（继承自 Attribute。）
	GetType	获取当前实例的 Type。（继承自 Object。）
	MemberwiseClone	创建当前 Object 的浅表副本。（继承自 Object。）
	ToString	返回表示当前 Object 的 String。（继承自 Object。）

## 属性



	名称	说明
	MesgGUID	获取或设置告警消息的唯一表示 GUID 码
	AlarmTarge	获取或设置检测告警的规则集合。
	Describe	获取或设置对告警的描述文本。
	ServiceName	获取或设置检测告警所对应的服务名称。
	DirName	获取或设置检测告警所对应的目录名称。
	TriggerTime	获取或设置告警发生的时间。
	TriggerPointName	获取或设置满足告警发生的检测点集合。
	AppCode	获取或设置告警对应的应用标示代码。
	SendTarget	获取或设置告警产生后发送的目标地址。（暂被否决）
	AlarmCode	获取或设置告警的唯一表示代码。



## 告警规则：

- 1 程序集： Apace.DataContract;
  - 2 命名空间： Apace.DataContract.File;
  - 3 契约： AlarmElement;
  - 4 形态
- 构造函数













	名称	说明
	AlarmElement	初始化 AlarmElement 类的新实例。

## 方法

	名称	说明
	GetHashCode	返回此实例的哈希代码。（继承自 Attribute。）
	GetType	获取当前实例的 Type。（继承自 Object。）

	MemberwiseClone	创建当前 Object 的浅表副本。（继承自 Object。）
	ToString	返回表示当前 Object 的 String。（继承自 Object。）


## 属性

	名称	说明
	Space	获取或设置一个值，标示告警检测的时间范围，单位秒。
	MaxValue	获取或设置一个值，标示告警检测的值的最大有效值。
	MinValue	获取或设置一个值，标示告警检测的值的最小有效值。
	TopValue	获取或设置一个值，标示告警检测的值的顶部界限值。
	BottomValue	获取或设置一个值，标示告警检测的值的底部界限值。
	WatchPointName	获取或设置告警检测点名称。
	Step	获取或设置一个值，标示告警检测的执行的步长，单位秒。
	CompareData	获取或设置告警参照数据集合。
	LastHisValueTime	获取或设置告警检测最后存放数据点的时间。
	CorrelValue	获取或设置一个值，标示满足告警的相识度值，单位百分比。
	LinkNextState	获取或设置一个状态，标示通过与、或的形式连接下一告警规则。
	PointName	获取或设置告警检测的点名称。
	CheckMethod	获取或设置告警检测的模式，ArarmCheckMethod 枚举。
	GetAlarmResult	获取告警检测的结果，客户端否决。
	IsCheck	获取或设置一个状态，表示是否需要继续检测。
	HisData	获取或设置一组数据，表示实际检测点按照步长产生的历史数据。

## 计算实体：


- 1 程序集： Apace.DataContract;
- 2 命名空间： Apace.DataContract.File;
- 3 契约： CalcElement;
- 4 形态

## 构造函数

	名称	说明
	CalcElement	初始化 CalcElement 类的新实例。

## 方法

	名称	说明
	GetHashCode	返回此实例的哈希代码。（继承自 Attribute。）
	GetType	获取当前实例的 Type。（继承自 Object。）
	MemberwiseClone	创建当前 Object 的浅表副本。（继承自 Object。）

	ToString	返回表示当前 Object 的 String。（继承自 Object。）
---	----------	--------------------------------------

## 属性

	名称	说明
	CalcPointName	获取或设置计算点名称。
	CalcPointDesc	获取或设置计算点描述。
	Assemblys	获取或设置计算法则需要用到的程序集合。
	Using	获取或设置计算法则引用的命名空间集合。
	Language	获取或设置计算法则执行语言。
	PointInfo	获取或设置需要写入的计算点信息集合
	Interval	获取或设置一个值，标示执行计算的周期。
	WritePointName	获取写入点集合。
	TargetServiceDir	获取或设置写入点所对应的目标服务目录名。
	Algorithm	获取或设置计算法则代码。

## 1.4 数据契约（枚举）






## 点标签状态：

- 1 程序集： Apace.DataContract;
- 2 命名空间： Apace.DataContract.File;
- 3 枚举： PointValueState;

	名称	说明
	NoInit	数据点标签没用被初始化。
	OK	数据点值状态正常。
	Lapse	数据点值失效。
	HighAlarm	数据点值发生高报。
	BosomAlarm	数据点值发生中报。
	LowAlarm	数据点值发生低报。
	Exceed	数据点值超出了设定的范围。

## 告警检测方式：

- 1 程序集： Apace.DataContract;
- 2 命名空间： Apace.DataContract.File;
- 3 枚举： ArarmCheckMethod;

	名称	说明
	Contrast	曲线趋势相识度检测法。
	Area	曲线区间检测法。
	Popple	曲线波动检测法。

## 第 2 章

### 接口调用方法

#### 2.1 构造相关方法



##### 创建客户端实体

通过 App.Config 配置方式登录:

1 在 App.Config 文件中加入连接配置节;

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="Apace.Client" type="Apace.Client.ApaceNetConfigSection, Apace.Client" />
  </configSections>
  <Apace.Client ServiceNodeName="apace">
    <ServiceNode Name="apace" DDMIPAddress="127.0.0.1" Port="2006" UserName="apace" PassWord="apace"
    NetType="NetTcp" IsBuffer="true"></ServiceNode>
  </Apace.Client>
</configuration>
```

2 调用方法;

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```



##### 关键点:

`NetClient` Apace 客户端访问实体类

`NetClientManage` 为应用提供对 Apace 客户端的管理类

`NetClientManage.GetNetClient()` 通过配置节获得到全局单例客户端对象

`NetClientManage.CreateNewNetClient()` 通过配置节创建新的客户端对象

通过参数方式登录:

```
Apace.Client.NetClient client = NetClient.CreateNet("127.0.0.1", "apace", "apace", NetType.NetTcp, true);
```



##### 关键点:

`NetClient.CreateNet` 创建一个新的客户端实体

参数: `CreateNet(string IPAddress, string UserName, string PassWord, NetType NetType, bool IsBuffer)`

## 2.2 点标签数据相关方法


















点标签数据相关方法是对采集标签点基本信息的访问与操作，Apace 实时数据库在处理点标签的基本信息时效率非常的高，一秒钟可处理 5000 点的添加、30 万点的删除、30 万点的检索。


### 2.2.1 AddNewPoint

```
bool AddNewPoint(PcmData value);
```

说明：向目标服务器提交新的点标签信息。

PARAMETER	DESCRIPTION		
PcmData	点标签基本信息		
		名称	说明
		PointName	获 取 或 设 数 据 点 标 签 名 称 格 式:[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含'.'字符。
		HighAlarm	获取或设置一个值，表示点标签的发生高报的界限值。
		BosomAlarm	获取或设置一个值，表示点标签的发生中报的界限值。
		LowAlarm	获取或设置一个值，表示点标签的发生低报的界限值。
		AllowMaxValue	获 取 或 设 置 一 个 值 ， 表 示 点 标 签 的 有 效 值 上 限 ， 当 与 AllowMinValue 相等时界限值将不起作用
		AllowMinValue	获 取 或 设 置 一 个 值 ， 表 示 点 标 签 的 有 效 值 下 限 ， 当 与 AllowMaxValue 相等时界限值将不起作用
		OperationType	获取或设置点标签的业务类型,长度 20 个字节，如：锅炉专业点或气机专业点。
		OperationCode	获取或设置点标签的业务代码，长度 20 个字节。
		DataShape	获取或设置一个值，表示点标签的数据形态,0: 动态 1:静态 2: 自适应
		StaticValue	获取或设置一个值，表示点标签静态时的值。
		Precision	获取或设置一个值，表示点标签最大允许的精度值，255 表示不做控制
		HisStar Tiem	获取或设置一个值，表示存放标签点历史数据的开始时间。
		BindingName	获取绑定格式点名，也可理解为短点名：[PointName]。
		PointDescribe	获取或设置点标签描述文本，总长度不能超过 20 个字节。
	PointUnit	获取或设置点标签单位文本，总长度不能超过 8 个字节。	



		Position	获取或设置点标签在数据流中的位置。
--	---	----------	-------------------

RETURN	DESCRIPTION
bool	提交新点基本信息成功状态。



## 实例演示

```

Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();

PcmData point = new PcmData();
point.PointName = "DDM.Apace.T5";
point.OperationCode = "MHXTO1JIZU001";
point.OperationType= "GLXT";
point.HighAlarm = 50000;
point.LowAlarm = 30000;
point.BosomAlarm = -100;
point.AllowMax Value = 80000;
point.AllowMin Value = -1000;
point.DataShape =0;
point.Static Value = 0;
point.PointUnit = "度";
point.PointDescribe ="Apace演示点标签";
point.Precision = 7;

bool returnState= client.AddNewPoint(point);
if (returnState)
{
    //操作成功
}
else
{
    //操作失败
}
    
```



## 相关方法:

AddNewPoint (string PointName, string PointDescribe, string PointUnit) 快速创建一个新的点标签，其它属性使用系统默认值。

AddNewPoints(PcmData[] PointList, string ServiceDirName) 批量创建点标签到目标服务目录

## 2.2.2 Contains

```
bool Contains(string PointName);
```

说明：检测点标签是否已经存在。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。

RETURN	DESCRIPTION
bool	True 已经存在，False 不存在。

## 2.2.3 EditPoint

```
bool EditPoint(PcmData value);
```

说明：编辑已存在点标签基本信息。

PARAMETER	DESCRIPTION
PcmData	数据格式请参见数据契约。

RETURN	DESCRIPTION
bool	True 编辑成功，False 编辑失败。 失败的原因：1 点名不存在 2 点所对应的服务目录不存在 3 目标服务器拒绝访问



### 实例演示

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
PcmData point = new PcmData();
point.PointName = "DDM.Apace.T5";
point.OperationCode = "MHXTO1JIZU001";
point.OperationType= "GLXT";
point.HighAlarm = 20000;
point.BosomAlarm = 15000;
point.LowAlarm = 10000;
```

```

    point.AllowMax Value = 30000;
    point.AllowMin Value = -1000;
    point.DataShape =0;
    point.Static Value = 0;
    point.PointUnit = "度";
    point.PointDescribe ="Apace演示";
    point.Precision = 7;

    bool returnState= client.Contains(point.PointName);
    if(returnState)
    {
        if(client.EditPoint(point))
        {
            //编辑成功
        }
    }
    else
    {
        //检测不存在
    }

```

## 2.2.4 GetPointInfo

```
PcmData GetPointInfo(string PointName)
```

说明：获得服务器点标签基本信息。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。

RETURN	DESCRIPTION
PcmData	点标签信息实体，请参见数据契约



相关方法：

`PcmData[] GetServiceDirPointInfo(string ServiceName, string DirName)` 获取对应服务目录下所有的点标签信息。

## 2.2.5 DeletePoint

```
bool DeletePoint(string PointName)
```

说明：删除点标签。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。

RETURN	DESCRIPTION
bool	True 删除成功，False 删除失败。 失败原因：1 点标签不存在 2 服务器拒绝访问



相关方法：

`void DeletePoint(string[] PointCollect)` 批量删除点标签。

## 2.3

## 实时数据访问相关方法



实时数据相关方法是对采集标签点时序数据进行高速缓存相关操作，Apace 实时数据库利用最新的高速缓存与多级缓存技术、再结合快速索引技术在一秒钟内能对上百万条数据执行检索操作，和上几十万条的数据时序追加，有着极高的数据吞吐率。

### 2.3.1 SetRealData

`bool SetRealData(McbData Realvalue)`

说明：设置一个新时序值追加到高速缓存。

PARAMETER	DESCRIPTION		
McbData	实时数据		
		名称	说明
		PointName	获取或设数据点标签名称格式：[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含'.'字符。
		PointValue	获取或设置一个值，表示点标签的数据值，类型为双精度。
		PointTime	获取或设置一个值，表示点标签时间戳，类型为整形。
		Time	获取一个值，表示点标签时间戳，类型为 DateTime。

	BindingName	获取绑定格式点名，也可理解为短点名：[PointName]。
	State	获取点标签状态，枚举类型 PointValueState。
	PointState	获取或设置一个值，表示点标签值状态，类型为 Byte。
	PointDescribe	获取或设置点标签描述文本，总长度不能超过 20 个字节。
	PointUnit	获取或设置点标签单位文本，总长度不能超过 8 个字节。
	Position	获取或设置点标签在数据流中的位置。
	ServiceDirName	获取点标签名称所对应的服务目录名：[ServiceName]. [DirName]

RETURN	DESCRIPTION
bool	True 追击成功，False 追击失败。 失败原因：1 点标签不存在 2 时序发生异常，一般为新追加时间戳小于点的实际时间戳 3 服务器拒绝访问



### 实例演示

```

//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好内存数据
McbData realpoint = new McbData();
realpoint.PointName = "DDM.Apace.T1";
realpoint.Point Value = new Random((int)DateTime.Now.Ticks).NextDouble() * 10;
//第三步：将时间转换成Utc时间
realpoint.PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);
//第四步：向服务器追加时序数据
bool returnState= client.SetRealData (realpoint);
if(returnState)
{
    MessageBox.Show("提交数据成功！");
}
else
{
    MessageBox.Show("提交数据失败！");
}

```



### 关键点：

Apace.Common.BaseConfig.GetUtcTime(DateTime.Now) 将当前系统时间转换为 Apace Utc 时间格式。



### 相关方法:

`bool SetRealData(string PointName, DateTime PointTime, Double value)` 通过形参方式快速追加时序数据

`SetRealData(string PointName, Double value)` 通过形参方式快速追加时序数据, 并使用远程 Apace 实时数据库系统当前的最新时间作为时序值



### 批量追加实例演示

```
//第一步: 得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步: 准备好批量内存数据
McbData[] realpoint = new McbData[3];
realpoint[0] = new McbData();
realpoint[0].PointName = "DDM. Apace. T1";
realpoint[0].PointValue = new Random((int)DateTime.Now.Ticks).NextDouble() * 10;
realpoint[0].PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);

realpoint[1] = new McbData();
realpoint[1].PointName = "DDM. Apace. T2";
realpoint[1].PointValue = new Random().NextDouble() * 20;
realpoint[1].PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);

realpoint[2] = new McbData();
realpoint[2].PointName = "DDM. Apace. T3";
realpoint[2].PointValue = new Random().NextDouble() * 30;
realpoint[2].PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);

//第三步: 向服务器追加时序数据
client.SetRealDataCollect(realpoint);
```



### 关键点:

`Apace.Common.BaseConfig.GetUtcTime(DateTime.Now)` 将当前系统时间转换为 Apace Utc 时间格式。



### 相关方法:

`void SetRealDataCollect(McbData[] Realvalue, string TargetServiceDir)` 将批量数据追加到指定的 Apace 服务目录, 注意该方法会提升追加的效率但批量数据中的点标签名称中的服务目录名必须和目标服务目录名相同, TargetServiceDir 格式为: [ServiceName]. [DirName]

说明:

批量数据的追加会减少网络 I/O 的操作, 极大提高追加的性能, 但是追加数据过多会影响网络吞吐效率, 所以合理的使用其追加的数据量值非常的重要, 建议单线程每秒追加的数据量不要操作过 5 万条。



### 事务性批量追加实例演示

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：开启追加数据事务
client.BeginTransaction();
client.SetRealData("DDM.Apac.T1", DateTime.Now, new Random().NextDouble() * 10);
client.SetRealData("DDM.Apac.T2", DateTime.Now, new Random().NextDouble() * 20);
client.SetRealData("DDM.Apac.T3", DateTime.Now, new Random().NextDouble() * 30);
client.SetRealData("DDM.Apac.T4", DateTime.Now, new Random().NextDouble() * 40);
//第三步：介绍追加数据事务
client.EndTransaction();
```



#### 关键点：

`void BeginTransaction()` 开始追加事务操作。

`void EndTransaction()` 结束追加事务操作。

说明：`NetClient` 在使用单列的情形下，要考虑多线程的并发事务提交操作，如果为多线程安全，事务的追加有利于提升操作的性能，在数据量较大的情况下可以考虑事务追加方式与批量提交方式。



#### 相关方法：

`void SetRealDataCollect(McbData[] Realvalue, string TargetServiceDir)` 将批量数据追加到指定的 Apac 服务目录，注意该方法会提升追加的效率但批量数据中的点标签名称中的服务目录名必须和目标服务目录名相同，TargetServiceDir 格式为：[ServiceName].[DirName]

## 2.3.2 GetPointValue

```
McbData GetPointValue(string PointName)
```

说明：获得点标签当前高速缓存中的时序值信息。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]

RETURN	DESCRIPTION				
McbData	<table border="1"> <thead> <tr> <th>名称</th><th>说明</th></tr> </thead> <tbody> <tr> <td>  PointName </td><td>获取或设数据点标签名称格式：[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含 '.' 字符。</td></tr> </tbody> </table>	名称	说明	 PointName	获取或设数据点标签名称格式：[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含 '.' 字符。
名称	说明				
 PointName	获取或设数据点标签名称格式：[ServiceName].[DirName].[PointName]，总长度为 20 个字节其中包含 '.' 字符。				

		PointValue	获取或设置一个值，表示点标签的数据值，类型为双精度。
		PointTime	获取或设置一个值，表示点标签时间戳，类型为整形。
		Time	获取一个值，表示点标签时间戳，类型为 DateTime。
		BindingName	获取绑定格式点名，也可理解为短点名：[PointName]。
		State	获取点标签状态，枚举类型 PointValueState。
		PointState	获取或设置一个值，表示点标签值状态，类型为 Byte。
		PointDescribe	获取或设置点标签描述文本，总长度不能超过 20 个字节。
		PointUnit	获取或设置点标签单位文本，总长度不能超过 8 个字节。
		Position	获取或设置点标签在数据流中的位置。
		ServiceDirName	获取点标签名称所对应的服务目录名：[ServiceName].[DirName]



## 实例演示

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManager.GetNetClient();
//第二步：准备好点名称
String PointName = "DDM.Apac.T1"; //[服务名].[目录名].[点名称]
//第三步：获取点标签
McbData returnValue = client.GetPointValue(PointName);
if(returnValue!=null)
{
    //获取成功代码
}
else
{
    //获取失败代码
}
```



## 相关方法：

`McbData[] GetPointValueCollect(string[] PointNameCollect)` 批量获得点标签当前高速缓存中的时序值信息。

`McbData[] GetServiceDirPoint(string ServiceDirName)` 获得指定服务目录下所有高速缓存中的点时序信息，注意：ServiceDirName 格式为：[ServiceName].[DirName]

说明：批量高速缓存时序数据的获取会减少网络 I/O 的操作，极大提高访问的性能，但是访问数据过多会影响网络吞吐效率，所以合理的使用其访问的数据量值非常的重要，建议单线程每秒追加的数据量不要操作过 30 万条。

## 2.3.3 FindPointState



```
McbData[] FindPointState(string ServiceName, string DirName, PointValueState value)
```

说明：获得高速缓存中当前时序值，符合指定值状态的所有点标签。

PARAMETER	DESCRIPTION																								
ServiceName	Apace 服务器名， 在安装 Apace 实时数据库服务端程序时所指定的服务器别名, 服务别名是在 Apace 分布式网络中对网络终端的唯一标示。																								
DirName	目录名，在单台 Apace 实时数据服务器内可以定义一个或多个目录名。																								
value	<div>点值态 PointValueState 枚举值如下：</div> <table><tr><td></td><td>名称</td><td>说明</td></tr><tr><td>⇒💎</td><td>NoInit =0</td><td>数据点标签没用被初始化。</td></tr><tr><td>⇒💎</td><td>OK =1</td><td>数据点值状态正常。</td></tr><tr><td>⇒💎</td><td>Lapse =2</td><td>数据点值失效。</td></tr><tr><td>⇒💎</td><td>HighAlarm =3</td><td>数据点值发生高报。</td></tr><tr><td>⇒💎</td><td>BosomAlarm =4</td><td>数据点值发生中报。</td></tr><tr><td>⇒💎</td><td>LowAlarm =5</td><td>数据点值发生低报。</td></tr><tr><td>⇒💎</td><td>Exceed =6</td><td>数据点值超出了设定的范围。</td></tr></table>		名称	说明	⇒💎	NoInit =0	数据点标签没用被初始化。	⇒💎	OK =1	数据点值状态正常。	⇒💎	Lapse =2	数据点值失效。	⇒💎	HighAlarm =3	数据点值发生高报。	⇒💎	BosomAlarm =4	数据点值发生中报。	⇒💎	LowAlarm =5	数据点值发生低报。	⇒💎	Exceed =6	数据点值超出了设定的范围。
	名称	说明																							
⇒💎	NoInit =0	数据点标签没用被初始化。																							
⇒💎	OK =1	数据点值状态正常。																							
⇒💎	Lapse =2	数据点值失效。																							
⇒💎	HighAlarm =3	数据点值发生高报。																							
⇒💎	BosomAlarm =4	数据点值发生中报。																							
⇒💎	LowAlarm =5	数据点值发生低报。																							
⇒💎	Exceed =6	数据点值超出了设定的范围。																							

RETURN	DESCRIPTION
McbData[]	实例数组，相关属性请参见数据契约



### 实例演示

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好匹配状态值
PointValueState checkState = PointValueState.OK;
//第三步：删除点标签
McData[] returnValue= client.FindPointState("DDM","Apace", checkState);
if(returnValue!=null && returnValue.Length >0)
{
    //查找成功
}
else
{
    //没有符合条件的标签
}
```

### 2.3.4 FindPointName

```
McbData[] FindPointName(string ServiceName, string DirName, string value)
```

说明：获得高速缓存中点标签名称包含指定字符串的所有点标签。

PARAMETER	DESCRIPTION
ServiceName	Apace 服务器名，在安装 Apace 实时数据库服务端程序时所指定的服务器别名, 服务别名是在 Apace 分布式网络中对网络终端的唯一标示。
DirName	目录名，在单台 Apace 实时数据服务器内可以定义一个或多个目录名。
value	用于匹配过滤点名的字符串，如 "test" 满足条件的有如下形式： 1: *test* 2: test* 3: *test

RETURN	DESCRIPTION
McbData[]	实例数组，相关属性请参见数据契约



#### 相关方法：

`McbData[] FindPointNameRegex(string ServiceName, string DirName, string rule)` 使用正则表达式规则来匹配点名, 演示: (过滤点标签名称以“test”字符串开始并后续只跟一个字符的点标签)  
`String FindValue ="test? ";`

`McbData[] returnValue= Login.netClient. FindPointNameRegex("DDM","Apace", FindValue);`

演示: (到的服务目录下所有点标签)

`String FindValue =".";`

`McbData[] returnValue= Login.netClient. FindPointNameRegex("DDM","Apace", FindValue);`

`McbData[] FindPointDescribe(string ServiceName, string DirName, string value)` 获得高速缓存中，点标签描述包含指定字符串的所有点标签。

`McbData[] FindPointDescribeRegex(string ServiceName, string DirName, string rule)` 使用正则表达式规则来匹配点描述，同 `FindPointNameRegex` 类似。

`McbData[] FindOperationType(string ServiceName, string DirName, string value)` 获得高速缓存中，点标签业务类型包含指定字符串的所有点标签。

`McbData[] FindOperationCode(string ServiceName, string DirName, string value)` 获得高速缓存中，点标签业务代码包含指定字符串的所有点标签。

`McbData[] FindOperationCodeAndType(string ServiceName, string DirName, string codevalue, string typevalue)`

点标签业务代码与业务类型同时匹配的所有点标签。

说明：Apace 实时数据库提供了丰富的快速检索数据点标签的方法，上层应用可根据业务需求来快速检索相关数据，Apace 实时数据库具有高效的检索效率，在一秒钟内可达到百万级的数据检索。

### 2.3.5 GetSystemTime

```
DateTime GetSystemTime(string ServiceName, string DirName)
```

说明：获得点标签当前高速缓存中的时序值信息。

PARAMETER	DESCRIPTION
ServiceName	Apace 服务器名，在安装 Apace 实时数据库服务端程序时所指定的服务器别名, 服务别名是在 Apace 分布式网络中对网络终端的唯一标示。
DirName	目录名，在单台 Apace 实时数据服务器内可以定义一个或多个目录名。

RETURN	DESCRIPTION
DateTime	目标服务器当前的系统时间



#### 实例演示

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：得到指定目录服务目录系统时间
DateTime returnValue= client.GetSystemTime("DDM","Apace");
```



#### 关键点：

如果目标服务器存在多个服务目录，如（Apace, Apace1）多个目录情况下，使用其中任何一个目录都是允许的，并且返回的时间值相同，例：

```
DateTime returnValue= Login.netClient.GetSystemTime("DDM","Apace");
```

与

```
DateTime returnValue= Login.netClient.GetSystemTime("DDM","Apace1");
```

具有相同的效果。

### 2.3.6 GetSystemUtcTime

```
int GetSystemUtcTime(string ServiceName, string DirName)
```

说明：获得点标签当前高速缓存中的时序值信息。

PARAMETER	DESCRIPTION
-----------	-------------

ServiceName	Apace 服务器名，在安装 Apace 实时数据库服务端程序时所指定的服务器别名, 服务别名是在 Apace 分布式网络中对网络终端的唯一标示。
DirName	目录名，在单台 Apace 实时数据服务器内可以定义一个或多个目录名。

RETURN	DESCRIPTION
int	Apace 实时数据 Utc 时间



### 关键点:

如果目标服务器存在多个服务目录，如（Apace, Apacel）多个目录情况下，使用其中任何一个目录都是允许的，并且返回的时间值相同，例：

```
int returnValue= Login.netClient.GetSystemTime("DDM","Apace");
```

与

```
int returnValue= Login.netClient.GetSystemTime("DDM","Apacel");
```

具有相同的效果。

## 2.4

## 历史数据访问相关方法






历史数据相关方法是对采集标签点时序历史数据进行相关操作，Apace 实时数据库采用高效的压缩方式与索引技术能快速的检索磁盘中所存放的历史数据。




### 2.4.1 GetHisBufferData

```
HisRowData[] GetHisBufferData(string pointName)
```

说明：得到当前历史服务缓冲区中指定点标签的时序数据集合。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。

RETURN	DESCRIPTION		
HisRowData[]		名称	说明
		PointName	获取或设数据点标签名称格式:[ServiceName].[DirName].[PointName],总长度为 20 个字节其中包含'.'字符。
		PointValue	获取或设置一个值，表示点标签的数据值，类型为双精度。
		PointTime	获取或设置一个值，表示点标签时间戳，类型为整形。

	Time	获取，表示点标签时间戳，类型为 DateTime。
	State	获取点标签状态，枚举类型 PointValueState。
	PointState	获取或设置一个值，表示点标签值状态，类型为 Byte。



### 实例演示

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.T1";
//第三步：删除点标签
HisRowData [] returnHisdata= client.GetHisBufferData(PointName);
if(returnHisdata!=null && returnHisdata.Length >0)
{
    //成功
}
else
{
    //失败或无历史数据
}
```



### 关键点：

GetHisBufferData 是非常高效的历史时序数据检索方法，此方法返回最新缓冲区中的历史数据，但是 GetHisBufferData 方法并不保证每次都能返回相同记录数的值。

## 2.4.2 GetHisData

```
HisRowData[] GetHisData(string pointName, DateTime stime, DateTime etime)
```

说明：得到指定时间段内的所有历史时序数据。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。
stime	DateTime 检索历史数据开始时间
etime	DateTime 检索历史数据结束时间

RETURN	DESCRIPTION
HisRowData[]	参见数据契约



## 实例演示

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();

//第二步：准备好点名
String PointName = "DDM.Apace.test1";

//第三步：删除点标签
HisRowData[] returnHisdata=Login.netClient.GetHisData(PointName, DateTime.Parse("2009-10-22 00:12:21"),
DateTime.Parse("2009-10-23 00:12:21"));
If(returnHisdata!=null && returnHisdata.Length >0)
{
    //存在历史数据代码
}
Else
{
    //不存在历史数据代码
}
```



### 关键点：

GetHisData 是用于检索指定标签点在某段时间范围内实践发生的数据集合，该方法不会对历史数据进行任何的处理，可理解为绝对的实际变化的时序值。

## 2.4.3 GetHisData(重写)

```
HisRowData[] GetHisData(string pointName, DateTime stime, DateTime etime, int step)
```

说明：得到指定时间段内的所有历史时序数据。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。
stime	DateTime 检索历史数据开始时间。
etime	DateTime 检索历史数据结束时间。
step	时序间隔步长，值单位：秒。

RETURN	DESCRIPTION
HisRowData[]	参见数据契约



## 实例演示(按步长检索时序历史数据)

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
//第二步：准备好点名
String PointName ="DDM.Apace.test1";
//第三步：删除点标签
HisRowData[] returnHisdata=Login.netClient.GetHisData(PointName, DateTime.Parse("2009-10-22 00:12:21"),
DateTime.Parse("2009-10-23 00:12:21"), 5);
If(returnHisdata!=null && returnHisdata.Length >0)
{
    //存在历史数据代码
}
Else
{
    //不存在历史数据代码
}
```



### 关键点：

GetHisData 带步长, 是用于检索指定标签点在某段时间范围按步长时序的数据集合, 该方法会根据实际的时序历史数据再依据步长值跳跃式抓取数据。



### 相关方法：

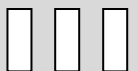
`HisRowData[] GetHisData2(string pointName, DateTime stime, DateTime etime, int step)` GetHisData2 方法用于补充 GetHisData 在大批量时序历史数据下所带来的网络延迟, 如果返回的数据量大于几十万条时最好使用 GetHisData2 方法

`Stream GetHisDataStream(string pointName, DateTime stime, DateTime etime, int step)` 以流的方式向上层应用提供时序历史数据。

`HisRowData[] GetHisDataProxy(string pointName, DateTime stime, DateTime etime, int step)` 该方法为 Apace 站点代理服务专用。

说明： 在对时序历史数据进行检索时如果时间跨度比较长, 或数据量比较大, 建议采用分时间段方式进行数据检索, 确保网络不被堵塞。

## 2.5 计算组件相关方法



请参见高级访问接口

## 2.6 告警组件相关方法



请参见高级访问接口

# 第 3 章

## 高级接口

### 3.1 DDM 服务(Apace 分布式管理中心)



DDM 服务为 Apace 服务器与分布式管理的控制中心,在调用所有数据接口之前都必须先登录到 DDM 服务经过权限的验证后才能正常的使用 Apace 服务数据。

服务对外提供的所有接口都支持同步于异步方式,上层应用系统可根据实际需求选择同步或异步方式。

#### 3.1.1 Login

```
bool Login(string userName,  
           string passWord,  
           string appName);
```

说明：登录到分布式管理中心服务器。

PARAMETER	DESCRIPTION
userName	登录用户名。
passWord	登录用户密码。
appName	登录目标客户端应用名称。

RETURN	DESCRIPTION
State	登录到服务器是否成功。



#### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```



```
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：登录
Bool returnState= ddmClient.Login("apace",
    Apac.Compress.MD5Encryption.StringToMD5("apace"), AppDomain.CurrentDomain.FriendlyName))
If(returnState)
{ //登录成功}
Else
{ //登录失败，用户名或密码错误}
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.LoginCompleted +=
new EventHandler<Apac.Client.DDMService.LoginCompletedEventArgs>(NetClient_LoginCompleted);
//第三步：开始异步登录
ddmClient.LoginAsync("apace",
    Apac.Compress.MD5Encryption.StringToMD5("apace"), AppDomain.CurrentDomain.FriendlyName));
//接受事件
void NetClient_LoginCompleted(object sender, Apac.Client.DDMService.LoginCompletedEventArgs e)
{
    if (e.Error == null)
    {
        if (e.Result)
        { //登录成功}
        else
        { //登录失败，用户名或密码错误}
    }
    else
    {
        //登录时发生网络异常
    }
}
```



### 关键点：

Apac.Compress.MD5Encryption.StringToMD5("apace") 用户密码采用 MD5 算法进行加密，在登录时发送到服务端的密码必须经过 MD5 编码。

AppDomain.CurrentDomain.FriendlyName 表示当前应用的标示字符串，当然也可以为空，也可以自定义应用标示。



### 相关方法（委托）：

System.IAsyncResult BeginLogin(string userName, string passWord, string appName, System.AsyncCallback callback, object asyncState) 开始异步委托  
 EndLogin(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.2 UnLogin

```
void UnLogin();
```

说明：注销登录。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：登录
Bool returnState= ddmClient.UnLogin ();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.UnLoginCompleted +=
new EventHandler<System.ComponentModel.AsyncCompletedEventArgs>(NetClient_UnLoginCompleted);
//第三步：开始异步
ddmClient.UnLoginAsync();
//接受事件
void NetClient_UnLoginCompleted(object sender, System.ComponentModel.AsyncCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginUnLogin (System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
EndUnLogin (System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

### 3.1.3 ServiceAddress

```
Dictionary<string, RemoteService> ServiceAddress();
```

说明：得到 Apace 分布式网络中所有远程服务器地址信息。

RETURN	DESCRIPTION
Key	远程数据服务器 Apace 服务名
Value	RemoteService 分布式远程 Apace 服务器信息 Ipaddress 服务器 IP 地址 Username 登录用户名 Password 登录用户密码



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
Dictionary<string, RemoteService> returnValue= ddmClient.ServiceAddress();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.ServiceAddressCompleted +=
new EventHandler<Apace.Client.DDMService.ServiceAddressCompletedEventArgs>(NetClient_ServiceAddress);
//第三步：开始异步
ddmClient.ServiceAddressAsync();
//接受事件
void NetClient_ServiceAddress(object sender, Apace.Client.DDMService.ServiceAddressCompletedEventArgs e)
{
    if (e.Error == null)
    {

```

```

        //成功取得换回值
        Dictionary<string, RemoteService> returnValue= e.Result;
    }
    else
    {
        //网络异常
    }
}

```



#### 相关方法（委托）：

System. **IAsyncResult** BeginServiceAddress(System. **AsyncCallback** callback, **object** asyncState) 开始异步委托  
 EndServiceAddress (System. **IAsyncResult** result) 结束异步委托

说明： System. **AsyncCallback** callback 回调方法

### 3.1.4 UpdateDDMConfig

```
bool UpdateDDMConfig(string DDMXml);
```

说明： 修改分布式数据管理服务的配置。

PARAMETER	DESCRIPTION
DDMXml	Apace.DDM 为 Apace 服务器启动入口程序和各服务调度与控制中心, 但需要配合 Apace.DDM.XML 配置文件一起使用, 客服端可通过修改配置文件控制 DDM 服务的行为方式。

RETURN	DESCRIPTION
State	修改服务器配置信息是否成功。



#### 实例演示（同步）

```

//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
String xml="<xml .....>";
Bool returnValue= ddmClient.UpdateDDMConfig(xml);

```



#### 实例演示（异步）

```

//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体

```

```

Apave.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.UpdateDDMConfigCompleted += new
    EventHandler<Apave.Client.DDMService.UpdateDDMConfigCompletedEventArgs>(NetClient_UpdateDDMConfigCompleted);
//第三步：开始异步
ddmClient.ServiceAddressAsync();
//接受事件
void NetClient_UpdateDDMConfigCompleted(object sender, Apave.Client.DDMService.UpdateDDMConfigCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue= e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginUpdateDDMConfig (string DDMXml, System.AsyncCallback callback, object asyncState)  
开始异步委托  
EndUpdateDDMConfig (System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 该方法操作需谨慎，在不了解 Apave.DDM.Xml 文件格式的前提下不要使用，对于 DDM 分布式管理中心配置可使用客户端工具操作，不正确的调用会带来灾难性后果。

## 3.1.5 GetDDMConfig

```
string GetDDMConfig();
```

说明： 获取分布式数据管理中心配置信息。

RETURN	DESCRIPTION
DDMXml	分布式数据管理中心配置信息 XML 格式文档。



### 实例演示（同步）

```

//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apave.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用

```

```
String returnValue = returnValue= ddmClient. GetDDMConfig();
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：注册事件

```
ddmClient.GetDDMConfigCompleted += new
```

```
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetDDMConfigCompleted);
```

//第三步：开始异步

```
ddmClient.GetDDMConfigAsync ();
```

//接受事件

```
void NetClient_GetDDMConfigCompleted(object sender, Apace.Client.DDMService.GetDDMConfigCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得返回值
        String returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetDDMConfig (string DDMXml, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetDDMConfig(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 该方法操作需谨慎，在不了解 Apace.DDM.Xml 文件格式的前提下不要使用，对于 DDM 分布式管理中心配置可使用客户端工具操作，不正确的调用会带来灾难性后果。

## 3.1.6 GetAllConnectinfo

```
UserLoginInfo[] GetAllConnectinfo();
```

说明：获取连接到分布式管理中心所有用户连接信息。

RETURN	DESCRIPTION
UserLoginInfo	用户登录信息。 IPAddress 登录客户端地址 Port 客户端连接端口

	Stime 登录开始时间
	AppName 承载客户端应用名称



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
UserLoginInfo[] returnValue = returnValue= ddmClient.GetAllConnectinfo();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetAllConnectinfoCompleted);
//第三步：开始异步
ddmClient.GetAllConnectinfoAsync();
//接受事件
void NetClient_GetAllConnectinfoCompleted(object sender, Apace.Client.DDMService.GetAllConnectinfoCompletedEventArgs
e) {
    if (e.Error == null)
    {
        //成功取得换回值
        UserLoginInfo[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetAllConnectinfo(System.AsyncCallback callback, object asyncState) 开始异步委托  
 EndGetAllConnectinfo(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.7 GetAllForbidConnectinfo

```
UserLoginInfo[] GetAllForbidConnectinfo();
```

说明：获取被禁止连接到分布式管理中心所有用户信息。

RETURN	DESCRIPTION
<b>UserLoginInfo</b>	用户登录信息。 IPAddress 登录客户端地址 Port 客户端连接端口 Stime 登录开始时间 AppName 承载客户端应用名称



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
UserLoginInfo[] returnValue = ddmClient.GetAllForbidConnectinfo();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetAllForbidConnectinfoCompleted);
//第三步：开始异步
ddmClient.GetAllForbidConnectinfoAsync();
//接受事件
void NetClient_GetAllForbidConnectinfoCompleted(object sender, Apace.Client.DDMService.
GetAllForbidConnectinfoCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        UserLoginInfo[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```





### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetAllForbidConnectinfo(System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndGetAllForbidConnectinfo(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.1.8 GetServiceAllDir

```
string[] GetServiceAllDir();
```

说明：得到当前服务器下所有的注册目录集合。

RETURN	DESCRIPTION
DirName	服务目录名称。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：调用

```
string[] returnValue = ddmClient.GetServiceAllDir();
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：注册事件

```
ddmClient.GetDDMConfigCompleted += new
```

```
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetServiceAllDirCompleted);
```

//第三步：开始异步

```
ddmClient.GetServiceAllDirAsync();
```

//接受事件

```
void NetClient_GetServiceAllDirCompleted(object sender, Apace.Client.DDMService.GetServiceAllDirCompletedEventArgs e) {
```

```

        if (e.Error == null)
        {
            //成功取得换回值
            string[] returnValue = e.Result;
        }
        else
        {
            //网络异常
        }
    }
}

```



#### 相关方法（委托）：

System.IAsyncResult BeginGetServiceAllDir(System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetServiceAllDir(System.IAsyncResult result) 结束异步委托

说明：

System.AsyncCallback callback 回调方法

### 3.1.9 ServiceState

```
Dictionary<string, ProcessRunInfo> ServiceState();;
```

说明：得到当前服务器,所有 Apace 服务组件运行状态。

RETURN	DESCRIPTION
string	服务组件名称 格式：[服务名].[目录名].[组件名]
ProcessRunInfo	Apace 服务进程运行信息。 ProcessName Apace 服务进程名称格式[服务名].[目录名].[进程名] RunState 服务进程当前运行状态中文描述（“停止”，“运行”） HasExited 服务进程是否已经退出(true, false) ErrorInfo 服务进程运行错误信息 ExitTime 服务进程退出时间戳 StartTime 服务进程开始时间戳 PagedMemorySize 进程占用的内存容量 PeakPagedMemorySize 虚拟内存占用的容量 PeakVirtualMemorySize 物理内存占用的容量 VirtualMemorySize 虚拟内存大小 PeakWorkingSet 物理内存最大容量 WorkingSet 物理内存 PrivateMemorySize 专用内存容量 PrivilegedProcessorTime 运行的时间 TotalProcessorTime 使用 CPU 时间 UserProcessorTime 进程代码执行时间 RunPath 进程运行路径



## 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：调用

```
Dictionary<string, ProcessRunInfo> returnValue = ddmClient.ServiceState();
```



## 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：注册事件

```
ddmClient.GetDDMConfigCompleted += new
```

```
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_ServiceStateCompleted);
```

//第三步：开始异步

```
ddmClient.ServiceStateAsync();
```

//接受事件

```
void NetClient_ServiceStateCompleted(object sender, Apace.Client.DDMService.ServiceStateCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Dictionary<string, ProcessRunInfo> returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



## 相关方法（委托）：

System.IAsyncResult BeginServiceState(System.AsyncCallback callback, object asyncState) 开始异步委托

EndServiceState(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.10 ServiceKeyState

```
ProcessRunInfo ServiceKeyState(string ServiceKey)
```

说明：得到当前服务器,指定服务组件进程运行状态。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apace.PCM [服务名].[目录名].[服务组件]

RETURN	DESCRIPTION
ProcessRunInfo	Apace 服务进程运行信息。



### 实例演示获取内存服务组件当前的运行状态（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
ProcessRunInfo returnValue = returnValue= ddmClient.ServiceKeyState("Apac.DDM.MCB");
```



### 实例演示获取内存服务组件当前的运行状态（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_ServiceKeyStateCompleted);
//第三步：开始异步
ddmClient.ServiceKeyStateAsync("Apac.DDM.MCB");
//接受事件
void NetClient_ServiceKeyStateCompleted(object sender, Apac.Client.DDMService.ServiceKeyStateCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        ProcessRunInfo returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```

}

**相关方法（委托）：**

System. [IAsyncResult](#) BeginServiceKeyState(string keyName, System. [AsyncCallback](#) callback, [object](#) asyncState)  
开始异步委托

EndServiceKeyState(System. [IAsyncResult](#) result) 结束异步委托

**Apace 服务组件列表：**

- [服务名].[目录名].PCM 点标签基本信息管理服务
- [服务名].[目录名].HIS 时序数据历史管理服务
- [服务名].[目录名].ALARM 告警检测服务
- [服务名].[目录名].MCB 实时数据管理服务（内存块控制服务）
- [服务名].[目录名].Syn 数据同步管理服务
- [服务名].[目录名].Algorithm 计算数据管理服务
- [服务名].[目录名].NetUdp 数据镜像管理服务（跨网闸数据服务）

**说明：** System. [AsyncCallback](#) callback 回调方法

### 3.1.11 StopService

```
StopService(string ServiceKey)
```

**说明：** 停止目标 Apace 实时数据库服务器上指定服务组件进程。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apace.PCM [服务名].[目录名].[服务组件]

**实例演示（同步）**

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
UserLoginInfo[] returnValue = returnValue= ddmClient.StopService("DDM.Apace.MCB");
```

**实例演示（异步）**

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_StopServiceCompleted);
```

```
//第三步：开始异步
ddmClient.StopServiceAsync("DDM.Apace.MCB");
//接受事件
void NetClient_StopServiceCompleted(object sender, Apace.Client.DDMService.StopServiceCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginStopService(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndStopService(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.12 RunService

```
RunService(string ServiceKey)
```

说明：运行目标 Apace 实时数据库服务器上指定服务组件进程。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apace.PCM [服务名].[目录名].[服务组件]



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
UserLoginInfo[] returnValue = ddmClient.RunService("DDM.Apace.MCB");
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
```

```

Apave.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apave.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_RunServiceCompleted);
//第三步：开始异步
ddmClient.RunServiceAsync("DDM. Apave. MCB");
//接受事件
void NetClient_RunServiceCompleted(object sender, Apave.Client.DDMService. RunServiceCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginRunService(string KeyName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndRunService(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.1.13 KillService

```
KillService(string ServiceKey)
```

说明： 强行终止目标 Apave 实时数据库服务器上指定服务组件进程。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如： DDM. Apave. PCM [服务名].[目录名].[服务组件]



### 实例演示（同步）

```

//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apave.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
UserLoginInfo[] returnValue = ddmClient.KillService("DDM. Apave. MCB");

```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_KillServiceCompleted);
//第三步：开始异步
ddmClient.KillServiceAsync("DDM. Apac. MCB");
//接受事件
void NetClient_ KillServiceCompleted(object sender, Apac.Client.DDMService. KillServiceCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginKillService(string KeyName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndKillService(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.1.14 GetLogInfo

```
Stream GetLogInfo()
```

说明：得到目标 Apace 实时数据库服务器上日志信息。

RETURN	DESCRIPTION
Stream	数据流, 需要反序列化该数据流后才能得到正确的日志信息, Apace 实时数据库日志文件完全符合 Window 系统日志格式的标准所以通过, List<System.Diagnostics.EventLogEntry> 类型可以对数据流进行数据的反序列化。





## 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
Stream returnValue = ddmClient.GetLogInfo();
BinaryFormatter serial = new BinaryFormatter();
List<System.Diagnostics.EventLogEntry> logObject = serial.Deserialize(returnValue) as
List<System.Diagnostics.EventLogEntry>;
```



## 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetLogInfoCompleted);
//第三步：开始异步
ddmClient.GetLogInfoAsync();
//接受事件
void NetClient_GetLogInfoCompleted(object sender, Apac.Client.DDMService.GetLogInfoCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
        List<System.Diagnostics.EventLogEntry> logObject
        = serial.Deserialize(returnValue) as List<System.Diagnostics.EventLogEntry>;
    }
    else
    {
        //网络异常
    }
}
```



## 相关方法（委托）：

System.IAsyncResult BeginGetLogInfo(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetLogInfo(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.1.15 ClearLog

```
void ClearLog()
```

说明：清除目标 Apace 实时数据库服务器上日志信息。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
ddmClient.ClearLog();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_ClearLogCompleted);
//第三步：开始异步
ddmClient.ClearLogAsync();
//接受事件
void NetClient_ClearLogCompleted(object sender, Apac.Client.DDMService. ClearLogCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```



#### 相关方法（委托）：

System.IAsyncResult BeginClearLog(System.AsyncCallback callback, object asyncState) 开始异步委托  
 EndClearLog(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.1.16 ReBootDDM

```
void ReBootDDM()
```

说明：重新启动目标 Apace 实时数据库服务器 DDM 控制服务。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
ddmClient.ReBootDDM();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_ReBootDDMCompleted);
//第三步：开始异步
ddmClient.ReBootDDMAsync();
//接受事件
void NetClient_ReBootDDMCompleted(object sender, Apace.Client.DDMService.ReBootDDMCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```



#### 相关方法（委托）：

System.IAsyncResult BeginReBootDDM(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndReBootDDM(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.1.17 GetLocalHostKey

```
string GetLocalHostKey();
```

说明：获得目标 Apace 实时数据库服务器注册所需要的密钥。

RETURN	DESCRIPTION
string	用 32 位 MD5 码值来表示服务器唯一标示, 该 Key 值主要用于申请授权证书时, 将 Key 发送给 Apace 客户服务人员, 相关人员会根据用户申请信息反馈给用户相应的证书文件。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
string returnValue = ddmClient.GetLocalHostKey();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetLocalHostKeyCompleted);
//第三步：开始异步
ddmClient.GetLocalHostKeyAsync();
//接受事件
void NetClient_GetLocalHostKeyCompleted(object sender, Apace.Client.DDMService.GetLocalHostKeyCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        string returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetLocalHostKey(System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
EndGetLocalHostKey(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.1.18 RegisterLocalHostlicence

```
bool RegisterLocalHostlicence(byte[] LicenceBody);
```

说明：将证书文件导入到目标 Apace 实时数据库服务器完成版本授权。

RETURN	DESCRIPTION
<a href="#">bool</a>	注册证书文件成功状态，失败代表证书文件验证错误。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
FileStream stream = new FileStream(this.textBoxFilePath.Text,
                                   System.IO.FileMode.Open,
                                   System.IO.FileAccess.Read,
                                   System.IO.FileShare.None, 1024, false);

byte[] filebody = new byte[stream.Length];
int readcount=stream.Read(filebody, 0, filebody.Length);
Bool returnValue = ddmClient.RegisterLocalHostlicence(filebody);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>
(NetClient_RegisterLocalHostlicenceCompleted);
//第三步：开始异步
FileStream stream = new FileStream(this.textBoxFilePath.Text,
                                   System.IO.FileMode.Open,
                                   System.IO.FileAccess.Read,
```

```

        System.IO.FileShare.None, 1024, false);

        byte[] filebody = new byte[stream.Length];
        int readcount=stream.Read(filebody, 0, filebody.Length);
        ddmClient.RegisterLocalHostlicenceAsync(filebody);
        //接受事件
        void NetClient_ RegisterLocalHostlicenceCompleted(object sender,
        Apace.Client.DDMService.RegisterLocalHostlicenceCompletedEventArgs e)
        {
            if (e.Error == null)
            {
                //成功取得换回值
                Bool returnValue = e.Result;
            }
            else
            {
                //网络异常
            }
        }
    }

```



### 相关方法（委托）：

System.IAsyncResult BeginRegisterLocalHostlicence(System.AsyncCallback callback, object asyncState) 开始异步委托

EndRegisterLocalHostlicence(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.19 GetOverdueTime

```
DateTime GetOverdueTime();
```

说明：获得目标服务器 Apace 实时数据库版本的到期时间。

RETURN	DESCRIPTION
DateTime	用于表示目标数据库版本过期的时间。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：获得 DDM 客户端实体

```
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
```

//第三步：调用

```
DateTime returnValue = ddmClient.GetOverdueTime();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apac.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apac.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetOverdueTimeCompleted);
//第三步：开始异步
ddmClient.GetOverdueTimeAsync();
//接受事件
void NetClient_GetOverdueTimeCompleted(object sender, Apac.Client.DDMService. GetOverdueTimeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        DateTime returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetOverdueTime(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetOverdueTime(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.20 GetIsRegister

```
bool GetIsRegister();
```

说明：获得目标服务器 Apace 实时数据库版本的过期时间。

RETURN	DESCRIPTION
bool	True 已经授权, False 没有授权



### 实例演示（同步）

```
//第一步：得到客户端实体
```

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
bool returnValue = ddmClient.GetIsRegister();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetIsRegisterCompleted);
//第三步：开始异步
ddmClient.GetIsRegisterAsync();
//接受事件
void NetClient_GetIsRegisterCompleted(object sender, Apace.Client.DDMService.GetIsRegisterCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetIsRegister(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetIsRegister(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.1.21 GetServiceVersion

```
string GetServiceVersion();
```

说明：获得目标服务器 Apace 实时数据库版本的名称。

RETURN	DESCRIPTION
--------	-------------



string

用于描述目标数据库版本文本信息。



## 实例演示（同步）

//第一步：得到客户端实体

Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();

//第二步：获得 DDM 客户端实体

Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")

//第三步：调用

string returnValue = ddmClient.GetServiceVersion();



## 实例演示（异步）

//第一步：得到客户端实体

Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();

//第二步：获得 DDM 客户端实体

Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")

//第三步：注册事件

ddmClient.GetDDMConfigCompleted += new

EventHandler&lt;Apace.Client.DDMService.GetDDMConfigCompletedEventArgs&gt;(NetClient\_GetServiceVersionCompleted);

//第三步：开始异步

ddmClient.GetServiceVersionAsync();

//接受事件

```
void NetClient_GetServiceVersionCompleted(object sender, Apace.Client.DDMService.GetServiceVersionCompletedEventArgs e) {
```

```
    if (e.Error == null)
```

```
    {
```

```
        //成功取得返回值
```

```
        string returnValue = e.Result;
```

```
    }
```

```
    else
```

```
    {
```

```
        //网络异常
```

```
    }
```

```
}
```



## 相关方法（委托）：

System.IAsyncResult BeginGetServiceVersion(System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetServiceVersion(System.IAsyncResult result) 结束异步委托

说明：

System.AsyncCallback callback 回调方法

## 3.1.22 GetAuthorizationMaxPointCount

```
int GetAuthorizationMaxPointCount();
```

说明：获得目标服务器 Apace 实时数据库版本授权最大点标签总数。

RETURN	DESCRIPTION
int	授权的最大点标签值。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：调用
int returnValue = ddmClient.GetAuthorizationMaxPointCount();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：获得 DDM 客户端实体
Apace.Client.DDMService.DDMContractClient ddmClient= client.GetSonDDMClient("DDM")
//第三步：注册事件
ddmClient.GetDDMConfigCompleted += new
EventHandler<Apace.Client.DDMService.GetDDMConfigCompletedEventArgs>(NetClient_GetAuthorizationMaxPointCountComple
ted);
//第三步：开始异步
ddmClient.GetAuthorizationMaxPointCountAsync();
//接受事件
void NetClient_GetAuthorizationMaxPointCountCompleted(object sender, Apace.Client.DDMService.
GetAuthorizationMaxPointCountCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        int returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetAuthorizationMaxPointCount(System. [AsyncCallback](#) callback, [object](#) asyncState)  
开始异步委托  
EndGetAuthorizationMaxPointCount(System. [IAsyncResult](#) result) 结束异步委托

说明: System. [AsyncCallback](#) callback 回调方法

## 3.2 PCM 服务(Apace 点标签信息管理)



### 3.2.1 AddPointInfo

```
bool AddPointInfo(PcmData Point, AddOptions AddOptions);
```

说明: 添加一个点标签信息到点配置区域

PARAMETER	DESCRIPTION
Point	<a href="#">PcmData</a> 点标签基本信息
AddOptions	添加点标签基本信息的方式。 NewOrUpdate 新建或更新 None 存在不替换

RETURN	DESCRIPTION
<a href="#">bool</a>	True 成功, False 失败。

### 3.2.2 AddPointInfoStream

```
void AddPointInfoStream(Stream Stream);
```

说明: 添加点标签信息流到点配置区域

PARAMETER	DESCRIPTION
<a href="#">Stream</a>	点标签基本信息格式化数据流 说明: Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式, 大部分的数据契约都有其自身的格式化和反格式方式。

### 3.2.3 DeletePointInfo

```
bool DeletePointInfo(string PointName);
```

说明：从点配置区域删除一个点标签

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apacetest1 [服务名].[目录名].[点名]

RETURN	DESCRIPTION
bool	True 删除点标签成功，False 删除点标签失败。

### 3.2.4 DeletePointInfos

```
void DeletePointInfos(string[] Points);
```

说明：删除一批点标签信息。

PARAMETER	DESCRIPTION
Points	需要删除的点名集合

### 3.2.5 UpdatePointInfo

```
bool UpdatePointInfo(PcmData PointInfo);
```

说明：更新一个点标签信息到点配置区域。

PARAMETER	DESCRIPTION
PointInfo	需要更新点标签的基本信息，PcmData 数据格式请参见数据契约

RETURN	DESCRIPTION
bool	True 更新点标签信息成功，False 更新点标签信息失败。

### 3.2.6 GetAllPointCollocate

```
PcmData[] GetAllPointCollocate(string ServiceName, string DirName);
```

说明：得到指定服务目录下所有点标签信息。

PARAMETER	DESCRIPTION
-----------	-------------

ServiceName	服务名
DirName	目录名

RETURN	DESCRIPTION
PcmData[]	点标签信息集合体, PcmData 数据实体请参见数据契约。

### 3.2.7 GetPointInfo

```
PcmData GetPointInfo(string PointName);
```

说明：得到当前服务器,指定服务组件进程运行状态。得到指定点的详细信息

PARAMETER	DESCRIPTION
PointName	点名称, 如: DDM.Apave.Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
PcmData	点标签信息, PcmData 数据实体请参见数据契约。

### 3.2.8 GetAllPointStream

```
Stream GetAllPointStream(string ServiceName, string DirName);
```

说明：得到指定服务目录下所有点标签数据流信息。

PARAMETER	DESCRIPTION
ServiceName	服务名
DirName	目录名

RETURN	DESCRIPTION
Stream	点标签基本信息格式化数据流 说明: Apave 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式, 大部分的数据契约都有其自身的格式化和反格式方式。

## 3.3 实时服务(Apace 高速缓存区管理)



### 3.3.1 GetAllPoint

```
McbData[] GetAllPoint();
```

说明：得到当前缓存区中所有实时数据点标签。

RETURN	DESCRIPTION
McbData[]	实时点标签信息集合体，McBData 数据格式请参见数据契约。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcClient(Servicename, Apac)
//第四步：调用
McBData[] returnValue = mcbClient.GetAllPoint();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcClient(Servicename, Apac)
//第三步：注册事件
mcbClient.GetAllPointCompleted += new EventHandler< Apac.Client.McbService.GetAllPointCompletedEventArgs>
(NetClient_GetAllPointCompleted);
//第三步：开始异步
mcbClient.GetAllPointAsync();
//接受事件
void NetClient_ GetAllPointCompleted(object sender, Apac.Client.McbService.GetAllPointCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McBData[] returnValue = e.Result;
    }
    else
```

```
{
    //网络异常
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetAllPoint(System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
EndGetAllPoint(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.2 GetAllPointStream

```
Stream GetAllPointStream();
```

说明：得到当前缓存区中数据流。

RETURN	DESCRIPTION
<a href="#">Stream</a>	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apace)
//第四步：调用
Stream returnValue = mcbClient.GetAllPointStream();
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apace)
//第三步：注册事件
mcbClient.GetAllPointStreamCompleted += new EventHandler<
Apace.Client.McbService.GetAllPointStreamCompletedEventArgs>
(NetClient_GetAllPointStreamCompleted);
```

```
//第三步：开始异步
mcbClient.GetAllPointStreamAsync();
//接受事件
void NetClient_GetAllPointStreamCompleted(object sender, Apac.Client.McbService.GetAllPointStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



#### 相关方法（委托）：

System.IAsyncResult BeginGetAllPointStream(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetAllPointStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.3 GetPoint

```
McbData GetPoint(string PointName);
```

说明：得到缓冲区中指定点标签的实时数据。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apac.Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
McbData	实时点标签信息，McData 数据格式请参见数据契约。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apac.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname); Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
```



//第四步：调用

```
McbData returnValue = mcbClient.GetPoint(Pointname);
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第三步：注册事件

```
mcbClient.GetPointCompleted += new EventHandler< Apace.Client.McbService.GetPointCompletedEventArgs>
(NetClient_GetPointCompleted);
```

//第三步：开始异步

```
mcbClient.GetPointAsync(Pointname);
```

//接受事件

```
void NetClient_GetPointCompleted(object sender, Apace.Client.McbService.GetPointCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetPoint(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetPoint(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.4 GetPointsStream

```
Stream GetPointsStream(string[] Points);
```

说明：得到缓冲区中指定批量点标签实时数据流。

PARAMETER	DESCRIPTION
Points	点名集合。

RETURN	DESCRIPTION
<code>Stream</code>	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apace.test";
Points [1]="DDM.Apace.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
//第四步：调用
Stream returnValue = mcbClient.GetPointsStream(Points);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apace.test";
Points [1]="DDM.Apace.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
//第三步：注册事件
mcbClient.GetPointsStreamCompleted += new EventHandler<
Apace.Client.McbService.GetPointsStreamCompletedEventArgs>
(NetClient_GetPointsStreamCompleted);
//第三步：开始异步
mcbClient.GetPointsStreamAsync(Points);
//接受事件
void NetClient_ GetPointsStreamCompleted(object sender, Apace.Client.McbService.GetPointsStreamCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
    }
}
```

```

        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetPointsStream(String[] Points, System. [AsyncCallback](#) callback, [object](#) asyncState)

开始异步委托

EndGetPointsStream(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.5 GetBufferData

```
HisRowData[] GetBufferData(string PointName);
```

说明：得到指定点标签最近缓冲区中存放的时序历史数据。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM. Apace. Test [服务名]. [目录名]. [点名]

RETURN	DESCRIPTION
<a href="#">HisRowData</a> []	时序历史数据集合体， <a href="#">HisRowData</a> 数据格式请参见数据契约



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM. Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule (Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
```

//第四步：调用

```
HisRowData [] returnValue = mcbClient.GetBufferData (Pointname);
```



### 实例演示（异步）

//第一步：得到客户端实体

```

Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apace.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.GetBufferDataCompleted += new EventHandler< Apace.Client.McbService.GetBufferDataCompletedEventArgs>
(NetClient_GetBufferDataCompleted);
//第三步：开始异步
mcbClient.GetBufferDataAsync(Pointname);
//接受事件
void NetClient_GetBufferDataCompleted(object sender, Apace.Client.McbService.GetBufferDataCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        HisRowData [] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



#### 相关方法（委托）：

System.IAsyncResult BeginGetBufferData(String PointName, System.AsyncCallback callback, object asyncState)  
 开始异步委托  
 EndGetBufferData(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.6 GetBufferDataStream

```
Stream GetBufferDataStream(string PointName);
```

说明：得到指定点标签最近缓冲区中存放的时序历史数据流。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
Stream	点标签历史信息格式化数据流

说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



## 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
Stream returnValue = mcbClient.GetBufferDataStream(Pointname);
```



## 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第三步：注册事件

```
mcbClient.GetBufferDataStreamCompleted += new EventHandler<Apace.Client.McbService.GetBufferDataStreamCompletedEventArgs>(NetClient_GetBufferDataStreamCompleted);
```

//第三步：开始异步

```
mcbClient.GetBufferDataStreamAsync(Pointname);
```

//接受事件

```
void NetClient_GetBufferDataStreamCompleted(object sender, Apace.Client.McbService.GetBufferDataStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



## 相关方法（委托）：

System. `IAsyncResult` BeginGetBufferDataStream(String PointName, System. `AsyncCallback` callback, `object` asyncState) 开始异步委托  
EndGetBufferDataStream(System. `IAsyncResult` result) 结束异步委托

说明： System. `AsyncCallback` callback 回调方法

### 3.3.7 SelectPointName

```
McbData[] SelectPointName(string value);
```

说明： 查找所有匹配点名称实时数据集合。

PARAMETER	DESCRIPTION
value	用来批准点标签名称条件值 说明： 匹配的方式为包含, 例 Value "test" 满足条件的有如下形式: 1: *test* 2: test* 3: *test

RETURN	DESCRIPTION
<code>McbData[]</code>	实时数据集合体, <code>McbData</code> 数据格式请参见数据契约。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
McData[] returnValue = mcbClient.SelectPointName(slevalue);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
```

```
//第三步：注册事件
mcbClient.SelectPointNameCompleted += new EventHandler<
Apave.Client.McbService.SelectPointNameCompletedEventArgs>
(NetClient_SelectPointNameCompleted);
//第三步：开始异步
mcbClient.SelectPointNameAsync(slevalue);
//接受事件
void NetClient_SelectPointNameCompleted(object sender, Apave.Client.McbService.SelectPointNameCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectPointName(String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState)

开始异步委托

EndSelectPointName(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.8 SelectPointNameStream

```
Stream SelectPointNameStream(string value);
```

说明： 查找所有匹配点名称实时数据流。

PARAMETER	DESCRIPTION
value	用来批准点标签名称条件值 说明： 匹配的方式为包含, 例 Value "test" 满足条件的有如下形式： 1: *test* 2: test* 3: *test

RETURN	DESCRIPTION
<a href="#">Stream</a>	点标签实时信息格式化数据流

说明：Apave 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apave";
//第三步：获得实时服务客户端访问实体
Apave.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectPointNameStream(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apave";
//第三步：获得实时服务客户端访问实体
Apave.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointNameStreamCompleted += new EventHandler<
Apave.Client.McbService.SelectPointNameStreamCompletedEventArgs>
(NetClient_SelectPointNameStreamCompleted);
//第三步：开始异步
mcbClient.SelectPointNameStreamAsync(slevalue);
//接受事件
void NetClient_SelectPointNameStreamCompleted(object sender,
Apave.Client.McbService.SelectPointNameStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectPointNameStream(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托



EndSelectPointNameStream(System. [IAsyncResult](#) result) 结束异步委托

说明: System. [AsyncCallback](#) callback 回调方法

### 3.3.9 SelectPointDescribe

```
McbData[] SelectPointDescribe(string value);
```

说明: 查找所有匹配点描述实时数据集。

PARAMETER	DESCRIPTION
value	用来批准点标签描述的条件值 说明: 匹配的方式为包含, 例 Value "test" 满足条件的有如下形式: <ol style="list-style-type: none"> <li>1: *test*</li> <li>2: test*</li> <li>3: *test</li> </ol>

RETURN	DESCRIPTION
<a href="#">McData</a> []	实时数据集集合, <a href="#">McData</a> 数据格式请参见数据契约。



#### 实例演示 (同步)

```
//第一步: 得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步: 准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apac";
//第三步: 获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步: 调用
McData[] returnValue = mcbClient. SelectPointDescribe (slevalue);
```



#### 实例演示 (异步)

```
//第一步: 得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步: 准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apac";
//第三步: 获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步: 注册事件
mcbClient.SelectPointDescribeCompleted += new EventHandler<
Apac.Client.McbService.SelectPointDescribeCompletedEventArgs>
(NetClient_SelectPointDescribeCompleted);
```

```
//第三步：开始异步
mcbClient.SelectPointDescribeAsync(slevalue);
//接受事件
void NetClient_ SelectPointDescribeCompleted(object sender,
Apace.Client.McbService.SelectPointDescribeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectPointDescribe(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectPointDescribe(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.10 GetHisServicePath

```
string GetHisServicePath();
```

说明：获取实时服务所对应的历史服务网络路径。

RETURN	DESCRIPTION
string	<p>Apace 服务组件的网络路径。</p> <p>如：DDM.Apace.HIS。</p> <p>格式：[服务名].[目录名].[组件名称]。</p> <p>说明：每一个实时服务必须对应一个历史服务，对应历史服务的目标可以是分布式网络中的任意 Apace 服务器，当然也可以是本地历史服务，对于实时服务和历史服务只能是一对一关系，当历史服务与实时服务可以存在一对多的关系，也可以理解为一个历史服务可以为多个实时服务承载时序的历史数据。</p>



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apace)
```

//第四步：调用

```
string returnValue = mcbClient.GetHisServicePath();
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好网络路径

```
String Servicename="DDM"; String Dirname="Apace";
```

//第三步：获得实时服务客户端访问实体

```
Apace.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apace)
```

//第三步：注册事件

```
mcbClient.GetHisServicePathCompleted += new EventHandler<
```

```
Apace.Client.McbService.GetHisServicePathCompletedEventArgs>
```

```
(NetClient_GetHisServicePathCompleted);
```

//第三步：开始异步

```
mcbClient.GetHisServicePathAsync();
```

//接受事件

```
void NetClient_GetHisServicePathCompleted(object sender, Apace.Client.McbService.GetHisServicePathCompletedEventArgs e) {
```

```
    if (e.Error == null)
```

```
    {
```

```
        //成功取得换回值
```

```
        string returnValue = e.Result;
```

```
    }
```

```
    else
```

```
    {
```

```
        //网络异常
```

```
    }
```

```
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetHisServicePath(System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetHisServicePath(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.11 Contains

```
bool Contains(string PointName);
```

说明： 判断高速缓冲区中是否存在指定的点标签。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
bool	True 存在, False 不存在。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
bool returnValue = mcbClient.Contains(Pointname);
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第三步：注册事件

```
mcbClient.ContainsCompleted += new EventHandler< Apace.Client.McbService.ContainsCompletedEventArgs>
(NetClient_ContainsCompleted);
```

//第三步：开始异步

```
mcbClient.ContainsAsync(Pointname);
```

//接受事件

```
void NetClient_ContainsCompleted(object sender, Apace.Client.McbService.ContainsCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. `IAsyncResult` BeginContains(String PointName, System. `AsyncCallback` callback, `object` asyncState) 开始异步委托

EndContains(System. `IAsyncResult` result) 结束异步委托

说明: System. `AsyncCallback` callback 回调方法

### 3.3.12 SelectPointDescribeStream

```
Stream SelectPointDescribeStream(string value);
```

说明: 查找所有匹配点描述实时数据流。

PARAMETER	DESCRIPTION
value	用来批准点标签描述的条件值 说明: 匹配的方式为包含, 例 Value "test" 满足条件的有如下形式: 1: *test* 2: test* 3: *test

RETURN	DESCRIPTION
<code>Stream</code>	点标签实时信息格式化数据流 说明: Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式, 大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示 (同步)

```
//第一步: 得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步: 准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apace";
//第三步: 获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步: 调用
Stream returnValue = mcbClient.SelectPointDescribeStream(slevalue);
```



#### 实例演示 (异步)

```
//第一步: 得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步: 准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apace";
```

```
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointDescribeStreamCompleted += new EventHandler<
Apac.Client.McbService.SelectPointDescribeStreamCompletedEventArgs>
    (NetClient_SelectPointDescribeStreamCompleted);
//第三步：开始异步
mcbClient.SelectPointDescribeStreamAsync(slevalue);
//接受事件
void NetClient_SelectPointDescribeStreamCompleted(object sender,
Apac.Client.McbService.SelectPointDescribeStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



#### 相关方法（委托）：

System.IAsyncResult BeginSelectPointDescribeStream(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectPointDescribeStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.13 SelectPointState

```
McbData[] SelectPointState(byte value);
```

说明： 查找所有匹配点状态实时数据集。

PARAMETER	DESCRIPTION		
value	用来批准点标签描述的条件值 说明： 状态值对应如下，		
	状态	值	描述
	NoInit	0	没有初始化
	OK	1	正常
	Lapse	2	失效
	HighAlarm	3	发生高报
	BosomAlarm	5	发生中报

	LowAlarm	6	发生低报
	Exceed	7	超过值的界限范围

RETURN	DESCRIPTION
McbData[]	实时数据集合体, McbData 数据格式请参见数据契约。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
byte slevalue=1; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
McbData[] returnValue = mcbClient.SelectPointState(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointStateCompleted += new EventHandler<
Apac.Client.McbService.SelectPointStateCompletedEventArgs>
(NetClient_SelectPointStateCompleted);
//第三步：开始异步
mcbClient.SelectPointStateAsync(slevalue);
//接受事件
void NetClient_SelectPointStateCompleted(object sender, Apac.Client.McbService.SelectPointStateCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {

```

//网络异常

}

}

**相关方法（委托）：**

System. [IAsyncResult](#) BeginSelectPointState(String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndSelectPointState(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

**3.3.14 SelectPointStateStream**

```
Stream SelectPointStateStream(byte value);
```

说明： 查找所有匹配点状态实时数据流。

PARAMETER	DESCRIPTION		
value	用来批准点标签描述的条件值 说明：状态值对应如下，		
	状态	值	描述
	NoInit	0	没有初始化
	OK	1	正常
	Lapse	2	失效
	HighAlarm	3	发生高报
	BosomAlarm	5	发生中报
	LowAlarm	6	发生低报
	Exceed	7	超过值的界限范围

RETURN	DESCRIPTION
<a href="#">Stream</a>	<p>点标签实时信息格式化数据流 说明： Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。</p>

**实例演示（同步）**

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好查询条件和网络名

```
byte svalue=1; String Servicename="DDM"; String Dirname="Apace";
```

//第三步：获得实时服务客户端访问实体



```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
```

```
//第四步：调用
```

```
Stream returnValue = mcbClient.SelectPointStateStream(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
```

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
//第二步：准备好查询条件和网络名
```

```
String slevalue=" test"; String Servicename="DDM"; String Dirname="Apace";
```

```
//第三步：获得实时服务客户端访问实体
```

```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
```

```
//第三步：注册事件
```

```
mcbClient.SelectPointStateStreamCompleted += new EventHandler<
```

```
Apace.Client.McbService.SelectPointStateStreamCompletedEventArgs>
```

```
(NetClient_SelectPointStateStreamCompleted);
```

```
//第三步：开始异步
```

```
mcbClient.SelectPointStateStreamAsync(slevalue);
```

```
//接受事件
```

```
void NetClient_SelectPointStateStreamCompleted(object sender,
```

```
Apace.Client.McbService.SelectPointStateStreamCompletedEventArgs e) {
```

```
    if (e.Error == null)
```

```
    {
```

```
        //成功取得换回值
```

```
        Stream returnValue = e.Result;
```

```
    }
```

```
    else
```

```
    {
```

```
        //网络异常
```

```
    }
```

```
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectPointStateStream(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectPointStateStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.15 SelectPointNameRegex

```
McbData[] SelectPointNameRegex(string regex);
```

说明：通过正则表达式查找所有匹配点名称实时数据集合。

PARAMETER	DESCRIPTION
-----------	-------------

regex	用来匹配点名称的正则表达式。
-------	----------------

RETURN	DESCRIPTION
McbData[]	实时数据集合体，McData 数据格式请参见数据契约。



## 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
McData[] returnValue = mcbClient.SelectPointNameRegex(slevalue);
```



## 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointNameRegexCompleted += new EventHandler<
Apac.Client.McbService.SelectPointNameRegexCompletedEventArgs>
(NetClient_SelectPointNameRegexCompleted);
//第三步：开始异步
mcbClient.SelectPointNameRegexAsync(slevalue);
//接受事件
void NetClient_ SelectPointNameRegexCompleted(object sender,
Apac.Client.McbService.SelectPointNameRegexCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectPointNameRegex (String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndSelectPointNameRegex (System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

### 3.3.16 SelectPointNameRegexStream

```
Stream SelectPointNameRegexStream(string regex);
```

说明：通过正则表达式查找所有匹配点名称实时数据流。

PARAMETER	DESCRIPTION
regex	用来匹配点名称的正则表达式。

RETURN	DESCRIPTION
<a href="#">Stream</a>	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectPointNameRegexStream(slevalue);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointNameRegexStreamCompleted += new EventHandler<
Apace.Client.McbService.SelectPointNameRegexStreamCompletedEventArgs>
```

```

        (NetClient_SelectPointNameRegexStreamCompleted);
        //第三步：开始异步
        mcbClient.SelectPointNameRegexStreamAsync(slevalue);
        //接受事件
        void NetClient_SelectPointNameRegexStreamCompleted(object sender,
        Apace.Client.McbService.SelectPointNameRegexStreamCompletedEventArgs e) {
            if (e.Error == null)
            {
                //成功取得换回值
                Stream returnValue = e.Result;
            }
            else
            {
                //网络异常
            }
        }
    }

```



### 相关方法（委托）：

System.IAsyncResult BeginSelectPointNameRegexStream(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectPointNameRegexStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.17 SelectOperationType

```
McbData[] SelectOperationType(string value);
```

说明：查找所有点标签中匹配签业务类型的实时数据集合。

PARAMETER	DESCRIPTION
value	用户过滤业务类型值 说明：业务类型为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务类型来划分点的业务范围，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。

RETURN	DESCRIPTION
McbData[]	实时数据集合体，McBData 数据格式请参见数据契约。



### 实例演示（同步）

//第一步：得到客户端实体

```

Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue="00001"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
McbData[] returnValue = mcbClient.SelectOperationType(slevalue);

```



### 实例演示（异步）

```

//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue="00001"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationTypeCompleted += new EventHandler<
Apac.Client.McbService.SelectOperationTypeCompletedEventArgs>
    (NetClient_SelectOperationTypeCompleted);
//第三步：开始异步
mcbClient.SelectOperationTypeAsync(slevalue);
//接受事件
void NetClient_SelectOperationTypeCompleted(object sender,
Apac.Client.McbService.SelectOperationTypeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginSelectOperationType(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectOperationType(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.18 SelectOperationTypeStream

```
Stream SelectOperationTypeStream(string value);
```

说明：查找所有点标签中匹配签业务类型的实时数据流。

PARAMETER	DESCRIPTION
value	<p>用户过滤业务类型值</p> <p>说明：业务类型为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务类型来划分点的业务范围，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。</p>

RETURN	DESCRIPTION
Stream	<p>点标签实时信息格式化数据流</p> <p>说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。</p>



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" 空气湿度"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectOperationTypeStream(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" 空气湿度"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationTypeStreamCompleted += new EventHandler<
Apac.Client.McbService.SelectOperationTypeStreamCompletedEventArgs>
(NetClient_SelectOperationTypeStreamCompleted);
//第三步：开始异步
mcbClient.SelectOperationTypeStreamAsync(slevalue);
//接受事件
void NetClient_SelectOperationTypeStreamCompleted(object sender,
Apac.Client.McbService.SelectOperationTypeStreamCompletedEventArgs e) {
```

```

if (e.Error == null)
{
    //成功取得换回值
    Stream returnValue = e.Result;
}
else
{
    //网络异常
}
}

```



#### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectOperationTypeStream(String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndSelectOperationTypeStream(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

### 3.3.19 SelectOperationCode

```
McbData[] SelectOperationCode(string value);
```

说明： 查找所有点标签中匹配签业务代码的实时数据集合并。

PARAMETER	DESCRIPTION
value	<p>用户过滤业务代码值</p> <p>说明：业务代码为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务代码或业务标示码来标记采集点标签，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。</p>

RETURN	DESCRIPTION
<a href="#">McbData</a> []	实时数据集合并， <a href="#">McbData</a> 数据格式请参见数据契约。



#### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好查询条件和网络名

```
String slevalue="00001"; String Servicename="DDM"; String Dirname="Apace";
```

//第三步：获得实时服务客户端访问实体

```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
```

//第四步：调用

```
McbData[] returnValue = mcbClient.SelectOperationCode(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" 00001"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationCodeCompleted += new EventHandler<
Apac.Client.McbService.SelectOperationCodeCompletedEventArgs>
(NetClient_SelectOperationCodeCompleted);
//第三步：开始异步
mcbClient.SelectOperationCodeAsync(slevalue);
//接受事件
void NetClient_SelectOperationCodeCompleted(object sender,
Apac.Client.McbService.SelectOperationCodeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectOperationCode(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectOperationCode(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.20 SelectOperationCodeStream

```
Stream SelectOperationCodeStream(string value);
```

说明：查找所有点标签中匹配签业务代码的实时数据流。

PARAMETER	DESCRIPTION
value	<p>用户过滤业务代码值</p> <p>说明：业务代码为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用</p>



	系统所需要使用的业务代码或业务标示码来标记采集点标签，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。
--	--

RETURN	DESCRIPTION
<code>Stream</code>	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue="00001"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectOperationCodeStream(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue="00001"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationCodeStreamCompleted += new EventHandler<
Apace.Client.McbService.SelectOperationCodeStreamCompletedEventArgs>
(NetClient_SelectOperationCodeStreamCompleted);
//第三步：开始异步
mcbClient.SelectOperationCodeStreamAsync(slevalue);
//接受事件
void NetClient_SelectOperationCodeStreamCompleted(object sender,
Apace.Client.McbService.SelectOperationCodeStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
```

```
{
    //网络异常
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectOperationCodeStream(String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndSelectOperationCodeStream(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.21 SelectOperationCodeAndType

```
McbData[] SelectOperationCodeAndType(string codevalue, string typevalue);
```

说明： 查找所有点标签中匹配签业务代码和业务类型的实时数据集合并。

PARAMETER	DESCRIPTION
codevalue	用户过滤业务代码值 说明：业务代码为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务代码或业务标示码来标记采集点标签，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。
typevalue	用户过滤业务类型值 说明：业务类型为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务类型来划分点的业务范围，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。

RETURN	DESCRIPTION
<a href="#">McbData</a> []	实时数据集合并， <a href="#">McbData</a> 数据格式请参见数据契约。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好查询条件和网络名

```
String slevaluecode="00001"; String slevaluetype="空气湿度"
```

```
String Servicename="DDM"; String Dirname="Apace";
```

//第三步：获得实时服务客户端访问实体

```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
```

//第四步：调用

```
McbData[] returnValue = mcbClient.SelectOperationCodeAndType(slevaluecode, slevaluetype);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevaluecode="00001"; String slevaluetype="空气湿度"
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationCodeAndTypeCompleted += new EventHandler<
Apac.Client.McbService.SelectOperationCodeAndTypeCompletedEventArgs>
    (NetClient_SelectOperationCodeAndTypeCompleted);
//第三步：开始异步
mcbClient.SelectOperationCodeAndTypeAsync(slevaluecode, slevaluetype);
//接受事件
void NetClient_SelectOperationCodeAndTypeCompleted(object sender,
Apac.Client.McbService.SelectOperationCodeAndTypeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectOperationCodeAndType(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectOperationCodeAndType(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.22 SelectOperationCodeAndTypeStream

```
Stream SelectOperationCodeAndTypeStream(string codevalue, string typevalue);
```

说明： 查找所有点标签中匹配签业务代码和业务类型的实时数据流。

PARAMETER	DESCRIPTION
codevalue	用户过滤业务代码值

	说明：业务代码为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务代码或业务标示码来标记采集点标签，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。
typevalue	用户过滤业务类型值 说明：业务类型为 Apace 实时数据库中为采集点标签加入的扩展属性，最大能存放 20 个字节的数据，该属性的加入能更好的为上乘应用提供支持，应用可通过自身的需求为点标签存入应用系统所需要使用的业务类型来划分点的业务范围，从而形成一种 Apace 数据点标签与上层应用系统的一种映射关系。

RETURN	DESCRIPTION
Stream	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevaluecode="00001"; String slevaluetype="空气湿度"
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectOperationCodeAndTypeStream(slevaluecode, slevaluetype);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevaluecode="00001"; String slevaluetype="空气湿度"
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectOperationCodeAndTypeStreamCompleted += new EventHandler<
Apace.Client.McbService.SelectOperationCodeAndTypeStreamCompletedEventArgs>
(NetClient_SelectOperationCodeAndTypeStreamCompleted);
//第三步：开始异步
mcbClient.SelectOperationCodeAndTypeStreamAsync(slevaluecode, slevaluetype);
//接受事件
```

```
void NetClient_ SelectOperationCodeAndTypeStreamCompleted(object sender,
Apave.Client.McbService. SelectOperationCodeAndTypeStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectOperationCodeAndTypeStream(String PointName, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
EndSelectOperationCodeAndTypeStream(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.23 SelectPointDescribeRegex

```
McbData[] SelectPointDescribeRegex(string regex);
```

说明： 通过正则表达式查找所有匹配点描述的实时数据集合。

PARAMETER	DESCRIPTION
regex	用来匹配点描述的正则表达式。

RETURN	DESCRIPTION
<a href="#">McData</a> []	实时数据集合体， <a href="#">McData</a> 数据格式请参见数据契约。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apave";
//第三步：获得实时服务客户端访问实体
Apave.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
McData[] returnValue = mcbClient.SelectPointDescribeRegex(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointDescribeRegexCompleted += new EventHandler<
Apac.Client.McbService.SelectPointDescribeRegexCompletedEventArgs>
(NetClient_SelectPointDescribeRegexCompleted);
//第三步：开始异步
mcbClient.SelectPointDescribeRegexAsync(slevalue);
//接受事件
void NetClient_SelectPointDescribeRegexCompleted(object sender,
Apac.Client.McbService.SelectPointDescribeRegexCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginSelectPointDescribeRegex(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndSelectPointDescribeRegex(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.24 SelectPointDescribeRegexStream

```
Stream SelectPointDescribeRegexStream(string regex);
```

说明：通过正则表达式查找所有匹配点描述的实时数据流。

PARAMETER	DESCRIPTION
regex	用来匹配点描述的正则表达式。

RETURN	DESCRIPTION
Stream	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第四步：调用
Stream returnValue = mcbClient.SelectPointDescribeRegexStream(slevalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String slevalue=" test?"; String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Dirname)
//第三步：注册事件
mcbClient.SelectPointDescribeRegexStreamCompleted += new EventHandler<
Apace.Client.McbService.SelectPointDescribeRegexStreamCompletedEventArgs>
(NetClient_SelectPointDescribeRegexStreamCompleted);
//第三步：开始异步
mcbClient.SelectPointDescribeRegexStreamAsync(slevalue);
//接受事件
void NetClient_ SelectPointDescribeRegexStreamCompleted(object sender,
Apace.Client.McbService.SelectPointDescribeRegexStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginSelectPointDescribeRegexStream(String PointName, System. [AsyncCallback](#) callback, object asyncState) 开始异步委托

EndSelectPointDescribeRegexStream(System. [IAsyncResult](#) result) 结束异步委托

说明: System. [AsyncCallback](#) callback 回调方法

### 3.3.25 AddNewPoint

```
bool AddNewPoint(PcmData value);
```

说明: 通过实时服务将新的数据点标签加入到点配置区域

PARAMETER	DESCRIPTION
value	<a href="#">PcmData</a> 点标签基本信息，具体数据格式请参见数据契约。

RETURN	DESCRIPTION
bool	True 加入成功, False 加入失败。



#### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好数据

```
PcmData point = New PcmData();
    point.PointName = "DDM.Apace.Test1";
    point.OperationCode = "MHXT01JIZU001";
    point.OperationType= "锅炉系统";
    point.HighAlarm ="50000";
    point.LowAlarm = "30000" ;
    point.BosomAlarm ="-100";
    point.AllowMaxValue = "80000" ;
    point.AllowMinValue ="-1000";
    point.DataShape =0;
    point.StaticValue = 0;
    point.PointUnit = "XXX" ;
    point.PointDescribe ="Apace演示点标签";
    point.Precision = 7;
```

//第三步：获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(point.PointName);
```

```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
Stream returnValue = mcbClient.AddNewPoint(point);
```





## 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好数据

```
PcmData point = New PcmData();
    point.PointName = "DDM.Apace.Test1";
    point.OperationCode = "MHXT01JIZU001";
    point.OperationType= "锅炉系统";
    point.HighAlarm ="50000";
    point.LowAlarm = "30000" ;
    point.BosomAlarm ="-100";
    point.AllowMaxValue = "80000" ;
    point.AllowMinValue ="-1000";
    point.DataShape =0;
    point.StaticValue = 0;
    point.PointUnit = "XXX" ;
    point.PointDescribe ="Apace演示点标签";
    point.Precision = 7;
```

//第三步：获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(point.PointName);
```

```
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第三步：注册事件

```
mcbClient.AddNewPointCompleted += new EventHandler< Apace.Client.McbService.AddNewPointCompletedEventArgs>
    (NetClient_AddNewPointCompleted);
```

//第三步：开始异步

```
mcbClient.AddNewPointAsync(point);
```

//接受事件

```
void NetClient_ AddNewPointCompleted(object sender, Apace.Client.McbService.AddNewPointCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



## 相关方法（委托）：

System.IAsyncResult BeginAddNewPoint(PcmData value, System.AsyncCallback callback, object asyncState) 开始异步委托

EndAddNewPoint(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.26 AddNewPointStream

```
void AddNewPointStream(Stream Stream);
```

说明：通过实时服务将新点标签批量加入到点配置区域。

PARAMETER	DESCRIPTION
Stream	点标签基本信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



实例演示(略)

### 3.3.27 DeletePoint

```
bool DeletePoint(string PointName);
```

说明：通过实时服务删除配置区域中的指定点标签。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
bool	True 删除成功, False 删除失败.



实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String Pointname="DDM.Apace.test";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Pointname); Apace.Client.McbService.RealTimeContractClient mcbClient  
= client.GetMcbClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
bool returnValue = mcbClient.DeletePoint(Pointname);
```



实例演示（异步）

//第一步：得到客户端实体

```

Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apace.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.DeletePointCompleted += new EventHandler< Apace.Client.McbService.DeletePointCompletedEventArgs>
(NetClient_DeletePointCompleted);
//第三步：开始异步
mcbClient.DeletePointAsync(Pointname);
//接受事件
void NetClient_DeletePointCompleted(object sender, Apace.Client.McbService.DeletePointCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginDeletePoint(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndDeletePoint(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.28 GetPointInfo

```
PcmData GetPointInfo(string PointName)
```

说明：通过实时服务得到配置区域的点标签基本信息。

PARAMETER	DESCRIPTION
PointName	删除点标签名称，格式：[服务名].[目录名].[点名称]。

RETURN	DESCRIPTION
PcmData	点标签信息实体，请参见数据契约



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apac.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
//第四步：调用
PcmData returnValue = mcbClient.GetPointInfo(Pointname);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apac.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.GetPointInfoCompleted += new EventHandler< Apac.Client.McbService.GetPointInfoCompletedEventArgs>
(NetClient_GetPointInfoCompleted);
//第三步：开始异步
mcbClient.GetPointInfoAsync(Pointname);
//接受事件
void NetClient_GetPointInfoCompleted(object sender, Apac.Client.McbService.GetPointInfoCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        PcmData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetPointInfo(String PointName, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetPointInfo(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.29 GetAllPointInfoStream

```
Stream GetAllPointInfoStream();
```

说明：通过实时服务得到配置区域中所有的点标签基本信息数据流。

RETURN	DESCRIPTION
Stream	点标签基本信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
Stream returnValue = mcbClient.GetAllPointInfoStream();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
mcbClient.GetAllPointInfoStreamCompleted += new EventHandler<
Apac.Client.McbService.GetAllPointInfoStreamCompletedEventArgs>
(NetClient_GetAllPointInfoStreamCompleted);
//第三步：开始异步
mcbClient.GetAllPointInfoStreamAsync();
//接受事件
void NetClient_GetAllPointInfoStreamCompleted(object sender,
Apac.Client.McbService.GetAllPointInfoStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
    }
}
```

```

        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetAllPointInfoStream(System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndGetAllPointInfoStream(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

### 3.3.30 GetAllPointInfo

```
PcmData[] GetAllPointInfo();
```

说明：通过实时服务得到配置区域中所有的点标签基本信息数据集。

RETURN	DESCRIPTION
<a href="#">PcmData[]</a>	点标签基本信息, <a href="#">PcmData</a> 数据格式请参见数据契约



#### 实例演示（同步）

```

//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
PcmData[] returnValue = mcbClient.GetAllPointInfo();

```



#### 实例演示（异步）

```

//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件

```

```

mcbClient.GetAllPointInfoCompleted += new EventHandler<
Apace.Client.McbService.GetAllPointInfoCompletedEventArgs>
    (NetClient_GetAllPointInfoCompleted);
//第三步: 开始异步
mcbClient.GetAllPointInfoAsync();
//接受事件
void NetClient_GetAllPointInfoCompleted(object sender, Apace.Client.McbService.GetAllPointInfoCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        PcmData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginGetAllPointInfo(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetAllPointInfo(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.31 EditPoint

```
bool EditPoint(PcmData value);
```

说明：通过实时服务编辑配置区域所对应的点标签基本信息

PARAMETER	DESCRIPTION
value	点标签基本信息，PcmData 数据格式请参见数据契约

RETURN	DESCRIPTION
bool	True 修改成功，False 修改失败.



### 实例演示（同步）

```

//第一步: 得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步: 准备好数据

```

```

PcmData point = New PcmData();

    point.PointName = "DDM.Apacetest1";
    point.OperationCode = "MHXT01JIZU001";
    point.OperationType= "锅炉系统";
    point.HighAlarm ="50000";
    point.LowAlarm = "30000" ;
    point.BosomAlarm ="-100";
    point.AllowMaxValue = "80000" ;
    point.AllowMinValue ="-1000";
    point.DataShape =0;
    point.StaticValue = 0;
    point.PointUnit = "XXX" ;
    point.PointDescribe ="Apacetest点标签";
    point.Precision = 7;

//第三步：获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(point.PointName);
Apacetest.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
//第四步：调用
    bool returnValue = mcbClient.EditPoint(point);

```



### 实例演示（异步）

```

//第一步：得到客户端实体
Apacetest.Client.NetClient client = Apacetest.Client.NetClientManage.GetNetClient();
//第二步：准备好数据
PcmData point = New PcmData();

    point.PointName = "DDM.Apacetest1";
    point.OperationCode = "MHXT01JIZU001";
    point.OperationType= "锅炉系统";
    point.HighAlarm ="50000";
    point.LowAlarm = "30000" ;
    point.BosomAlarm ="-100";
    point.AllowMaxValue = "80000" ;
    point.AllowMinValue ="-1000";
    point.DataShape =0;
    point.StaticValue = 0;
    point.PointUnit = "XXX" ;
    point.PointDescribe ="Apacetest点标签";
    point.Precision = 7;

//第三步：获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(point.PointName);
Apacetest.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.EditPointCompleted += new EventHandler< Apacetest.Client.McbService.EditPoint CompletedEventArgs>
    (NetClient_EditPointCompleted);

```



```
//第三步：开始异步
mcbClient.EditPointAsync(point);
//接受事件
void NetClient_ EditPointCompleted(object sender, Apave.Client.McbService. EditPoint CompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. **IAsyncResult** Begin EditPoint (PcmData value, System. **AsyncCallback** callback, **object** asyncState) 开始异步委托

End EditPoint(System. **IAsyncResult** result) 结束异步委托

说明： System. **AsyncCallback** callback 回调方法

## 3.3.32 NewRealData

```
bool NewRealData(string PointName, Double value);
```

说明：追加新的时序数据到高速缓冲区。

PARAMETER	DESCRIPTION
PointName	点名，格式：[服务名].[目录名].[点名称]。
value	时值。

RETURN	DESCRIPTION
<b>bool</b>	True 追加成功，False 追加失败。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apave.test";
Double Pointvalue=12342.34;
//第三步：根据点名获得实时服务客户端访问实体
```

```
PointRule rule = BaseConfig. GetPointRule (Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
//第四步：调用
bool returnValue = mcbClient.NewRealData(Pointname, Pointvalue);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apace.test";
Double Pointvalue=12342.34;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
//第三步：注册事件
mcbClient.NewRealDataCompleted += new EventHandler< Apace.Client.McbService.NewRealDataCompletedEventArgs>
(NetClient_NewRealDataCompleted);
//第三步：开始异步
mcbClient.NewRealDataAsync(Pointname, Pointvalue);
//接受事件
void NetClient_ NewRealDataCompleted(object sender, Apace.Client.McbService.NewRealDataCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. **IAsyncResult** BeginNewRealData(String PointName, Double PointValue, System. **AsyncCallback** callback, **object** asyncState) 开始异步委托  
EndNewRealData(System. **IAsyncResult** result) 结束异步委托

说明： System. **AsyncCallback** callback 回调方法

## 3.3.33 NewRealDatas

```
void NewRealDatas(Stream data);
```

说明：追加一批新的时序数据到高速缓冲区。

PARAMETER	DESCRIPTION
data	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



实例演示（略）

### 3.3.34 NewRealDataValue

```
bool NewRealDataValue(string PointName, DateTime PointTime, Double value);
```

说明：追加新的时序数据到高速缓冲区,并指定追加所使用的时间戳。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apace.PCM [服务名].[目录名].[服务组件]

RETURN	DESCRIPTION
ProcessRunInfo	Apace 服务进程运行信息。



实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apace.test";
Double Pointvalue=12342.34;
DateTime PointTime=DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule.ServiceName, rule.DirName)
//第四步：调用
bool returnValue = mcbClient.NewRealDataValue(Pointname, Pointvalue, PointTime);
```



实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
//第二步：准备好点名
String Pointname="DDM.Apave.test";
Double Pointvalue=12342.34;
DateTime PointTime=DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Pointname);    Apave.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.NewRealDataValueCompleted += new EventHandler<
Apave.Client.McbService.NewRealDataValueCompletedEventArgs>
(NetClient_NewRealDataValueCompleted);
//第三步：开始异步
mcbClient.NewRealDataValueAsync (Pointname, Pointvalue, PointTime);
//接受事件
void NetClient_NewRealDataValueCompleted(object sender, Apave.Client.McbService.NewRealDataValueCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#)

BeginNewRealDataValue (String PointName, Double PointValue, DateTime PointTime, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndNewRealDataValue (System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

### 3.3.35 NewRealDataObjcet

```
bool NewRealDataObjcet (McbData Realvalue);
```

说明：追加新的时序数据到高速缓冲区。

PARAMETER	DESCRIPTION
Realvalue	实时数据实体， <a href="#">McbData</a> 数据格式请参见数据契约。

RETURN	DESCRIPTION
bool	True 追加成功, False 追加失败。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
McbData dat=new McbData();
dat.PointName="DDM.Apace.test";
dat.PointValue=12342.34;
dat.PointTime=DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第四步：调用
bool returnValue = mcbClient.NewRealDataObjcet (dat);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String Pointname="DDM.Apace.test";
Double Pointvalue=12342.34;
DateTime PointTime=DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (Pointname);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.NewRealDataObjcetCompleted += new EventHandler<
Apace.Client.McbService.NewRealDataObjcetCompletedEventArgs>
(NetClient_NewRealDataObjcetCompleted);
//第三步：开始异步
mcbClient.NewRealDataObjcetAsync (Pointname, Pointvalue, PointTime);
//接受事件
void NetClient_ NewRealDataObjcetCompleted(object sender, Apace.Client.McbService.NewRealDataObjcetCompletedEventArgs
e) {
    if (e.Error == null)
    {
        //成功取得换回值
        bool returnValue = e.Result;
```

```

    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#)

BeginNewRealDataObjcet (String PointName, Double PointValue, DateTime PointTime, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndNewRealDataObjcet (System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.36 GetSystemTime

```
DateTime GetSystemTime();
```

说明：得到实时服务器当前系统时间。

RETURN	DESCRIPTION
<a href="#">DateTime</a>	系统时间。



### 实例演示（同步）

```

//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
DateTime returnValue = mcbClient.GetSystemTime();

```



### 实例演示（异步）

```

//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
mcbClient.GetSystemTimeCompleted += new EventHandler< Apac.Client.McbService.GetSystemTimeCompletedEventArgs>

```

```

        (NetClient_GetSystemTimeCompleted);
        //第三步：开始异步
        mcbClient.GetSystemTimeAsync();
        //接受事件
void NetClient_GetSystemTimeCompleted(object sender, Apace.Client.McbService.GetSystemTimeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        DateTime returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginGetSystemTime(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetSystemTime(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.37 GetSystemUtcTime

```
int GetSystemUtcTime();
```

说明：得到实时服务器当前系统 ApaceUtc 时间。

RETURN	DESCRIPTION
int	Apace Utc 时间格式。



### 实例演示（同步）

```

//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apace)
//第四步：调用
int returnValue = mcbClient.GetSystemUtcTime();

```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apave";
//第三步：获得实时服务客户端访问实体
Apave.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apave)
//第三步：注册事件
mcbClient.GetSystemUtcTimeCompleted += new EventHandler<
Apave.Client.McbService.GetSystemUtcTimeCompletedEventArgs>
    (NetClient_GetSystemUtcTimeCompleted);
//第三步：开始异步
mcbClient.GetSystemUtcTimeAsync();
//接受事件
void NetClient_GetSystemUtcTimeCompleted(object sender, Apave.Client.McbService.GetSystemUtcTimeCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        int returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetSystemUtcTime(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetSystemUtcTime(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.38 GetConvertTime

```
DateTime GetConvertTime(int Utc);
```

说明： 将 Apave Utc 时间转换成.net 时间格式。

PARAMETER	DESCRIPTION
Utc	Apave Utc 时间格式。

RETURN	DESCRIPTION
--------	-------------





## 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
DateTime returnValue = mcbClient.GetSystemUtcTime(95594656300);
```



## 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
mcbClient.GetSystemUtcTimeCompleted += new EventHandler<
Apac.Client.McbService.GetSystemUtcTimeCompletedEventArgs>
(NetClient_GetSystemUtcTimeCompleted);
//第三步：开始异步
mcbClient.GetSystemUtcTimeAsync();
//接受事件
void NetClient_GetSystemUtcTimeCompleted(object sender, Apac.Client.McbService.GetSystemUtcTimeCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        DateTime returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



## 相关方法（委托）：

System.IAsyncResult BeginGetSystemUtcTime(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetSystemUtcTime(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.39 GetPointsGuid

```
string GetPointsGuid(string[] PointNames);
```

说明：通过发送指定的数据点名集合得到该集合的唯一标示码。

PARAMETER	DESCRIPTION
PointNames	点标签名称集合，每个点必须遵循全点名格式。

RETURN	DESCRIPTION
string	Guid 唯一表示码。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();

//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";

//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Points[0]);  Apac.Client.McbService.RealTimeContractClient mcbClient =
client.GetMcbClient(rule. ServiceName, rule.DirName)

//第四步：调用
string returnValue = mcbClient.GetPointsGuid(Points);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();

//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";

//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Points[0]);  Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)

//第三步：注册事件
mcbClient.GetPointsGuidCompleted += new EventHandler< Apac.Client.McbService.GetPointsGuidCompletedEventArgs>
```

```

        (NetClient_GetPointsGuidCompleted);
        //第三步：开始异步
        mcbClient.GetPointsGuidAsync (Points);
        //接受事件
void NetClient_ GetPointsGuidCompleted(object sender, Apave.Client.McbService.GetPointsGuidCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        string returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetPointsGuid(String[] Points, System. [AsyncCallback](#) callback, object asyncState) 开始异步委托

EndGetPointsGuid(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.40 GetPointsGuidStream

```
Stream GetPointsGuidStream(string PointGuid);
```

说明：通过唯一表示码得到最新实时数据流。

PARAMETER	DESCRIPTION
PointGuid	Guid 唯一标示码。

RETURN	DESCRIPTION
<a href="#">Stream</a>	点标签实时信息格式化数据流 说明：Apave 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
```

```
//第二步: 准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apace.test";
Points [1]="DDM.Apace.test1";
//第三步: 根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule(Points[0]);    Apace. Client. McbService. RealTimeContractClient mcbClient =
client. GetMcbClient(rule. ServiceName, rule. DirName)
//第四步: 调用
string returnGuid = mcbClient. GetPointsGuid(Points);
Stream returnValue = mcbClient. GetPointsGuidStream(returnGuid);
```



### 实例演示（异步）

```
//第一步: 得到客户端实体
Apace. Client. NetClient client = Apace. Client. NetClientManage. GetNetClient ();
//第二步: 准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apace.test";
Points [1]="DDM.Apace.test1";
//第三步: 根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apace. Client. McbService. RealTimeContractClient mcbClient
= client. GetMcbClient(rule. ServiceName, rule. DirName)
//第三步: 注册事件
mcbClient. GetPointsGuidStreamCompleted += new EventHandler< Apace. Client. McbService. GetPointsGuidStream
CompletedEventArgs>
(NetClient_ GetPointsGuidStreamCompleted);
//第三步: 开始异步
string returnGuid = mcbClient. GetPointsGuid(Points);
mcbClient. GetPointsGuidAsync(returnGuid);
//接受事件
void NetClient_ GetPointsGuidStreamCompleted(object sender, Apace. Client. McbService. GetPointsGuidStream
CompletedEventArgs e) {
    if (e. Error == null)
    {
        //成功取得换回值
        Stream returnValue = e. Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）:

System. [IAsyncResult](#) BeginGetPointsGuidStream(String PointGuid, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
End GetPointsGuidStream(System. [IAsyncResult](#) result) 结束异步委托

说明: System. [AsyncCallback](#) callback 回调方法

### 3.3.41 RemovePointGuid

```
void RemovePointGuid(string PointGuid);
```

说明: 移除高速缓冲区中用于标示集合点的 Guid 表示码。

PARAMETER	DESCRIPTION
PointGuid	Guid 唯一标示码。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";
string returnGuid = mcbClient.GetPointsGuid(Points);
mcbClient.RemovePointGuid(returnGuid);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
mcbClient.RemovePointGuidCompleted += new EventHandler<
Apac.Client.McbService.RemovePointGuidCompletedEventArgs>
(NetClient_RemovePointGuidCompleted);
//第四步：开始异步
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
```

```

Points [1]="DDM.Apace.test1";
string returnGuid = mcbClient.GetPointsGuid(Points);
mcbClient.RemovePointGuidAsync(returnGuid);
//接受事件
void NetClient_ RemovePointGuidCompleted(object sender, Apace.Client.McbService.RemovePointGuidCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        McbData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



#### 相关方法（委托）：

System.IAsyncResult BeginRemovePointGuid(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndRemovePointGuid(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.42 GetSpacePointStream

```
Stream GetSpacePointStream(int Offset, int Length);
```

说明：获得高速缓冲区中按索引指定长度的实时数据流。

PARAMETER	DESCRIPTION
Offset	索引偏移量
Length	提取实时数据点的总数

RETURN	DESCRIPTION
Stream	点标签实时信息格式化数据流 说明：Apace 所使用的格式化并非是.Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示（同步）

```

//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();

```

```
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apace)
//第四步：调用
Stream returnValue = mcbClient.GetSpacePointStream(0, 100000);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apace";
//第三步：获得实时服务客户端访问实体
Apace.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apace)
//第三步：注册事件
mcbClient.GetSpacePointStreamCompleted += new EventHandler<
Apace.Client.McbService.GetSpacePointStreamCompletedEventArgs>
(NetClient_GetSpacePointStreamCompleted);
//第三步：开始异步
mcbClient.GetSpacePointStreamAsync(0, 100000);
//接受事件
void NetClient_GetSpacePointStreamCompleted(object sender,
Apace.Client.McbService.GetSpacePointStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetSpacePointStream(int Offset, int Length, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetSpacePointStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.43 GetPointInfos

```
PcmData[] GetPointInfos(string[] PointNames);
```

说明：通过实时服务得到配置区域中指定点集合的基本信息。

PARAMETER	DESCRIPTION
PointNames	点标签名称集合，每个点必须遵循全点名格式。

RETURN	DESCRIPTION
PcmData[]	点标签基本信息集合体，PcmData 数据格式请参见数据契约



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Points[0]);    Apac.Client.McbService.RealTimeContractClient mcbClient =
client.GetMcbClient(rule. ServiceName, rule.DirName)
//第四步：调用
PcmData[] returnValue = mcbClient.GetPointInfos(Points);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (Points[0]);    Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.GetPointInfosCompleted += new EventHandler< Apac.Client.McbService.GetPointInfosCompletedEventArgs>
(NetClient_GetPointInfosCompleted);
//第三步：开始异步
mcbClient.GetPointInfosAsync(Points);
//接受事件
```



```
void NetClient_ GetPointInfosCompleted(object sender, Apave.Client.McbService.GetPointInfosCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        PcmData[] returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetPointInfos(String[] Points, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetPointInfos(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.44 GetPointInfosStream

```
Stream GetPointInfosStream(string[] PointNames);
```

说明：通过实时服务得到配置区域中指定点集合的基本信息数据流。

PARAMETER	DESCRIPTION
PointNames	点标签名称集合，每个点必须遵循全点名格式。

RETURN	DESCRIPTION
Stream	点标签基本信息格式化数据流 说明：Apave 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apave.test";
Points [1]="DDM.Apave.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Points[0]);    Apave.Client.McbService.RealTimeContractClient mcbClient =
```

```
client.GetMcbClient(rule. ServiceName, rule. DirName)
```

//第四步：调用

```
Stream returnValue = mcbClient.GetPointInfosStream(Points);
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String[] Points=new String[2];
```

```
Points [0]="DDM.Apave.test";
```

```
Points [1]="DDM.Apave.test1";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apave.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule. DirName)
```

//第三步：注册事件

```
mcbClient.GetPointInfosStreamCompleted += new EventHandler<
```

```
Apave.Client.McbService.GetPointInfosStreamCompletedEventArgs>
```

```
(NetClient_GetPointInfosStreamCompleted);
```

//第三步：开始异步

```
mcbClient.GetPointInfosStreamAsync(Points);
```

//接受事件

```
void NetClient_ GetPointInfosStreamCompleted(object sender,
Apave.Client.McbService.GetPointInfosStreamCompletedEventArgs e) {
```

```
    if (e.Error == null)
```

```
    {
```

```
        //成功取得换回值
```

```
        Stream returnValue = e.Result;
```

```
    }
```

```
    else
```

```
    {
```

```
        //网络异常
```

```
    }
```

```
}
```



### 相关方法（委托）：

System. IAsyncResult BeginGetPointInfosStream(String[] Points, System. AsyncCallback callback, object asyncState) 开始异步委托

EndGetPointInfosStream(System. IAsyncResult result) 结束异步委托

说明： System. AsyncCallback callback 回调方法

## 3.3.45 GetSpacePointInfoStream

```
Stream GetSpacePointInfoStream(int Offset, int Length);
```

说明：通过实时服务获得配置区域中按索引指定长度的点标签基本信息数据流。

PARAMETER	DESCRIPTION
Offset	索引偏移量
Length	提取实时数据点的总数

RETURN	DESCRIPTION
<code>Stream</code>	点标签基本信息格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的格式化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
Stream returnValue = mcbClient.GetSpacePointInfoStream(0, 100000);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
mcbClient.GetSpacePointInfoStreamCompleted += new EventHandler<
Apac.Client.McbService.GetSpacePointInfoStreamCompletedEventArgs>
(NetClient_GetSpacePointInfoStreamCompleted);
//第三步：开始异步
mcbClient.GetSpacePointInfoStreamAsync(0, 100000);
//接受事件
void NetClient_GetSpacePointInfoStreamCompleted(object sender,
Apac.Client.McbService.GetSpacePointInfoStreamCompletedEventArgs e) {
    if (e.Error == null)
    {
```

```
//成功取得换回值
Stream returnValue = e.Result;
}
else
{
    //网络异常
}
}
```



#### 相关方法（委托）：

System.IAsyncResult BeginGetSpacePointInfoStream(int Offset, int Length, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetSpacePointInfoStream(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

### 3.3.46 GetPointCount

```
int GetPointCount();
```

说明：得到高速缓冲区中实时数据总数。

RETURN	DESCRIPTION
int	数据点总数。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetMcbClient(Servicename, Apac)
//第四步：调用
int returnValue = mcbClient.GetPointCount();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好网络路径
String Servicename="DDM"; String Dirname="Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient= client.GetMcbClient(Servicename, Apac)
//第三步：注册事件
```

```

mcbClient.GetPointCountCompleted += new EventHandler< Apave.Client.McbService.GetPointCountCompletedEventArgs>
(NetClient_GetPointCountCompleted);
//第三步：开始异步
mcbClient.GetPointCountAsync();
//接受事件
void NetClient_GetPointCountCompleted(object sender, Apave.Client.McbService.GetPointCountCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        int returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginGetPointCount(System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetPointCount(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.47 Math\_Point\_Sum

```
double Math_Point_Sum(string[] Points);
```

说明： 计算指定点集合的累加值。

PARAMETER	DESCRIPTION
Points	点标签名称集合，每个点必须遵循全点名格式。

RETURN	DESCRIPTION
double	累加结果值。



### 实例演示（同步）

```

//第一步：得到客户端实体
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apave.test";

```

```
Points [1]="DDM.Apace.test1";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient =
client.GetMcbClient(rule. ServiceName, rule.DirName)
```

//第四步：调用

```
double returnValue = mcbClient.Math_Point_Sum(Points);
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String[] Points=new String[2];
```

```
Points [0]="DDM.Apace.test";
```

```
Points [1]="DDM.Apace.test1";
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
```

//第三步：注册事件

```
mcbClient.Math_Point_SumCompleted += new EventHandler< Apace.Client.McbService.Math_Point_SumCompletedEventArgs>
(NetClient_Math_Point_SumCompleted);
```

//第三步：开始异步

```
mcbClient.Math_Point_SumAsync(Points);
```

//接受事件

```
void NetClient_ Math_Point_SumCompleted(object sender, Apace.Client.McbService.Math_Point_SumCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Point\_Sum(String[] Points, System. [AsyncCallback](#) callback, [object](#) asyncState)

开始异步委托

EndMath\_Point\_Sum(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

## 3.3.48 Math\_Point\_Avg

```
double Math_Point_Avg(string[] Points);
```

说明：计算指定点集的算术平均值。

PARAMETER	DESCRIPTION
Points	点标签名称集合，每个点必须遵循全点名格式。

RETURN	DESCRIPTION
double	算术平均值。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Points[0]);    Apac.Client.McbService.RealTimeContractClient mcbClient =
client.GetMcbClient(rule. ServiceName, rule.DirName)
//第四步：调用
double returnValue = mcbClient.Math_Point_Avg(Points);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apac.test";
Points [1]="DDM.Apac.test1";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (Points[0]);    Apac.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.Math_Point_AvgCompleted += new EventHandler< Apac.Client.McbService.Math_Point_AvgCompletedEventArgs>
(NetClient_Math_Point_AvgCompleted);
//第三步：开始异步
mcbClient.Math_Point_AvgAsync(Points);
//接受事件
```

```
void NetClient_ Math_Point_AvgCompleted(object sender, Apave.Client.McbService.Math_Point_AvgCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginMath\_Point\_Avg(String[] Points, System.AsyncCallback callback, object asyncState)

开始异步委托

EndMath\_Point\_Avg(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

## 3.3.49 Math\_Point\_Wtd

```
double Math_Point_Wtd(string[] Points, string[] wPoints);
```

说明： 计算指定点集合的加权平均值。

PARAMETER	DESCRIPTION
Points	点标签名称集合, 计算目标点, 每个点必须遵循全点名格式。
wPoints	点标签名称集合, 计算权点, 每个点必须遵循全点名格式。

RETURN	DESCRIPTION
double	加权平均值。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apave.Client.NetClient client = Apave.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String[] Points=new String[2];
```

```
Points [0]="DDM.Apave.test";
```

```
Points [1]="DDM.Apave.test1";
```

```
String[] wPoints=new String[2];
```

```
wPoints [0]="DDM.Apave.test11";
```



```
wPoints [1]="DDM.Apace.test12";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient =
client.GetMcbClient(rule. ServiceName, rule.DirName)
//第四步：调用
double returnValue = mcbClient.Math_Point_Wtd(Points, wPoints);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String[] Points=new String[2];
Points [0]="DDM.Apace.test";
Points [1]="DDM.Apace.test1";

String[] wPoints=new String[2];
wPoints [0]="DDM.Apace.test11";
wPoints [1]="DDM.Apace.test12";

//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (Points[0]);    Apace.Client.McbService.RealTimeContractClient mcbClient
= client.GetMcbClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
mcbClient.Math_Point_WtdCompleted += new EventHandler< Apace.Client.McbService.Math_Point_WtdCompletedEventArgs>
(NetClient_Math_Point_WtdCompleted);
//第三步：开始异步
mcbClient.Math_Point_WtdAsync(Points, wPoints);
//接受事件
void NetClient_ Math_Point_WtdCompleted(object sender, Apace.Client.McbService.Math_Point_WtdCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Point\_Wtd(String[] Points, String[] wPoints, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托  
EndMath\_Point\_Wtd(System. [IAsyncResult](#) result) 结束异步委托

说明： System.[AsyncCallback](#) callback 回调方法

## 3.4 历史服务(Apace 时序历史数据管理)



### 3.4.1 AddNewRows

```
void AddNewRows(HisRowData[] values);
```

说明：将新的数据追加到历史管理区域。

PARAMETER	DESCRIPTION
values	追加历史数据集合， <a href="#">HisRowData</a> 格式请参见数据契约



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
HisRowData[] HisDatas=new HisRowData[2];
HisDatas [0].PointName ="DDM.Apace.test";
HisDatas [0].PointTime =Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);
HisDatas [0].PointValue=123345.3455;
HisDatas [1].PointName ="DDM.Apace.test1";
HisDatas [1].PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);
HisDatas [1].PointValue =123345.3455;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (HisDatas[0].PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
HisClient.AddNewRows(HisDatas);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
HisRowData [] HisDatas=new HisRowData[2];
HisDatas [0].PointName ="DDM.Apace.test";
HisDatas [0].PointTime =Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);
HisDatas [0].PointValue=123345.78773;
```

```
HisDatas [1].PointName = "DDM.Apace.test1";
HisDatas [1].PointTime = Apace.Common.BaseConfig.GetUtcTime(DateTime.Now);
HisDatas [1].PointValue = 123345.3455;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule (HisDatas[0].PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.AddNewRowsCompleted += new EventHandler< Apace.Client.HisService.AddNewRowsCompletedEventArgs>
(NetClient_AddNewRowsCompleted);
//第三步：开始异步
mcbClient.AddNewRowsAsync (HisDatas);
//接受事件
void NetClient_AddNewRowsCompleted(object sender, Apace.Client.HisService.AddNewRowsCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功
    }
    else
    {
        //网络异常
    }
}
```

**相关方法（委托）：**

System.IAsyncResult BeginAddNewRows (HisRowData[] values, System.AsyncCallback callback, object asyncState)  
开始异步委托  
EndAddNewRows (System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示追加历史数据的形式，建议在实际的调用过程中请使用 Apace.Client.NetClient 中所提供的对应方法

### 3.4.2 AddNewRowStream

```
void AddNewRowStream(Stream values);
```

说明： 以流的信息将新的数据追加到历史管理区域。

PARAMETER	DESCRIPTION
values	历史数据格式化数据流 说明： Apace 所使用的格式化并非是.Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



实例演示（略）

### 3.4.3 GetHisData

```
Stream GetHisData(string PointName, DateTime start, DateTime end);
```

说明：得到指点标签规定时间范围类所有实际发生的时序历史数据流。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间

RETURN	DESCRIPTION
Stream	历史数据格式化数据流 说明：Apace 所使用的格式化并非是.Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
Stream returnvalue= HisClient.GetHisData(PointName, stime, etime);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
```

```
//第三步：注册事件
HisClient.GetHisDataCompleted += new EventHandler< Apave.Client.HisService.GetHisDataCompletedEventArgs>
(NetClient_GetHisDataCompleted);
//第三步：开始异步
mcbClient.GetHisDataAsync(PointName, stime, etime);
//接受事件
void NetClient_GetHisDataCompleted(object sender, Apave.Client.HisService.GetHisDataCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginGetHisData([HisRowData\[\]](#) values, System. [AsyncCallback](#) callback, [object](#) asyncState)

开始异步委托

EndGetHisData(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apave.Client. [NetClient](#) 中所提供的对应方法

## 3.4.4 GetBufferData

```
Stream GetBufferData(string PointName);
```

说明：得到指定点当前存放在历史区域的数据流。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM. Apave. Test [服务名].[目录名].[点名]

RETURN	DESCRIPTION
<a href="#">Stream</a>	历史数据格式化数据流 说明：Apave 所使用的格式化并非是 .Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apac.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
Stream returnvalue= HisClient.GetBufferData(PointName);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apac.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.GetBufferDataCompleted += new EventHandler< Apac.Client.HisService.GetBufferDataCompletedEventArgs>
(NetClient_GetBufferDataCompleted);
//第三步：开始异步
mcbClient.GetBufferDataAsync(PointName);
//接受事件
void NetClient_GetBufferDataCompleted(object sender, Apac.Client.HisService.GetBufferDataCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetBufferData(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托  
 EndGetBufferData(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 `Apace.Client.NetClient` 中所提供的对应方法

### 3.4.5 GetStepHisData

```
Stream GetStepHisData(string PointName, DateTime start, DateTime end, int step);
```

说明：得到指点标签规定时间范围类所有按步长划分的历史数据流。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间
step	用于划分历史数据的步长值

RETURN	DESCRIPTION
Stream	历史数据格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
String PointName = "DDM.Apace.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
Stream returnvalue= HisClient.GetStepHisData(PointName, stime, etime, Step);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
```

```
String PointName = "DDM.Apave.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
Int Step=30;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (PointName);
Apave.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
HisClient.GetStepHisDataCompleted += new EventHandler< Apave.Client.HisService.GetStepHisDataCompletedEventArgs>
(NetClient_GetStepHisDataCompleted);
//第三步：开始异步
mcbClient.GetStepHisDataAsync(PointName, stime, etime, Step);
//接受事件
void NetClient_ GetStepHisDataCompleted(object sender, Apave.Client.HisService.GetStepHisDataCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. IAsyncResult BeginGetStepHisData(HisRowData[] values, System. AsyncCallback callback, object asyncState) 开始异步委托  
EndGetStepHisData(System. IAsyncResult result) 结束异步委托

说明： System. AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apave.Client.NetClient 中所提供的对应方法

## 3.4.6 GetStepHisDataProxy

```
Stream GetStepHisDataProxy(string PointName, DateTime start, DateTime end, int step);
```

说明： 得到指点标签规定时间范围类所有按步长划分的历史数据流,代理服务专用。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apave.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间



step	用于划分历史数据的步长值
------	--------------

RETURN	DESCRIPTION
Stream	历史数据格式化数据流 说明：Apace 所使用的格式化并非是 .Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
String PointName = "DDM.Apace.test";
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
Stream returnvalue= HisClient.GetStepHisDataProxy(PointName, stime, etime, Step);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
Int Step=30;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.GetStepHisDataProxyCompleted += new EventHandler<
Apace.Client.HisService.GetStepHisDataProxyCompletedEventArgs>
(NetClient_GetStepHisDataProxyCompleted);
//第三步：开始异步
mcbClient.GetStepHisDataProxyAsync(PointName, stime, etime, Step);
//接受事件
void NetClient_GetStepHisDataProxyCompleted(object sender,
```

```

Apave.Client.HisService.GetStepHisDataProxyCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System.IAsyncResult BeginGetStepHisDataProxy(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托

EndGetStepHisDataProxy(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apave.Client.NetClient 中所提供的对应方法

## 3.4.7 GetNextHisData

```
Stream GetNextHisData(HisData headData);
```

说明： 获得下一块时序历史数据。

PARAMETER	DESCRIPTION
headData	服务进程名称，如：DDM.Apave.PCM [服务名].[目录名].[服务组件]

RETURN	DESCRIPTION
Stream	历史数据格式化数据流 说明：Apave 所使用的格式化并非是.Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。

## 3.4.8 GetPreHisData

```
Stream GetPreHisData(HisData headData);
```

说明： 获得上一块时序历史数据。

PARAMETER	DESCRIPTION
-----------	-------------

ServiceKey	服务进程名称，如：DDM.Apave.PCM [服务名].[目录名].[服务组件]
------------	---

RETURN	DESCRIPTION
Stream	历史数据格式化数据流 说明：Apave 所使用的格式化并非是.Net 框架类库中所提供的序列化方式，大部分的数据契约都有其自身的格式化和反格式方式。



实例演示（略）

### 3.4.9 UpdateHisData

```
bool UpdateHisData(int filename, HisData headData, HisRowData[] values);
```

说明：修改历史数据。



实例演示（略）

### 3.4.10 GetHisPointMoment

```
HisRowData GetHisPointMoment(string PointName, DateTime Time);
```

说明：得到指定点限定时间开始的第一块历史数据。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apave.Test [服务名].[目录名].[点名]
Time	获取历史数据限定的开始时间

RETURN	DESCRIPTION
HisRowData	历史数据，HisRowData 格式请参见数据契约。



实例演示（略）

### 3.4.11 Math\_Avg

```
double Math_Avg(string PointName, DateTime start, DateTime end);
```

说明：计算出指定点在规定范围内的算术平均值。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间

RETURN	DESCRIPTION
double	平均值。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
double returnvalue= HisClient.Math_Avg(PointName, stime, etime);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_AvgCompleted += new EventHandler< Apace.Client.HisService.Math_AvgCompletedEventArgs>
(NetClient_Math_AvgCompleted);
//第三步：开始异步
```

```
mcbClient.Math_AvgAsync(PointName, stime, etime);
//接受事件
void NetClient_ Math_AvgCompleted(object sender, Apace.Client.HisService.Math_AvgCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Avg(HisRowData[] values, System. [AsyncCallback](#) callback, object asyncState) 开始异步委托

EndMath\_Avg(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client. [NetClient](#) 中所提供的对应方法

## 3.4.12 Math\_Min

```
double Math_Min(string PointName, DateTime start, DateTime end);
```

说明： 计算出指定点在规定范围内的最小值。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apacetest [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间

RETURN	DESCRIPTION
double	最小值。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
String PointName = "DDM.Apacetest";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
double returnvalue= HisClient.Math_Min(PointName, stime, etime);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_MinCompleted += new EventHandler< Apace.Client.HisService.Math_MinCompletedEventArgs>
(Math_MinCompleted);
//第三步：开始异步
mcbClient.Math_MinAsync(PointName, stime, etime);
//接受事件
void NetClient_ Math_MinCompleted(object sender, Apace.Client.HisService.Math_MinCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginMath\_Min(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托

EndMath\_Min(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client.NetClient 中所提供的对应方法

### 3.4.13 Math\_Max

```
double Math_Max(string PointName, DateTime start, DateTime end);
```

说明：计算出指定点在规定范围内的最大值。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间

RETURN	DESCRIPTION
double	最大值。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
double returnvalue= HisClient.Math_Max(PointName, stime, etime);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_MaxCompleted += new EventHandler< Apace.Client.HisService.Math_MaxCompletedEventArgs>
```

```

        (NetClient_Math_MaxCompleted);
        //第三步：开始异步
        mcbClient.Math_MaxAsync(PointName, stime, etime);
        //接受事件
void NetClient_Math_MaxCompleted(object sender, Apave.Client.HisService.Math_MaxCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        double returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Max([HisRowData](#)[] values, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndMath\_Max(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apave.Client. [NetClient](#) 中所提供的对应方法

## 3.4.14 Math\_MaxMinAvg

```
MathData Math_MaxMinAvg(string PointName, DateTime start, DateTime end)
```

说明： 计算出指定点在规定范围内的最大值，最小值，平均值。

PARAMETER	DESCRIPTION
PointName	点名称，如：DDM.Apave.Test [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间

RETURN	DESCRIPTION
<a href="#">MathData</a>	最大值、最小值、平均值实体。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apave.Client. NetClient client = Apave.Client. NetClientManage.GetNetClient();
```



```
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
MathData returnvalue= HisClient.Math_MaxMinAvg(PointName, stime, etime);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apace.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_MaxMinAvgCompleted += new EventHandler< Apace.Client.HisService.Math_MaxMinAvgCompletedEventArgs>
(MathClient_Math_MaxMinAvgCompleted);
//第三步：开始异步
mcbClient.Math_MaxMinAvgAsync(PointName, stime, etime);
//接受事件
void NetClient_Math_MaxMinAvgCompleted(object sender, Apace.Client.HisService.Math_MaxMinAvgCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        MathData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginMath\_MaxMinAvg(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托

EndMath\_MaxMinAvg(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client.NetClient 中所提供的对应方

### 3.4.15 Math\_Wtd

```
double Math_Wtd(string SPointName, string WPointName, DateTime start, DateTime end, int step)
```

说明：计算出指定点在规定范围内的加权平均值。

PARAMETER	DESCRIPTION
SPointName	计算源点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
WPointName	加权点名称，如：DDM.Apace.Test1 [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间
step	步长

RETURN	DESCRIPTION
double	加权平均值。



#### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
String SPointName = "DDM.Apace.test";
```

```
String WPointName = "DDM.Apace.test1";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(PointName);
```

```
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
MathData returnvalue= HisClient.Math_Wtd(SPointName, WPointName, stime, etime,);
```



#### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String PointName = "DDM.Apace.test";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
```

```
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig. GetPointRule (PointName);
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule. ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_WtdCompleted += new EventHandler< Apace.Client.HisService.Math_WtdCompletedEventArgs>
(MathClient_Math_WtdCompleted);
//第三步：开始异步
mcbClient.Math_WtdAsync(SPointName, WPointName, PointName, stime, etime);
//接受事件
void NetClient_ Math_WtdCompleted(object sender, Apace.Client.HisService.Math_WtdCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        MathData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Wtd(HisRowData[] values, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndMath\_Wtd(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client. [NetClient](#) 中所提供的对应方法

## 3.4.16 Math\_WtdAndCheck

```
double Math_WtdAndCheck(string SPointName, string WPointName, DateTime start, DateTime end, int step, double wMax, double wMin)
```

说明： 计算出指定点在规定范围内的加权平均值,并检测权值的有效范围。

PARAMETER	DESCRIPTION
SPointName	计算源点名称，如：DDM. Apace. Test [服务名]. [目录名]. [点名]
WPointName	加权点名称，如：DDM. Apace. Test1 [服务名]. [目录名]. [点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间
step	步长

RETURN	DESCRIPTION
double	加权平均值。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
String SPointName = "DDM.Apac.test";
String WPointName = "DDM.Apac.test1";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
MathData returnvalue= HisClient.Math_WtdAndCheck(SPointName, WPointName, stime, etime, 0, 100);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apac.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_WtdAndCheckCompleted += new EventHandler<
Apac.Client.HisService.Math_WtdAndCheckCompletedEventArgs>
(MathClient_Math_WtdAndCheckCompleted);
//第三步：开始异步
mcbClient.Math_WtdAndCheckAsync(SPointName, WPointName, PointName, stime, etime, 0, 100);
//接受事件
void NetClient_Math_WtdAndCheckCompleted(object sender, Apac.Client.HisService.Math_WtdAndCheckCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        MathData returnValue = e.Result;
    }
}
```

```

else
{
    //网络异常
}
}

```



### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_WtdAndCheck([HisRowData\[\]](#) values, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndMath\_WtdAndCheck(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 [Apace.Client.NetClient](#) 中所提供的对应方法

## 3.4.17 Math\_Correlation

```
double Math_Correlation(string SPointName, string WPointName, DateTime start, DateTime end, int step)
```

说明： 计算出两个点曲线相识度。

PARAMETER	DESCRIPTION
SPointName	计算源点名称，如：DDM.Apace.Test [服务名].[目录名].[点名]
WPointName	加权点名称，如：DDM.Apace.Test1 [服务名].[目录名].[点名]
start	获取历史数据时限定的开始时间
end	获取历史数据时限定的结束时间
step	步长

RETURN	DESCRIPTION
<a href="#">double</a>	曲线相识度值。



### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
String SPointName = "DDM.Apace.test";
```

```
String WPointName = "DDM.Apace.test1";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(PointName);
```

```
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
MathData returnValue= HisClient.Math_Correlation(SPointName, WPointName, stime, etime,0,100);
```



### 实例演示（异步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

//第二步：准备好点名

```
String PointName = "DDM.Apace.test";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(PointName);
```

```
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
```

//第三步：注册事件

```
HisClient.Math_CorrelationCompleted += new EventHandler<
```

```
Apace.Client.HisService.Math_CorrelationCompletedEventArgs>
```

```
(NetClient_Math_CorrelationCompleted);
```

//第三步：开始异步

```
mcbClient.Math_CorrelationAsync(SPointName, WPointName, PointName, stime, etime, 0, 100);
```

//接受事件

```
void NetClient_Math_CorrelationCompleted(object sender, Apace.Client.HisService.Math_CorrelationCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        MathData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginMath\_Correlation(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托

EndMath\_Correlation(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client.NetClient 中所提供的对应方法

## 3.5

## 计算服务(Apace 计算管理)



**说明：**

下面将会给出计算服务的所有接口形式，但没有给出实际的代码示例，原因并不建议通过代码形式进行计算服务的相关操作，可以通过管理中心提供的计算管理工具进行相应的操作。

### 3.5.1 AddAigorithm

```
string AddAigorithm(Stream context);
```

说明：加入新的计算法则。

PARAMETER	DESCRIPTION
Stream	加入新的计算法则。

RETURN	DESCRIPTION
string	状态信息，如果为空表示成功，否则失败并返回异常信息。

### 3.5.2 DeleteAigorithm

```
bool DeleteAigorithm(int Interval, string Language, string CalcPointName);
```

说明：删除指定的计算法则。

PARAMETER	DESCRIPTION
Interval	计算周期（单位秒）
Language	计算法则所使用的语言 C#、VB.Net、JScript
CalcPointName	计算法则名称

RETURN	DESCRIPTION
bool	True 删除成功，False 删除失败。

### 3.5.3 DeleteAigorithmElement

```
bool DeleteAigorithmElement(Stream context);
```

说明：删除指定的计算法则。

PARAMETER	DESCRIPTION
context	服务进程名称，如：DDM.Apac.PCM [服务名].[目录名].[服务组件]

RETURN	DESCRIPTION
bool	True 删除成功，False 删除失败。

### 3.5.4 Contains

```
bool Contains(string CalcName, string Language, int Interval);
```

说明：检测是否存在对应的计算法则。

PARAMETER	DESCRIPTION
Interval	计算周期（单位秒）
Language	计算法则所使用的语言 C#、VB.Net、JScript
CalcName	计算法则名称

RETURN	DESCRIPTION
bool	True 存在，False 不存在

### 3.5.5 UpdateAigorithm

```
string UpdateAigorithm(Stream context);
```

说明：更新对应的计算法则。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apac.PCM [服务名].[目录名].[服务组件]

RETURN	DESCRIPTION
string	状态信息，如果为空表示成功，否则失败并返回异常信息。

### 3.5.6 GetAllAigorithm



```
Stream GetAllAigorithm();
```

说明：获取所有的告警法则配置信息流。

RETURN	DESCRIPTION
Stream	计算法则配置集合序列化流。

### 3.5.7 AddFiledll

```
bool AddFiledll(string filename, int Interval, string Language, byte[] dllfile);
```

说明：添加指定的程序集文件到远程服务器。

PARAMETER	DESCRIPTION
filename	程序集文件名
Interval	计算周期
Language	脚本语言
dllfile	程序集内容

RETURN	DESCRIPTION
bool	True 成功, False 失败

### 3.5.8 UpdateFiledll

```
bool UpdateFiledll(string filename, int Interval, string Language, byte[] dllfile);
```

说明：更新指定的程序集文件到远程服务器。

PARAMETER	DESCRIPTION
filename	程序集文件名
Interval	计算周期
Language	脚本语言
dllfile	程序集内容

RETURN	DESCRIPTION
--------	-------------

<code>bool</code>	True 成功, False 失败
-------------------	-------------------

### 3.5.9 GetIntervalAigorithm

```
Stream GetIntervalAigorithm(int Interval);
```

说明：得到指定计算周期下所有的计算法则配置信息流。

PARAMETER	DESCRIPTION
Interval	计算周期

RETURN	DESCRIPTION
Stream	计算法则配置集合序列化流。

### 3.5.10 GetAllLocalConfig

```
Stream GetAllLocalConfig();
```

说明：得到计算服务的全局配置信息。

RETURN	DESCRIPTION
Stream	计算服务全局配置信息流。



#### 实例演示（同步）

//第一步：得到客户端实体

```
Apace.Client.NetClient client = Apace.Client.NetClientManage.GetNetClient();
```

```
String SPointName = "DDM.Apace.test";
```

```
String WPointName = "DDM.Apace.test1";
```

```
DateTime stime = DateTime.Now.AddHours(-24);
```

```
DateTime etime = DateTime.Now;
```

//第三步：根据点名获得实时服务客户端访问实体

```
PointRule rule = BaseConfig.GetPointRule(PointName);
```

```
Apace.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
```

//第四步：调用

```
MathData returnvalue= HisClient.Math_Correlation(SPointName, WPointName, stime, etime, 0, 100);
```



## 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();

//第二步：准备好点名
String PointName = "DDM.Apac.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;

//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);

Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第三步：注册事件
HisClient.Math_CorrelationCompleted += new EventHandler<
Apac.Client.HisService.Math_CorrelationCompletedEventArgs>
    (NetClient_Math_CorrelationCompleted);

//第三步：开始异步
mcbClient.Math_CorrelationAsync(SPointName, WPointName, PointName, stime, etime, 0, 100);

//接受事件
void NetClient_Math_CorrelationCompleted(object sender, Apac.Client.HisService.Math_CorrelationCompletedEventArgs e)
{
    if (e.Error == null)
    {
        //成功取得换回值
        MathData returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginMath\_Correlation(HisRowData[] values, System.AsyncCallback callback, object asyncState) 开始异步委托

EndMath\_Correlation(System.IAsyncResult result) 结束异步委托

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apac.Client.NetClient 中所提供的对应方法

### 3.5.11 UpdateAllLocalConfig

```
bool UpdateAllLocalConfig(Stream stream);
```

说明：更新全局配置信息。

PARAMETER	DESCRIPTION
<code>Stream</code>	计算服务全局配置信息流。

RETURN	DESCRIPTION
<code>bool</code>	True 提交更新成功, False 提交更新失败。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
String SPointName = "DDM.Apac.test";
String WPointName = "DDM.Apac.test1";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)

//第四步：调用
MathData returnvalue= HisClient.Math_Correlation(SPointName, WPointName, stime, etime, 0, 100);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好点名
String PointName = "DDM.Apac.test";
DateTime stime = DateTime.Now.AddHours(-24);
DateTime etime = DateTime.Now;
//第三步：根据点名获得实时服务客户端访问实体
PointRule rule = BaseConfig.GetPointRule(PointName);
Apac.Client.HisService.HistoryContractClient HisClient = client.GetHisClient(rule.ServiceName, rule.DirName)
//第三步：注册事件
HisClient.Math_CorrelationCompleted += new EventHandler<
Apac.Client.HisService.Math_CorrelationCompletedEventArgs>
(NetClient_Math_CorrelationCompleted);
//第三步：开始异步
mcbClient.Math_CorrelationAsync(SPointName, WPointName, PointName, stime, etime, 0, 100);
//接受事件
void NetClient_Math_CorrelationCompleted(object sender, Apac.Client.HisService.Math_CorrelationCompletedEventArgs e)
{
```

```

if (e.Error == null)
{
    //成功取得换回值
    MathData returnValue = e.Result;
}
else
{
    //网络异常
}
}

```



#### 相关方法（委托）：

System. [IAsyncResult](#) BeginMath\_Correlation([HisRowData\[\]](#) values, System. [AsyncCallback](#) callback, [object](#) asyncState) 开始异步委托

EndMath\_Correlation(System. [IAsyncResult](#) result) 结束异步委托

说明： System. [AsyncCallback](#) callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 [Apace.Client.NetClient](#) 中所提供的对应方法

### 3.5.12 UpdateIntervalConfig

```
bool UpdateIntervalConfig(byte[] stream, int Interval, string Language);
```

说明： 更新指定计算周期和语言的配置信息。

PARAMETER	DESCRIPTION
stream	全局配置信息流
Interval	计算周期
Language	脚本语言 “C#、VB.NET、JScript”

RETURN	DESCRIPTION
<a href="#">bool</a>	True 提交更新成功，False 提交更新失败。

### 3.5.13 GetIntervalConfig

```
Stream GetIntervalConfig(int Interval, string Language);
```

说明： 获得指定计算周期和语言的 Config 配置信息。

PARAMETER	DESCRIPTION
Interval	计算周期

Language	脚本语言 “C#、VB.NET、JScript”
----------	--------------------------

RETURN	DESCRIPTION
<code>Stream</code>	配置文件流。

### 3.5.14 GetIntervalPublicCode

```
string GetIntervalPublicCode(int Interval, string Language);
```

说明：得到指定的计算周期和脚本语言的公共代码段。

PARAMETER	DESCRIPTION
Interval	计算周期
Language	脚本语言 “C#、VB.NET、JScript”

RETURN	DESCRIPTION
<code>string</code>	公共代码段。

### 3.5.15 UpdataIntervalPublicCode

```
string UpdataIntervalPublicCode(string codeContext, int Interval, string Language);
```

说明：更新指定计算周期和脚本语言的公共代码段。

PARAMETER	DESCRIPTION
codeContext	公共代码段
Language	脚本语言 “C#、VB.NET、JScript”
Interval	计算周期

RETURN	DESCRIPTION
<code>string</code>	如果返回字符串为空表示编译成功，否则代码段编译异常。

## 3.6 告警服务(Apace 告警检测管理)



### 3.6.1 AddAlarm

```
string AddAlarm(Stream context);
```

说明：添加新的告警规则。

PARAMETER	DESCRIPTION
context	告警规则配置序列化数据流

RETURN	DESCRIPTION
string	Apace 服务进程运行信息。

### 3.6.2 DeleteAlarm

```
bool DeleteAlarm(int Interval, string AlarmCode);
```

说明：删除指定告警检测周期内对应的告警规则。

PARAMETER	DESCRIPTION
Interval	告警检测周期
AlarmCode	告警标示码

RETURN	DESCRIPTION
bool	True 删除成功，False 删除失败。

### 3.6.3 DeleteAlarmElement

```
bool DeleteAlarmElement(Stream context);
```

说明：删除对应的告警规则。

PARAMETER	DESCRIPTION
Interval	告警检测周期

RETURN	DESCRIPTION
--------	-------------

<code>bool</code>	True 删除成功, False 删除失败。
-------------------	------------------------

### 3.6.4 Contains

```
bool Contains(string AlarmCode, int Interval);
```

说明：检测对应的告警检测周期内是否存在有指定的告警标示码。

PARAMETER	DESCRIPTION
ServiceKey	服务进程名称，如：DDM.Apac.PCM [服务名].[目录名].[服务组件]
AlarmCode	告警标示码

RETURN	DESCRIPTION
<code>bool</code>	True 存在, False 不存在。

### 3.6.5 UpdateAlarm

```
string UpdateAlarm(Stream context);
```

说明：更新指定的告警规则。

PARAMETER	DESCRIPTION
context	告警规则配置序列化数据流

RETURN	DESCRIPTION
<code>string</code>	如果返回字符串为空表示成功，否则异常。

### 3.6.6 GetAllAlarm

```
Stream GetAllAlarm();
```

说明：得到所有的告警规则配置集合数据流。

RETURN	DESCRIPTION
<code>Stream</code>	告警规则配置数据流。



### 3.6.7 GetIntervalAlarm

```
Stream GetIntervalAlarm(int Interval);
```

说明：得到指定告警检测周期内的所有告警规则配置数据流。

PARAMETER	DESCRIPTION
Interval	告警检测周期

RETURN	DESCRIPTION
Stream	告警规则配置数据流。

### 3.6.8 GetAllAlarmMessage

```
Stream GetAllAlarmMessage();
```

说明：得到所有的已发生的告警信息数据流。

RETURN	DESCRIPTION
Stream	告警信息序列化数据流。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String Servicename="DDM"; String Dirname="Apac";
//第三步：告警服务客户端访问实体
Apac.Client.AlarmService.AlarmContractClient alarmClient = client.GetAlarmClient(Servicename, Dirname)

//第四步：调用
Stream returnvalue= alarmClient.GetAllAlarmMessage();
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String Servicename="DDM"; String Dirname="Apac";
//第三步：告警服务客户端访问实体
```

```

Apace.Client.AlarmService.AlarmContractClient alarmClient = client.GetAlarmClient(Servicename, Dirname)
//第三步：注册事件
AlarmClient.GetAllAlarmMessageCompleted += new EventHandler< Apace.Client.
AlarmService.GetAllAlarmMessageCompletedEventArgs>
(NetClient_GetAllAlarmMessageCompleted);
//第三步：开始异步
mcbClient.GetAllAlarmMessageAsync();
//接受事件
void NetClient_GetAllAlarmMessageCompleted(object sender,
Apace.Client.AlarmService.GetAllAlarmMessageCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



#### 相关方法（委托）：

System.IAsyncResult BeginGetAllAlarmMessage(string AlarmCode, System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetAllAlarmMessage(System.IAsyncResult result) 结束异步委托

GetAlarmMessagesForAlarmCode(string[] AlarmCodes)

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apace.Client.NetClient 中所提供的对应方法

### 3.6.9 GetAllLocalConfig

```
Stream GetAllLocalConfig();
```

说明： 得到告警服务全部配置数据流。

RETURN	DESCRIPTION
Stream	配置信息数据流

### 3.6.10 UpdateAllLocalConfig

```
bool UpdateAllLocalConfig(Stream stream);
```

说明：更新告警服务全局配置数据流。

PARAMETER	DESCRIPTION
<code>Stream</code>	配置信息数据流

RETURN	DESCRIPTION
<code>bool</code>	True 更新成功， False 更新失败。

### 3.6.11 GetAlarmMessage

```
Stream GetAlarmMessage(string AppCode);
```

说明：获得指定应用标示码下对应的所有已发生的告警信息数据流。

PARAMETER	DESCRIPTION
<code>AppCode</code>	应用标示码。

RETURN	DESCRIPTION
<code>Stream</code>	告警消息数据流。



#### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String AppCode = "testAppSystem"; String Servicename = "DDM"; String Dirname = "Apac";
//第三步：告警服务客户端访问实体
Apac.Client.AlarmService.AlarmContractClient mcbClient = client.GetAlarmClient(Servicename, Dirname)
//第四步：调用
Stream returnvalue = AlarmClient.GetAlarmMessageForAlarmCode(AppCode);
```



#### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String AppCode = "testAppSystem"; String Servicename = "DDM"; String Dirname = "Apac";
//第三步：获得实时服务客户端访问实体
```

```

Apave.Client.AlarmService.AlarmContractClient mcbClient = client.GetAlarmClient(Servicename, Dirname)
//第三步：注册事件
AlarmClient.GetAlarmMessageForAlarmCodeCompleted += new EventHandler< Apave.Client.
AlarmService.GetAlarmMessageForAlarmCodeCompletedEventArgs>
(NetClient_GetAlarmMessageForAlarmCodeCompleted);
//第三步：开始异步
mcbClient.GetAlarmMessageForAlarmCodeAsync(AlarmCode);
//接受事件
void NetClient_GetAlarmMessageForAlarmCodeCompleted(object sender,
Apave.Client.AlarmService.GetAlarmMessageForAlarmCodeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}

```



#### 相关方法（委托）：

System.IAsyncResult BeginGetAlarmMessageForAlarmCode(string AlarmCode, System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetAlarmMessageForAlarmCode(System.IAsyncResult result) 结束异步委托

GetAlarmMessagesForAlarmCode(string[] AlarmCodes)

说明： System.AsyncCallback callback 回调方法

注意： 此实例只是为演示获取历史数据的形式，建议在实际的调用过程中请使用 Apave.Client.NetClient 中所提供的对应方法

### 3.6.12 GetAlarmMessageForAlarmCode

```
Stream GetAlarmMessageForAlarmCode(string AlarmCode);
```

说明： 获得指定告警规则码对应的告警信息数据流。

PARAMETER	DESCRIPTION
AlarmCode	告警标示码

RETURN	DESCRIPTION
Stream	告警信息序列化数据流。



### 实例演示（同步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String AppCode = "Alarm001"; String Servicename = "DDM"; String Dirname = "Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetAlarmClient(Servicename, Dirname)
//第四步：调用
Stream returnvalue= AlarmClient.GetAlarmMessageForAlarmCode(AlarmCode);
```



### 实例演示（异步）

```
//第一步：得到客户端实体
Apac.Client.NetClient client = Apac.Client.NetClientManage.GetNetClient();
//第二步：准备好查询条件和网络名
String AlarmCode = "Alarm001"; String Servicename = "DDM"; String Dirname = "Apac";
//第三步：获得实时服务客户端访问实体
Apac.Client.McbService.RealTimeContractClient mcbClient = client.GetAlarmClient(Servicename, Dirname)
//注册事件
AlarmClient.GetAlarmMessageForAlarmCodeCompleted += new EventHandler< Apac.Client.
AlarmService.GetAlarmMessageForAlarmCodeCompletedEventArgs>
(NetClient_GetAlarmMessageForAlarmCodeCompleted);
//开始异步
mcbClient.GetAlarmMessageForAlarmCodeAsync(AlarmCode);
//接受事件
void NetClient_GetAlarmMessageForAlarmCodeCompleted(object sender,
Apac.Client.AlarmService.GetAlarmMessageForAlarmCodeCompletedEventArgs e) {
    if (e.Error == null)
    {
        //成功取得换回值
        Stream returnValue = e.Result;
    }
    else
    {
        //网络异常
    }
}
```



### 相关方法（委托）：

System.IAsyncResult BeginGetAlarmMessageForAlarmCode(string AlarmCode, System.AsyncCallback callback, object asyncState) 开始异步委托  
EndGetAlarmMessageForAlarmCode(System.IAsyncResult result) 结束异步委托

GetAlarmMessagesForAlarmCode([string](#)[] AlarmCodes)

说明: System.[AsyncCallback](#) callback 回调方法

注意: 此实例只是为演示获取历史数据的形式, 建议在实际的调用过程中请使用 `Apace.Client.NetClient` 中所提供的对应方法