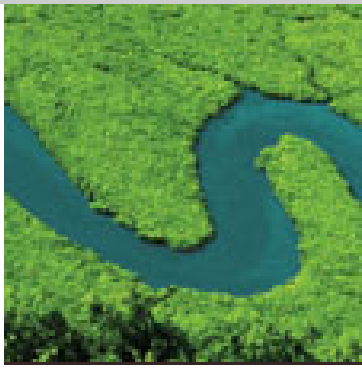
OPC
DATA ACESS



OPC
SIMATIC NET

Highlights of the OPC

SIEMENS

OPC DA 概述

OPC
DATA ACESS

OPC DA 规范主要用于读取过程变量

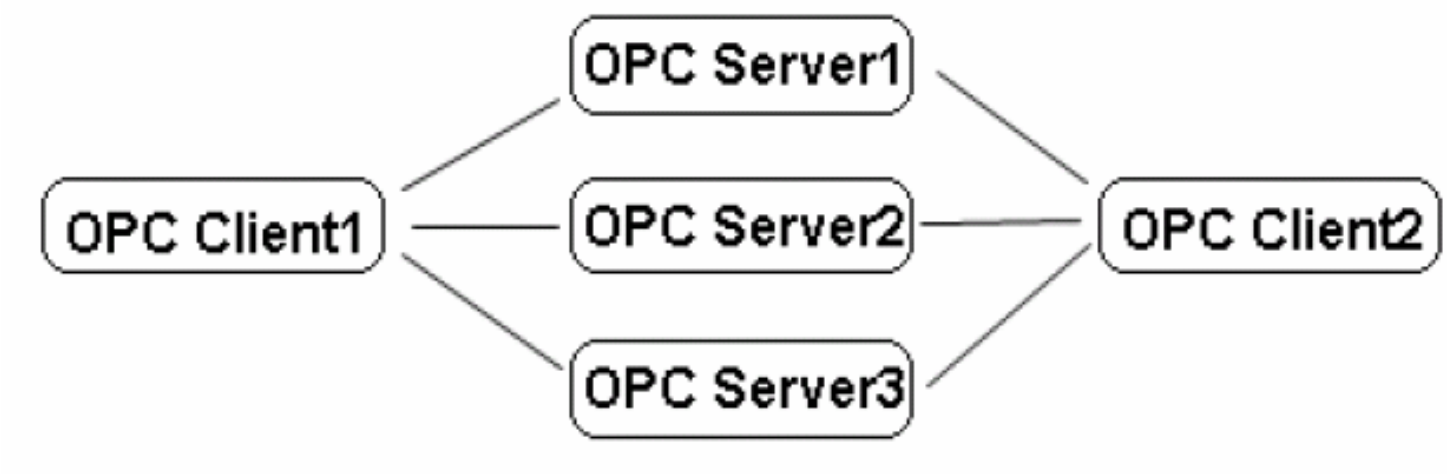
该规范制定了OPC 服务器和OPC 客户程序的COM 接口标准，通过制定标准的接口来实现多个厂家的OPC 服务器和OPC 客户程序开发

- 读取一个或多个过程变量
- 修改一个或多个过程变量
- 监控一个或多个过程变量.

SIEMENS

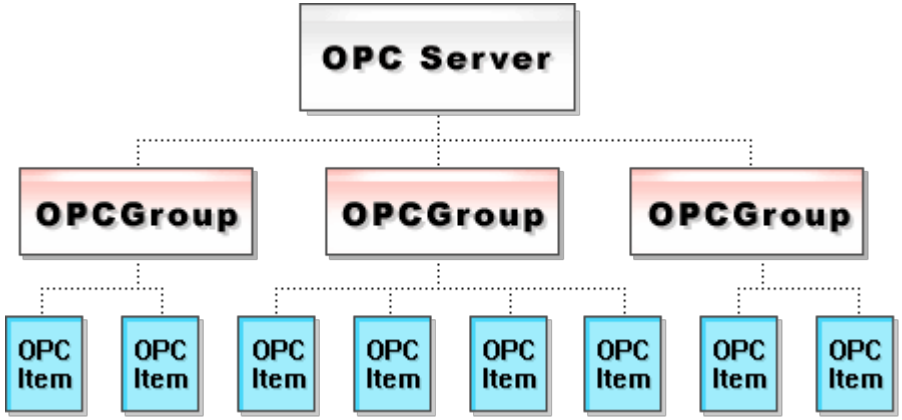
客户机服务器

一个OPC CLIENT 可以连接一个或多个OPC 服务器，而多个OPC CLIENT 也可以同时连接一个OPC 服务器，如图1.1 所示



服务器

在高层, 一个OPC 服务器由这几个对象组成: the server, the group, and the item.



这里最需要注意的是项并不是数据源，项代表了到数据源的连接。

OPC 接口体系

OPC
DATA ACESS

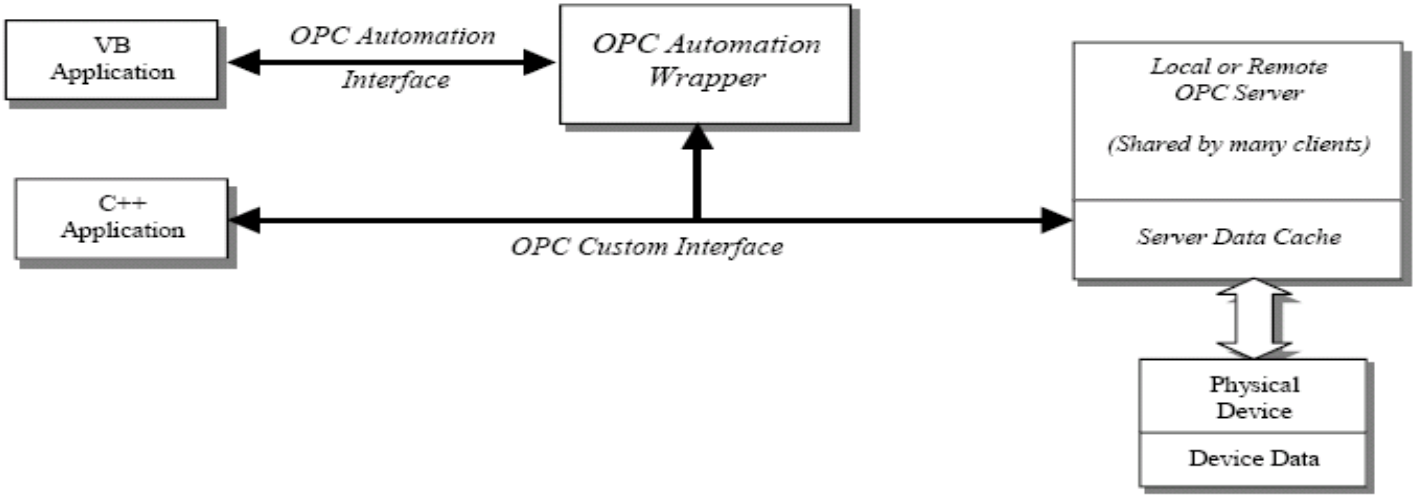
OPC 规范提供两种接口：自定义接口（the OPC Custom Interfaces），自动化接口（the OPC Automation interfaces）

.

一个OPC客户应用可以通过OPC服务器题供的定制接口或自动化接口与服务器通讯. 定制接口是OPC服务器必须完成的而自动化接口是可选的.

OPC 结构和组件

如前所述，象所有的COM 结构一样，OPC 是C/S 结构，OPC 服务器提供标准的接口供OPC 客户程序访问。OPC 服务器必须提供自定义接口，对于自动化接口，在OPC 规范定义中是可选的。自动化接口的发布使得像Viaual Basic类似的语言通过一个动态的DLL库文件（包装器）来间接的访问定制接口是可选的。



OPC 对象接口定义

OPC
DATA ACESS

- OPC 服务器对象提供一些方法去读取或连接一些数据源。

- OPC 客户程序连接到OPC 服务器对象，并通过标准接口与OPC 服务器联系。

- . OPC服务器对象提供接口供OPC 客户程序创建组对象并将需要操作的项添加到组对象中，并且组对象可以被激活，也可以被赋予未激活状态。

- .对于OPC 客户程序而言，所有OPC 服务器和OPC 组对象可见的仅仅是COM 接口。

OPC 对象接口定义

下面的两个图例是OPC 规范中定义的OPC 服务器对象和OPC 组对象的COM 接口，其中任选的接口均以[ ]表示。（注：任选指开发OPC 服务器时，这些接口可以根据实际情况选择实现还是不实现。）

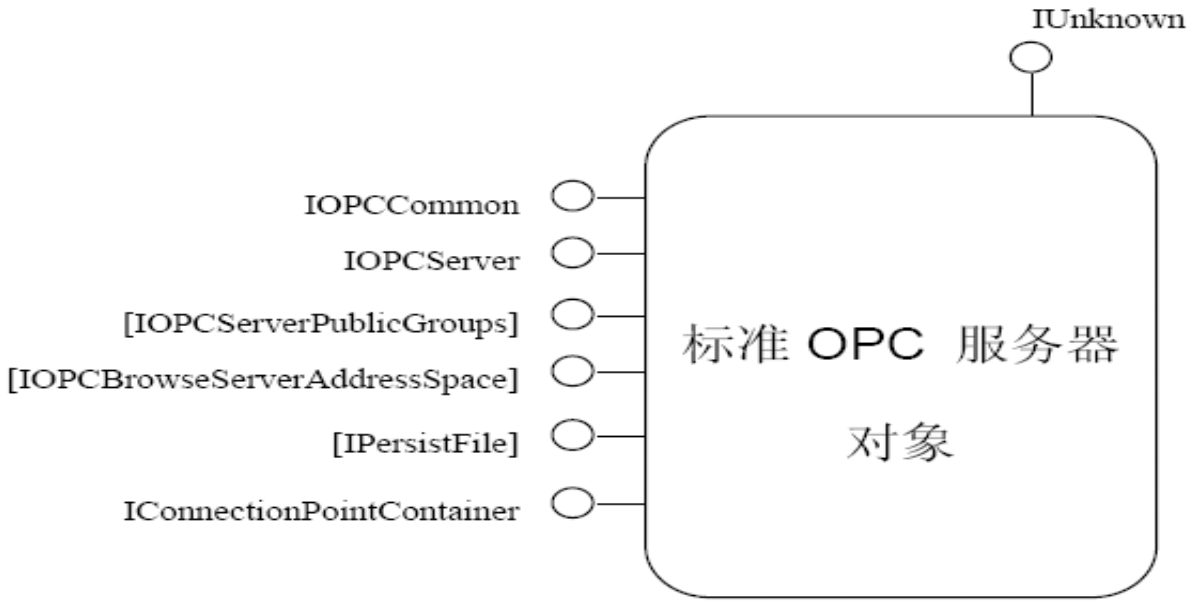


图 1.5-标准 OPC 服务器对象及接口

OPC 对象接口定义

IOPCCommon

通过该接口可为某个特定的客户/服务器对话（session）设置和查询本地标识（LocateID）。这样，一个客户程序的操作将不会影响其它客户程序。

IOPCServer

接口及成员函数主要用于对组对象进行创建、删除、枚举和获取当前状态等操作。是OPC 服务器对象的主要接口。

IConnectionPointContainer

支持可连接点对象，当OPC 服务器关闭时需要通知所有的客户程序释放OPC 组对象和其中的OPC 组员，此时可利用该接口调用客户程序方的IOPCShutdown 接口实现服务器的正常关闭。

OPC 对象接口定义

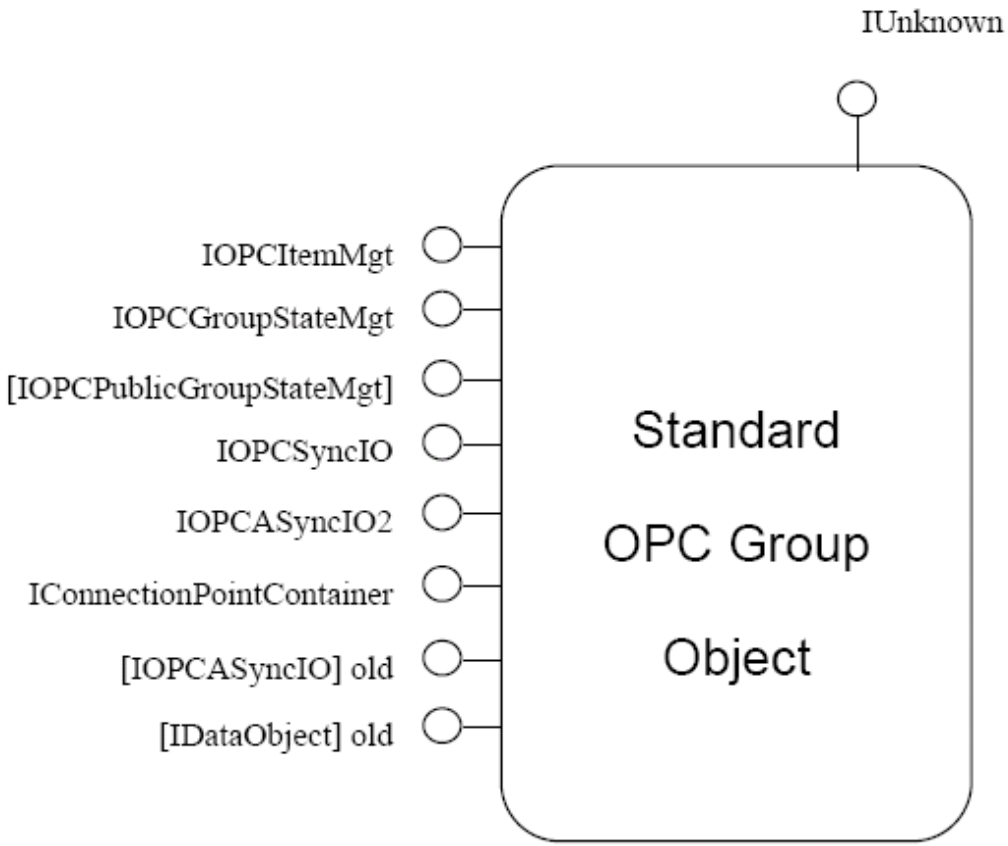


图 1.6 标准 OPC 组对象及接口

OPC 对象接口定义

IOPCItemMgt

接口及成员函数用于OPC 客户程序添加、删除和组对象中组员等控制。

IOPCGroupStateMgt

接口及成员函数允许OPC客户程序操作或获取用户组对象的全部状态（主要是组对象的刷新率和活动状态）

IOPCSyncIO

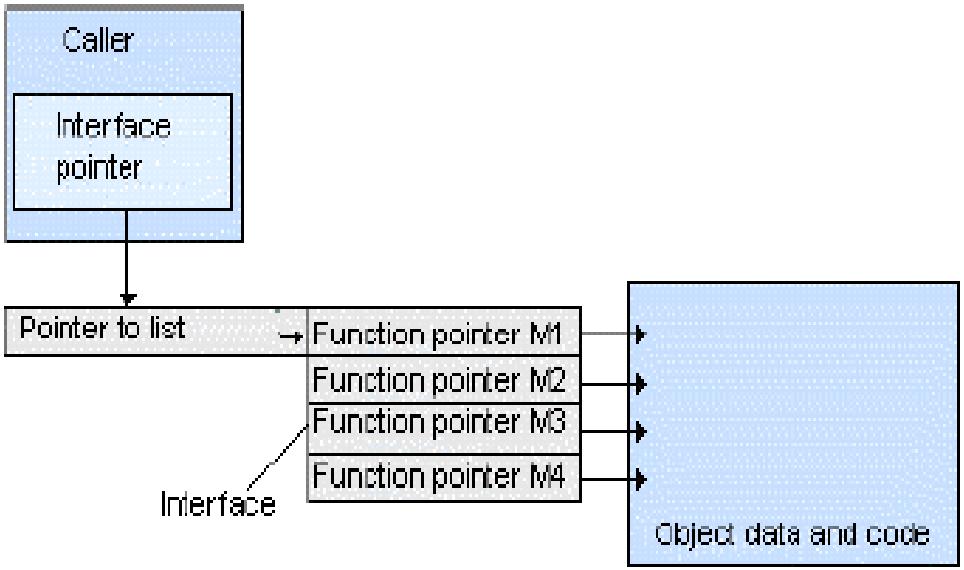
用于同步数据访问。

IOPCAsyncIO2

用于异步数据访问。

对象和接口

下面为一个接口的基本结构



对象和接口

OPC
DATA ACESS

IOPCServer

HRESULT **AddGroup**(szName, bActive, dwRequestedUpdateRate, hClientGroup, pTimeBias, pPercentDeadband, dwLCID, phServerGroup, pRevisedUpdateRate, riid, ppUnk) 新增一个 OPCGroup

HRESULT **GetErrorString**(dwError, dwLocale, ppString) 取得错误信息字符串

HRESULT **GetGroupByName**(szName, riid, ppUnk) 依据名称取得 OPCGroup 的介面

HRESULT **GetStatus**(ppServerStatus) 取得 OPCServer的状态信息

HRESULT **SetClientName** (szName)设定Clinet的名称

RemoveGroup(hServerGroup, bForce)移除一个OPCGroup

HRESULT **CreateGroupEnumerator**(dwScope, riid, ppUnk) 产生一个 OPCGroup 列举器

定制接口的编程方法

OPC 服务器是包含了一个或多个coclass的二进制（DLL或EXE）。

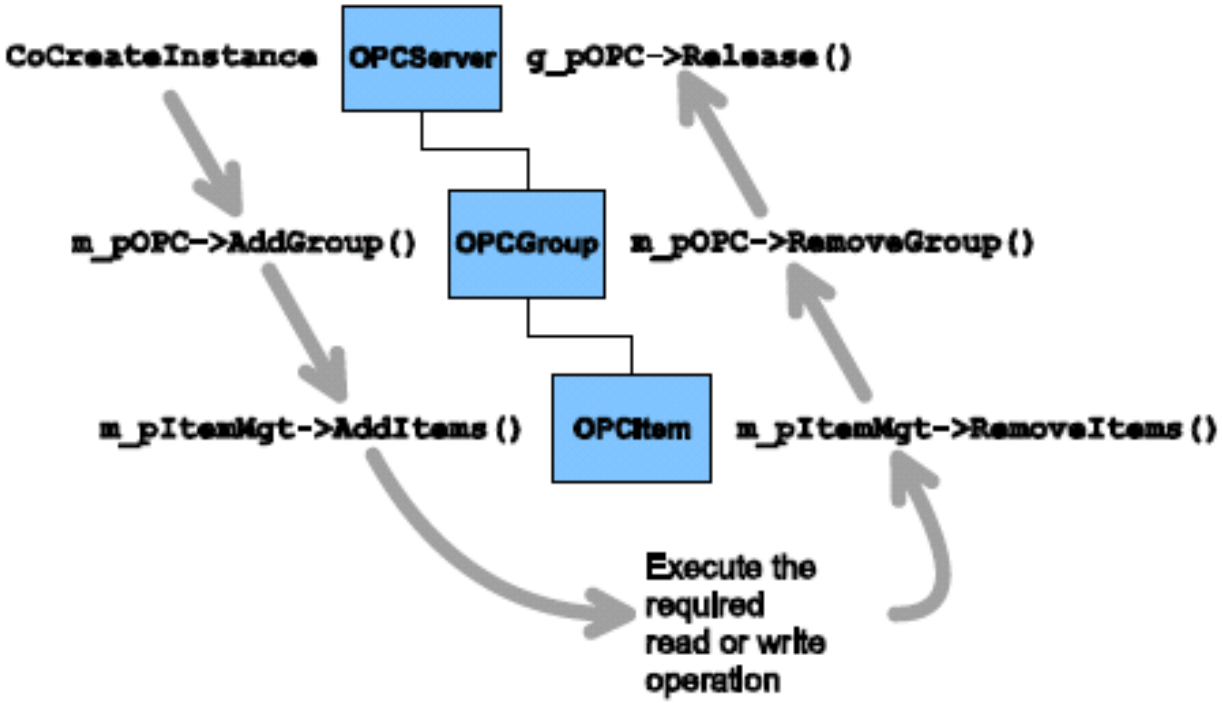
CLSID
每个COM 类有一个唯一的识别号，是一个128位的数字，用CLSID操作系统可以找到相应的DLL或者是可执行文件用来完成类的执行，如果一个客户想要用来对象，就必须指定它的。

ProgID
为了识别OPC 服务器必须分配几ClSID给一个可读的ProgID，CLSID和ProgID是由服务器的供应商来制定的.

定制接口的编程方法

OPC
DATA ACESS

定制接口编程实现过程



定制接口的编程方法

OPC
DATA ACESS

步1：

如果程序要调用COM 库的某一个函数必须先登录COM 函数CoInitialize()可
以完成此功能从函数CoGetMalloc()可以得到一个指向COM内存管理接口的指
针

```
HRESULT rl;
rl=CoInitialize(NULL);
rl=CoGetMalloc(MEMCTX_TASK,&g_pIMalloc);
```

步2：

每个COM 服务器有一个ProgID 通过它可以得到一个全球唯一的CLSID用
CLSIDFromProgID()函数可以实现这个变换ProgID 用变量szName进行参数传
递在示例程序中它的值是L"OPC.SimaticNET"

```
m_Popc
rl=CLSIDFromProgID(szName,&clsid);
```

```
r1 = CLSIDFromProgID(L"OPC.SimaticNET", &clsid);
```

SIEMENS

定制接口的编程方法

OPC
DATA ACESS

步3：

CoCreateInstance()函数创建一个类实例其CLSID 值设定如下:

```
r2=CocreateInstance(clsid,NULL,CLSCTX_LOCAL_SERVER,IID_IUnkown,
(void**)&Punk);
```

这段程序的结果是一个指向服务器对象IUnknown 接口的指针变量pUNK

步4：

IOPCServer 接口的AddGroup()方法可以创建OPC 组

```
m_pItemMgt)
HRESULT r1;
r1=m_Popc->AddGroup(szName,TRUE,500,1,&TimeBias,
&PercDeadband,dwLCID,&m_GrpServerHandle,
&RevUpRate,IID_IOPCItemMgt,
(LPUNKNOWN*)&m_pItemMgt);
```

这段程序的执行结果是创建一个有指定名称和属性的组在返回的参数中有一个指向所需要的进程组对象接口的指针在这里是IOPCItemMgt(变量

SIEMENS

定制接口的编程方法

OPC
DATA ACESS

步5：

IOPCItemMgt 接口有AddItem()方法可以创建OPC 项

HRESULT r1;

r1=m_pItemMgt->AddItems(NumItems,pItems, &m_pItResult,&pErrors);

这段程序的结果是创建有特殊属性的指定数量的项除此之外事件结构变量

m_pItResult 服务器句柄目标系统上的项数据类型等也被赋值

用于执行所需操作的指针需要通过现有的指向IOPCItemMgt 接口的指针得到如

如果用户要进行异步通信就需要指向IOPCAsyncIO 接口的指针

HRESULT rl;

r1=m_pItemMgt->Queryface(IID_IOPCAsyncIO, (void * *) & pAsyncIO);

通过该接口的Read() 和Write()两个方法可以读写项的数值

r2=pAsyncIO->Read(m_dwConnection,OPC_DS_CACHE,dwNumItems,phServer, &m_TransactionID,&pErrors);

这段程序的执行结果是OPC 项的数据被送到客户程序的IAdviseSink 接口

SIEMENS

定制接口的编程方法

OPC
DATA ACESS

步6：

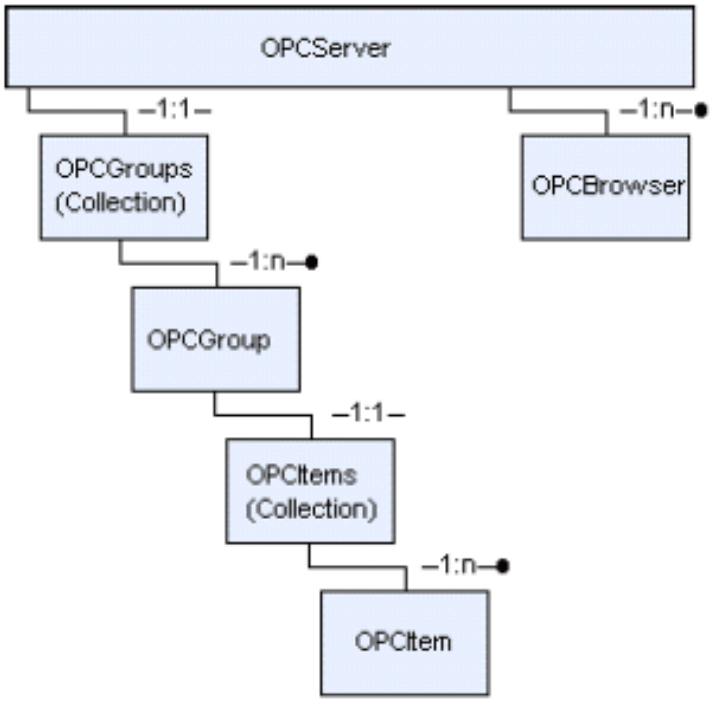
在程序停止运行之前必须删除已创建的OPC 对象并释放内存到目前为止用到的各种接口都有相应的函数

```
5.4 r1=m_pItemMgt->RemoveItems(dwNumItems,phServer,&pErrors);
r1=m_Popc->RemoveGroup(m_GrpServerHandle,TRUE);
m_pItemMgt->Release();
m_pOPC->Release();
```

SIEMENS

自动化接口中的对象模型

对象模型

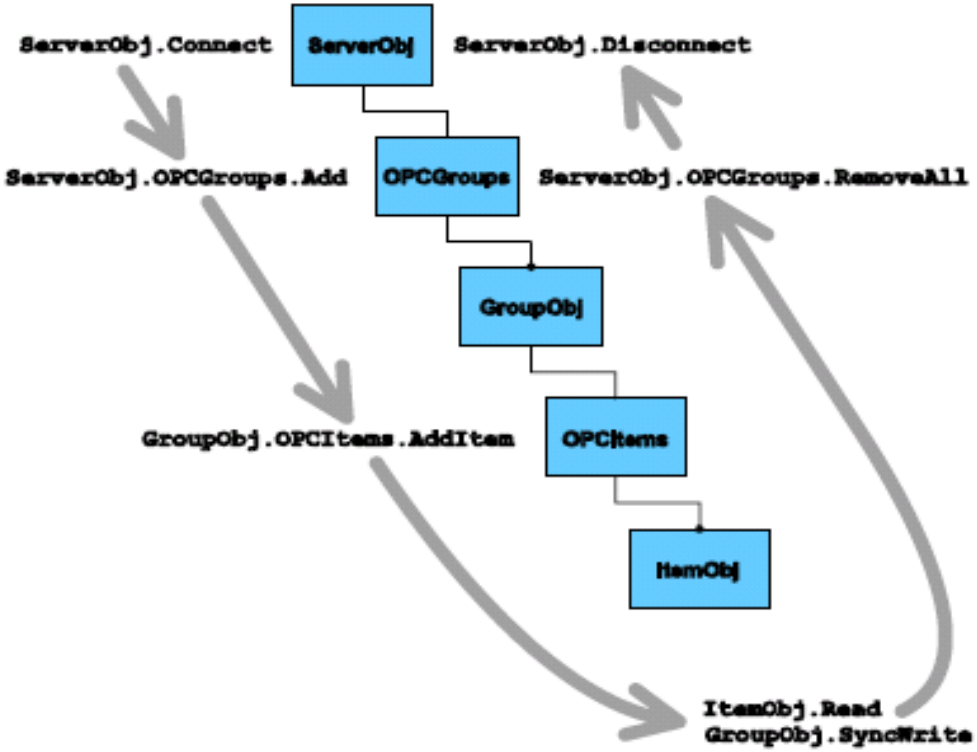


Object Model of OPC Data Access

自动化接口的编程方法

OPC
DATA ACESS

编程步骤



自动化接口的编程方法

OPC
DATA ACESS

步1：初始化SERVER

```
Set Svr=CreateObject(ProgID)
Set Svr=CreateObject(ProgID)
```

步2：添加组

```
Set Group=Svr.AddGroup "szTaoYanBinGRP" false
RateRequested,GroupClientHdl, PercentDeadband,
dwLangId_ENGLISH,GroupServerHdl, RateRevised

Set PtrItemMgt =Group
```

步3：添加项

```
ItemsActive(0)= True:
 ItemsActive(1)= True
PtrItemMgt.AddItems NbrItems,
temsIDs,ItemsActivity,
ItemsClientHdls,ItemsSvrHdls, ItemsErrors,
ItemsObjects, accessPath
```

SIEMENS

自动化接口的编程方法

OPC
DATA ACESS

步4：读写项

```
Set ptrSyncIO = group
        WHILE (true)
        PtrSyncIO.OPCRead  OPC_DS_DEVICE, NbrItems,
ItemsSvrHdls,pValues, pQualities, pTimeStamps, ItemsErrors
        WEND
```

步5：释放组和项

```
Svr.RemoveGroup GroupServerHdl, False
        Set Group = Nothing
        Set Svr = NothingS
```

Automation接口与Custum接口的不同

对于客户端应用程序的开发：

- 采用自定义接口的方式运行效率高，但开发难度大
- 采用自动化接口的方式运行效率低，但开发简单
- 自定义接口是一组COM接口，主要用于采用C++语言的应用程序开发；
- 自动化接口是一组OLE接口，主要用于采用VB，DELPHI等基于脚本编程语言的应用程序开发。
- 自定义接口是服务商必须提供的，而自动化接口则是可选的
- OPC基金会提供了一个“自动化包装器”的动态连接库，用于在两者间转换

Automation接口与Custum接口的不同

不同语言的特性

Criterion	VBA (MSOffice)	VB V6.0	VC++	.NET-languages (managed)
Custom interface	-	-	✓	✓ (with RCW)
Automation interface	✓	✓	✓ (*)	✓ (*) (with RCW)
Simple implementation	++	++	- (**)	+ (**)
Error handling	-	-	+ (**)	++ (**)
Performance	-	-	++ (**)	+ (**)

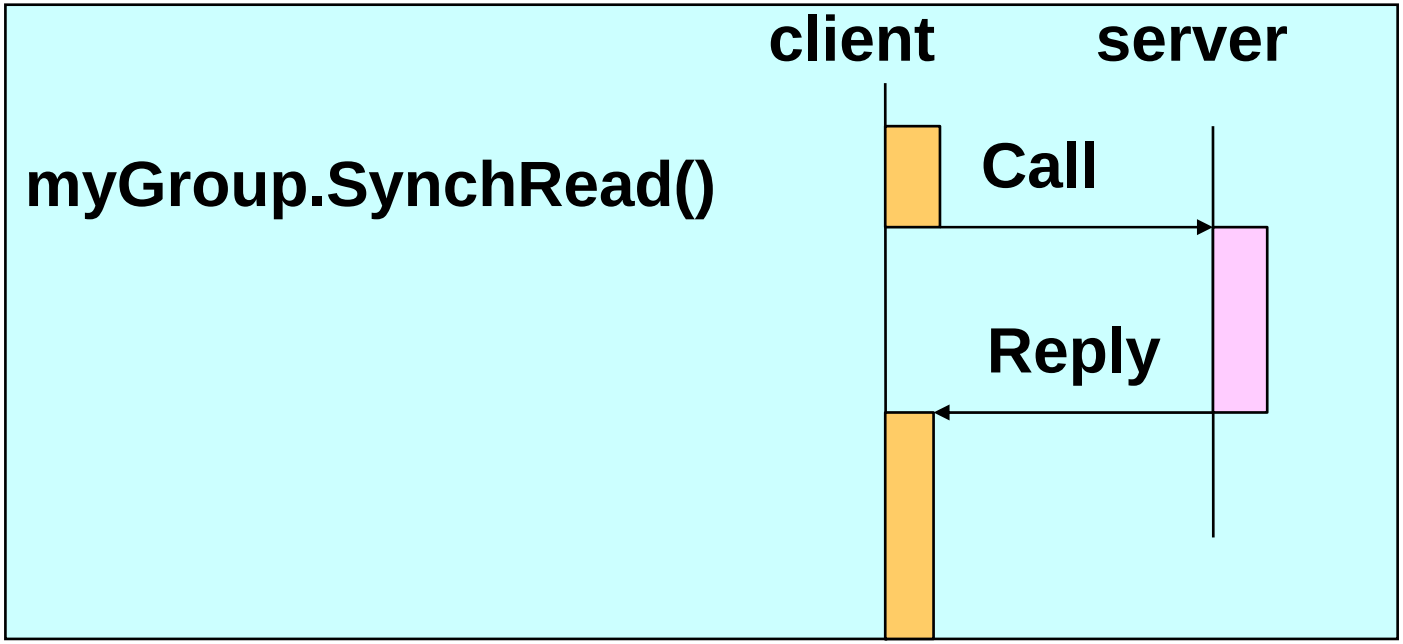
(*) should not be used

(**) using the custom interface

同步访问与异步访问的不同

OPC
DATA ACESS

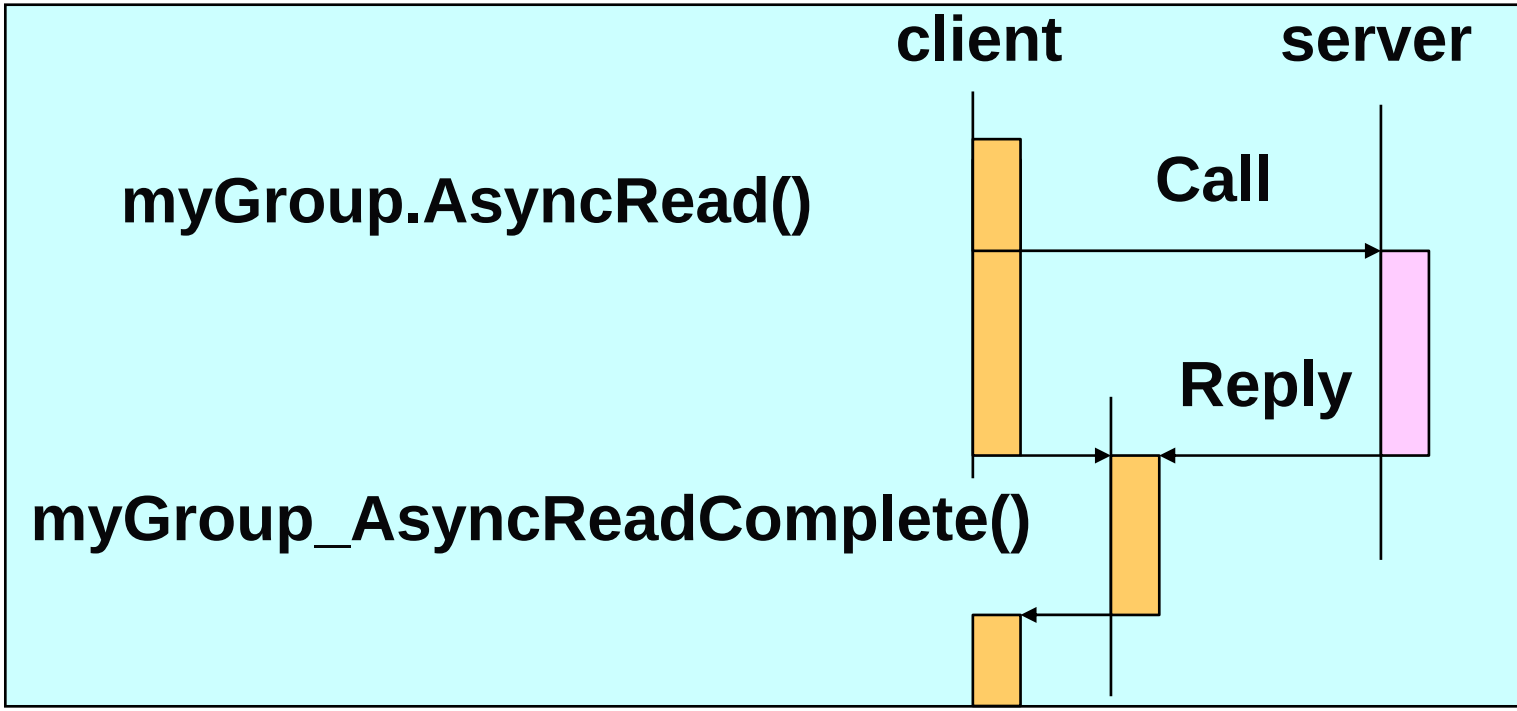
DA 同步读



同步访问与异步访问的不同

OPC
DATA ACESS

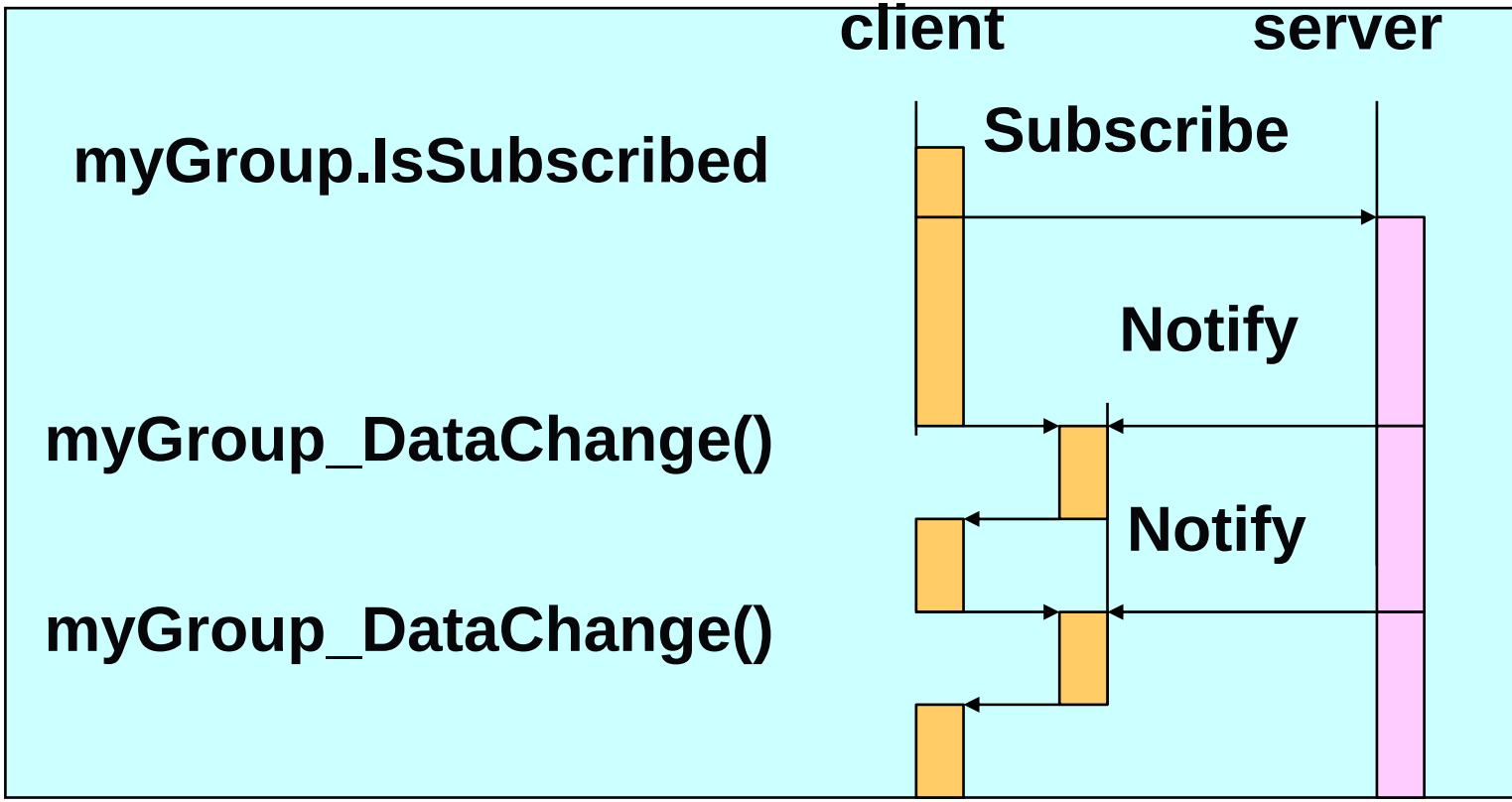
DA 异步读



同步访问与异步访问的不同

OPC
DATA ACESS

DA 数据订阅



同步访问与异步访问的不同

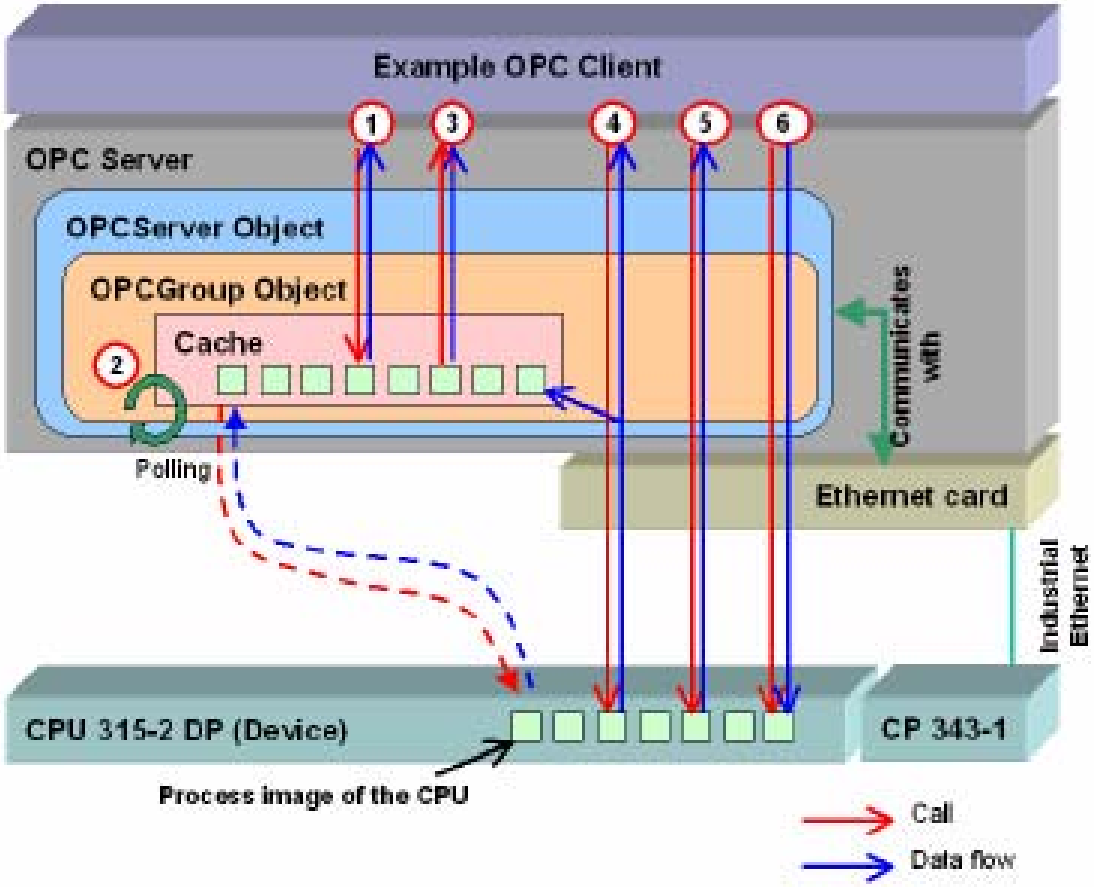
- 同步方式实现较为简单，客户向服务器发出读写请求，然后等待服务器返回信息，当客户数据较少而且同服务器交互的数据量比较少的时候可以采用这种方式，然而当网络堵塞或大量客户访问时，会造成系统的性能效率下降。

- 异步方式实现较为复杂，客户向服务器发出读写请求后，服务器立刻返回信息表示请求已接受，客户可以进行其他处理，当服务器完成读写操作后，通过调用回调函数，通知客户程序操作完成，并传递相应的信息，因此异步方式的效率更高，能够避免多客户大数据请求的阻塞，并可以最大限度地节省CPU和网络资源。

- 另外一种异步方式是服务器周期性的扫描缓冲区的数据，发现数据变化范围超过死区后，立刻通知客户程序，传递相应信息。

读Cache 与 Device 的不同

同步与异步的读写任务：



读Cache 与 Device 的不同

- Cache是在OPC Server上产生的零时的缓冲区，对于一个特定的组，在其里面包含了过程变量的本地映像区（这就组对象管理的项）

- DEVICE是指控制器中的过程变量.

读Cache 与 Device 的不同

客户可以选择从CACHE读数据或从DEVICE读数据。

- 由于直接从数据提供设备讯取数据速度比较慢，所以为了提高数据读取速度，OPC服务器按照一定的刷新率把数据读进一个数据缓存区(CACHE)，当客户需要数据时，可以直接从数据缓存区该数据缓存区(即CACHE)读取数据。

- 直接从设备(DEVICE)读取数据主要用在一些诊断操作或一些关键操作中。

访问Cache 与 Device

对CACHE和DEVICE访问

Operation	DEVICE	CACHE	Note
Synchronous reading	✓	✓	Reading from the CACHE requires the group and the respective item to be active.
Asynchronous reading	✓	-	The OPV server reads the variable from the DEVICE. This also updates its CACHE.
Synchronous writing	✓	-	Write jobs are always written to the DEVICE.
Asynchronous writing	✓	-	Write jobs are always written to the DEVICE.
Monitoring (event controlled)	-	✓	Monitoring of variables is only possible using the CACHE.

OPC server的优化的读取方案

对OPC server的读取有四种优化的方案

- 对组对象进行操作
- 访问OPC server 的缓冲区OPC Cache
- 对项对象的优化访问
- 利用缓冲区的Send/Receive服务

OPC server的优化的读取方案

OPC
DATA ACESS

对组对象进行操作

在众多的方法中, 可以通过一次函数调用来完成大量的过程的数据的传输.通过组操作大量减少了函数的调用，这样通OPC client和OPC server仅有几次调用，这种组操作能够在网络上自身优化通讯，如果要通过Dcom访问远程的Server，这种操作方式特别有效可能在网络上只有一个功能函数的调用.

注意

当利用数据订阅方式也常用到这种方式

Examples of quantity calls with OPC Data Access:

IOPCItemMgt::AddItems	Combining many OPC items to a group
	TIP: Calling AddItems is faster if the group is NOT active. Don't activate the group until all OPC items have been added.
IOPCSyncIO::Read	Synchronous reading of value, time stamp and quality of many OPC items.
IOPCSyncIO::Write	Synchronous writing of value, time stamp and quality of many OPC items.

SIEMENS

OPC server的优化的读取方案

OPC
DATA ACESS

访问OPC server 的缓冲区OPC Cache

OPC cache是OPC server中的中间缓冲区，在它里面存放着OPC项的最后一次获得的值. OPC server不断更新着在激活组里的激活项.假使在Cache中的项的值是有效的，即项被成功读取.那么从Cache的读取要比从网路的读取要快的多. 如果cache种的值在足够短的时间更新的话，对于一般的应用建议采用这种方法. 对于Cache的刷新速率是由*UpdateRate来设定的*

Examples of services that can be used on the cache:

IOPCSyncIO::Read	Synchronous reading of value, time stamp and quality of many OPC items from the cache.
IOPCAsyncIO::Refresh	A callback is generated for all active OPC items, regardless of value, and the value stored in the cache is returned.

SIEMENS

OPC server的优化的读取方案

OPC
DATA ACESS

对项对象的优化访问

在下列协议的变量服务，SIMATIC NET 的OPC Server提供了一种优化的算法:
S7 协议
基于工业以太网的S5的兼容通讯
当有几个单独的任务同时访问变量时会转换成一个任务以数组的方式访问对方设备.这样就减少了在网路上传输的数据包, 提高包的利用率.
这中算法可以用在读和写而且是缺省的设置.对于 OPC client 这种优化的算法是不可见的.

Example: Organization of a data block for read access

DB10,W10	Executed by the communication system as one read field access to DB10,B10,12. (Please note that this is converted to a read job despite the gaps at bytes 18 and 19. The “superfluously” read data are rejected.)
DB10,B12	
DB10,B13	
DB10,DW14	
DB10,W20	

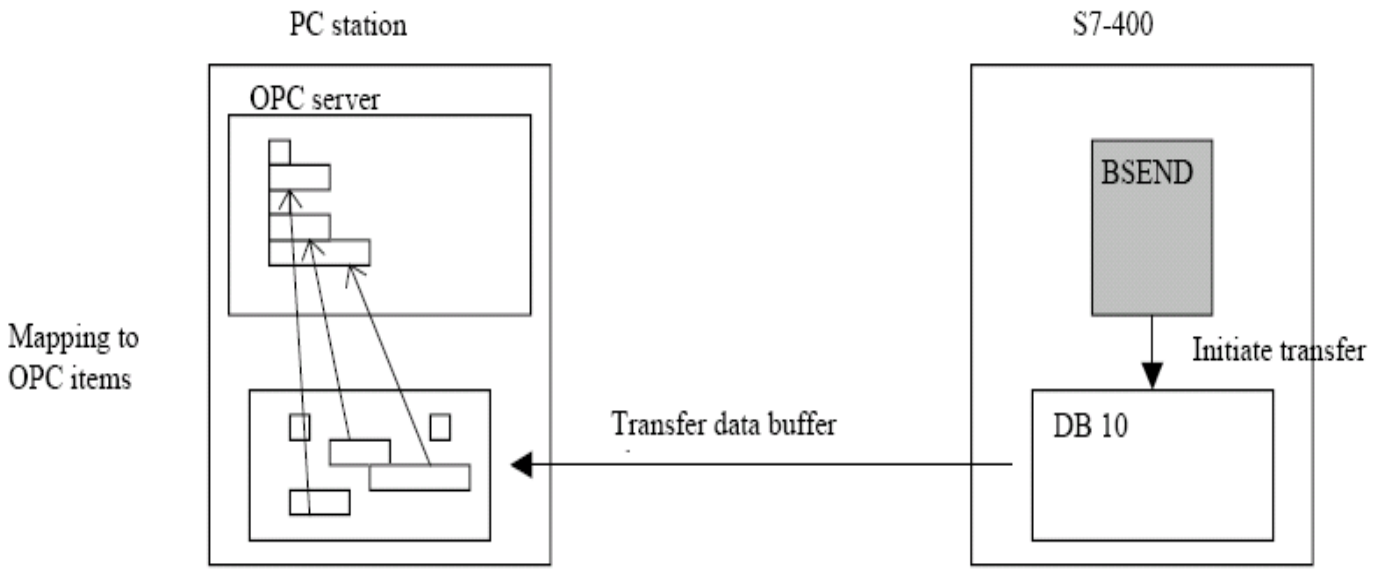
SIEMENS

OPC server的优化的读取方案

OPC
DATA ACESS

利用缓冲区的Send/Receive服务

在传输大数据量的包时, 基于以太网的S7 通讯和 S5的兼容通讯提供了面像缓冲的 send/receive 服务. 数据包在通讯的双方发送. 这种仅当初始发送任务的时候使有网路的负载. 当使用SIMATIC NET的OPC Server, 可以结构化数据块. 这样可以把OPC项分成几个独立的部分.



OPC server的优化的读取方案

OPC
DATA ACESS

在OPC CIIENT端应用适当的访问方法

对于DATA ACESS规范OPC接口提供了下列的数据访问的方法：

- Synchronous read/write
- Asynchronous read/write
- Monitoring of variables

根据不同的情况选择合适的访问方法

SIEMENS