

Unified Automation C++ SDK & Tools

测试报告

1. 各方 OPC UA SDK 概述

1.1 OPC 基金会

OPC Foundation 提供 C 语言的 UA Stack SDK , 以及基于 .Net 框架的 UA Client/UA Server SDK。同时基金会还提供 Global Discovery Server, Local Discovery Server 以及 Server Test Tool 等工具。

UA Ansi C Stack SDK 提供以下内容：

- 提供服务端与客户端通讯协议栈接口，包括 Security Channel , Endpoint , Session , Serialization , Discovery Service Interface, 报文收发接口等。
- SDK 预定义了各种 Request/Response (读、写、订阅等) 的结构体，并提供对应服务的 Callback 注册机制。
- 提供 UA 规格中定义的数据类型，其中包括 OpcUa_Int32 , OpcUa_String 等基本数据类型，还包括如 OpcUa_Node , OpcUa_Reference , OpcUa_Variable 等与地址空间相关的数据类型。
- 提供基于 SSL 的安全相关接口，如证书管理、公密钥加密等

UA .Net SDK 除了提供上述接口外，还提供以下模块

- Node Manager 地址空间 Interface
- UA Client/Server 模块

OPC Foundation 提供的 UA Stack SDK 仅仅为开发者提供了安全、统一的通讯方法，但并未提供地址空间管理与服务的具体实现；在 UA .NET SDK 中，OPC Foundation 提供了 Node Manager Interface 用于管理地址空间，并通过 UA Server 将 UA Stack 与 UA Node

Manager 进行统一管理，从而完整的实现了内容、服务与通讯的一体化。但 OPC Foundation 的 .NET SDK 在跨平台与运行效率方面存在着局限性，因此并不适用于作为大型服务器的开发环境。

1.2 Unified Automation

Unified Automation 提供基于 OPC UA 标准的跨平台 C++ SDK，该 SDK 有以下特点：

- 封装性：对 OPC Foundation 提供的 Stack SDK 进行了封装。
- 一致性：该 C++ SDK 是基于 OPC Foundation 的 Stack SDK 接口实现的，因此支持 UA Stack 所定义的通讯协议与数据类型。
- 跨平台：该 C++ SDK 支持 Windows 与 Linux 系统
- 完整性：除了对 UA Stack 进行了封装外，该 C++ SDK 还提供了类似于 OPC Foundation UA .NET SDK 中所提供的 NodeManager 与 Server/Client SDK 模块。同时，该 SDK 的 Professional 版本还支持 Client redundancy 等 OPC Foundation 尚未提供的功能。

Unified Automation 提供以下工具：

- UA Modeler：用于地址空间建模，并生成 OPC UA 规范中定义的 XML 格式文件。
- UA Expert: 用于发现 UA 服务器，浏览服务器的地址空间，调用地址空间的方法以及测试服务器的性能。

Unified Automation 的 C++ SDK 已得到 OPC Foundation 的认证，具体信息如下：

- 产品信息：

<http://opcfoundation.org/Products/ProductDetails.aspx?CM=1&RI=8927&CU=22>

- Compliance Test：

<http://opcfoundation.org/Products/CompanyComplianceTests.aspx?CM=1&RI=8927&CN=COM&CI=902&CU=5>

- 认证信息：



1.3 Prosys

Prosys 是一家工控领域的服务型公司，它提供的 OPC UA JAVA SDK 的类命名与类结构和 Unified Automation 的 C++ SDK 极为相似，同时该公司还是 Unified Automation C++ SDK 的 Reseller，以及 C++ SDK 售后服务的提供者，这使我们不得不推测其 JAVA SDK 就是一个将 Unified Automation C++ SDK 接口映射为 JAVA 接口的产品。

由于与 Unified Automation C++ SDK 的相似性，这里就不对其进行详细介绍了。

1.4 Softing

Softing 提供的 UA C++ SDK 在跨平台性、功能完整性方面与 Unified Automation 相当，但 Softing 的 C++ SDK 的类层次结构略显简单，且对 UA Stack SDK 采用了 High level 的封装方法，完全抹去了底层 UA Stack 库的痕迹。换句话说，Unified Automation 采用的方法就好像是 STL 对 C 标准库的封装，而 Softing 采用的是类似于 MFC 封装 C 接口的方法。

在工具支持方面，Softing 缺少像 UA Expert 与 UA Modeler 这样的工具。

1.5 SDK 供应方列表

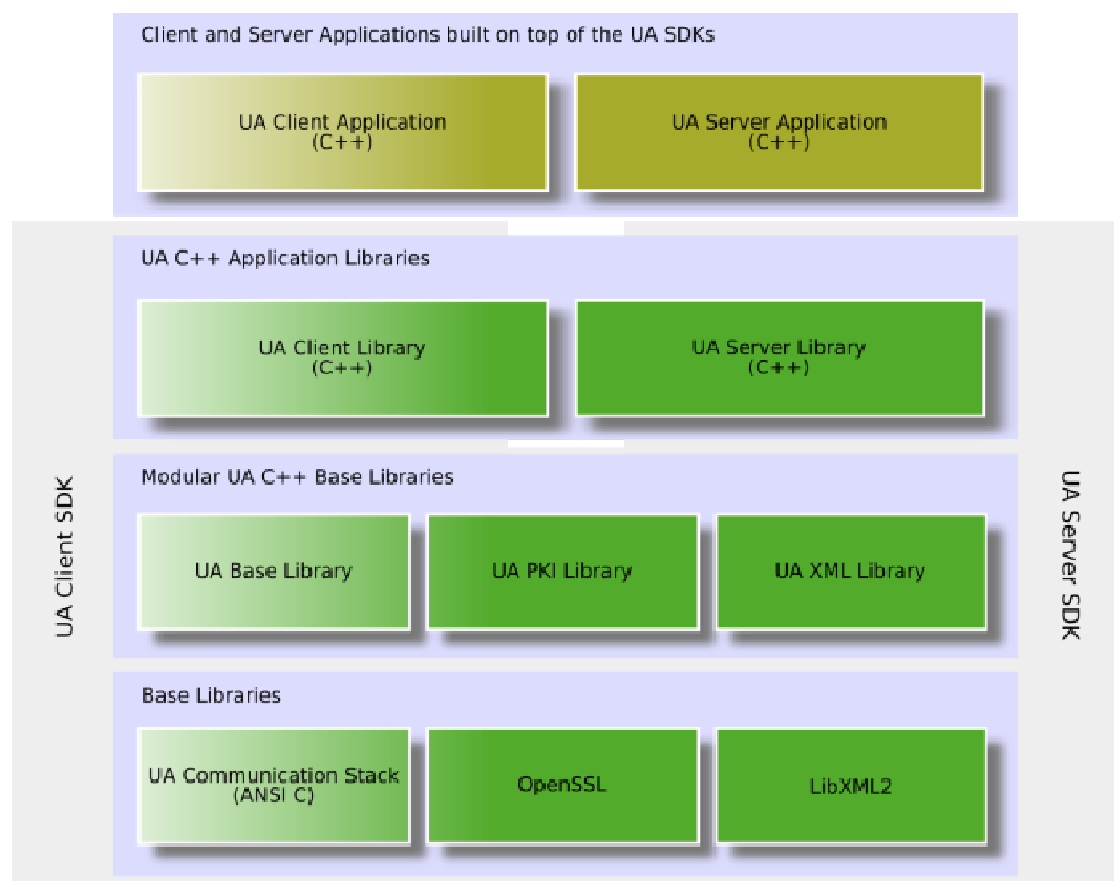
Supplier	Toolkit Types		Platform / Language			Notes
	Client	Server	C / C++	Java	.NET	
OPC Foundation	✓	✓	✓ *	✓ *	✓	* UA stack plus sample code only - not a full SDK. See note below.
Prosys	✓	✓	✓	✓		
Softing	✓	✓	✓		✓	Planned availability: 2009-Q4
Unified Automation	✓	✓	✓	✓		

2. Unified Automation C++ SDK

2.1 介绍

The UA SDK is a C++ library that supports you in writing portable C++ OPC UA Servers and Clients. The UA SDK actually consists of two SDKs, a Server SDK and a Client SDK. Both use the same UA Base Library which does all the C++ encapsulation of the raw ANSI C types that are defined in the OPC UA Communication Stack by the OPC Foundation.

2.2 架构概述



从该图中，我们能清晰的看到 C++ SDK 的 UA Base Library 与 Ansi C Stack SDK 的对应关系。

2.3 主要 Class 介绍

2.3.1 CoreModule

UA Server 的核心模块，用于管理地址空间，服务器配置，以及 Session 等。

2.3.2 UaModule

UaModule 与 UA Stack 接口，用于处理来自客户端的各种请求，并通过调用 Core Module 接口获取地址空间服务，最后生成对应的应答发送给客户端。

2.3.3 ServerConfig

ServerConfig 包含 UA 服务器启动时需要加载的信息，其中包括 Discovery Server，Endpoint，安全配置等信息等。在服务器启动时，CoreModule 将加载 ServerConfig 的配置信息，从而完成服务器内核的初始化。

2.3.4 NodeManagerUaNode

NodeManagerUaNode 与 CoreModule 接口，用于管理 CoreModule 中的地址空间数据。

其主要功能如下：

- 添加删除节点与引用
- 节点值读写
- 节点索引与浏览

2.3.5 IoManager, EventManager, HistoryManager

以上三个 Class 为 NodeManagerUaNode 的 Additional Service 选项，开发者通过对 NodeManagerUaNode 以及这三个 Class 的多重继承来实现地址空间对相应服务的加载。

- IoManager：用于支持客户端的读写请求
- EventManager：用于支持订阅与事件
- HistoryManager：用于支持对历史数据的访问

2.3.6 基本数据类型

基本数据类型包括 UaInt32, UaInt16, UaInt64, UaFloat, UaDouble, UaString, UaByte, UaByteString, UaBoolean, UaArray, UaVariant 等等。其中 UaVariant 是泛型数据类型，可以转换为任意类型。

2.3.7 地址空间相关数据类型

地址空间由 UaNode 与 UaReference 组成，节点又被分为不同的 Node Class，其中包括 UaObject, UaVariable, UaProperty, UaFolder 等，以上类型均为 UaNode 的子类。

2.4 License

Unified Automation C++ SDK 的授权方式概括如下：

- 用户可以使用该 SDK 编写自己的 End User 程序，在程序销售过程中，Unified Automation 将不收取附加费用
- 用户不能将该 SDK 封装成自己的开发包，卖给第三方。
- 每个开发者需购买一套 SDK

3.功能接口测试

3.1 服务器启动

- 初始化协议栈：UaPlatformLayer::init()
- 服务器配置加载：ServerConfig::loadConfiguration()
- CoreModule 启动：CoreModule::startUp()
- UaModule 启动：UaModule::startup()
- 获取并启动 ServerManager：CoreModule::getServerManager(),
ServerManager::startup()
- 启动 NodeManager: NodeManager::startup()

测试结果：UaExpert 客户端工具能够访问 UA Server 创建的 endpoint

3.2 构建地址空间

- 对象类型创建：new UaObjectType
- 对象类型加入到地址空间：addNodeAndReference(ObjectTypeBase,pObjectType,
Has_Subtype)
- 创建类成员：new DataItemTypeBase
- Instance Declaration：addNodeAndReference(ObjectType, pDataItem,
HasComponent)
- 对象类型创建：new UaObject
- 对象加入到地址空间：addNodeAndReference(ObjectFolder ,pObject, Organizes)

测试结果：UaExpert 能够访问程序创建的类型与对象。

3.3 Discovery service

- 在 ServerConfig::LoadConfiguration()函数中修改其成员函数 m_DiscoveryUrlArray 的值，将 Discovery Server Url 地址添加进去。

测试结果：UaExpert 能够通过访问 Local Discovery Server 来找到 UA Server

3.4 客户端通过局域网连接 UA Server

- 创建连接并注册 connectStatusChange 时触发的 Callback：

UaSession::connect(endPointUrl, endPointInfo, securityInfo, callbackPointer)

测试结果：成功连接到服务端后，当服务器关闭时，客户端会接收到断开通知，并调用对应的 callback。

3.5 客户端数据读写

- 读：UaSession::read(nodeIdsToRead, Value, Result)
- 写：UaSession::write(nodesToWrite, Result)

测试结果：客户端正确的打印读取的 node 值，同时通过 UaExpert 能够观察到客户端写入服务端 Node 的值

3.6 客户端订阅

- 订阅创建并注册 DataChange 时触发的 Callback :

```
UaSession::createSubscription(callback, subscriptionSetting,  
  
UaSubscriptionPointer)
```

- 创建监视项 : UaSubscription::createMonitoredItems(MonitoredItems,Result)
- 获取变更数据 : Callback::dataChange(clientHandler, data, diagnosticInfo)

测试结果：订阅成功后，对服务器的 Node 进行写操作，Node 值的变化会被反馈到客户端，并调用注册的 callback，打印变化的值。

3.7 客户端方法调用

- 方法调用 : UaSession::call(method, inParameter, outParameter)

测试结果：调用成功后，通过 UaExpert 观察到服务端的 Node 值发生变化

3.8 客户端创建节点与引用

- 添加节点 : UaSession::addNodes(nodesToAdd, Result)
- 添加引用 : UaSession::addReference(referenceToAdd, Result)

测试结果：添加成功后，通过 UaExpert 观察到新的 Node 的出现

3.9 与 OPC Foundation 标准协议栈的交互

- OPC Foundation Server Test Tool 与 Unified Automation C++ Server 交互。
- UaExpert 与 OPC Foundation Sample Server 交互。

测试结果：OPC Foundation Server Test Tool 成功的浏览 Unified Automation Server 中的地址空间，而 UaExpert 作为 Unified Automation 的客户端，也能浏览 OPC Foundation UA Server 的地址空间。

4.性能测试

4.1 主机性能参数

CPU: 2.8GHz

内存: 3.4GB

局域网带宽: 接近 100M

4.2 Server 端索引

地址空间中的节点由 NodeId 进行标示 , C++ SDK 提供的 NodeManager::find(NodeId)

接口可以用于索引地址空间中的 Node , 该函数在调用时会返回一个 UaNode 类型的指针。

NodeId 由 identity 与 NameSpaceIndex 两部分组成 ,其中 Identity 可以是以下几种数据

类型 : UInt32, UaString, ByteString 与 Guid。

经过测试 , 在 HashTable Size 为 200 万的情况下 , 采用不同 Identity 类型索引 100 万个

节点的时间如下 :

UInt32: 280 ms

ByteString (长度 6 字节) : 560 ms

UaString: 1200 ms

Guid: 并未测试 , 但根据其结构 , 可以断定出索引时间将大于 ByteString

4.2 Server 端读写

在地址空间中索引出 UaNode 后，将对 UaNode 中的数据进行读写操作，其中读取 100 万个点的值需要 300 ms，而写入数据需要 480 ms 的时间。写入时间比读取时间略长，这是因为写入前需要对数据的类型进行检查操作，

4.3 客户端读（同步）

Case 1：一个 Client 通过局域网对 Ua Server 中的 10 万个 Node 分为 10 万次读取。

结果：所用时间为 50-60 秒，即客户端从发送一个 Request 后，等待 Response 的间隔时间为 0.5 -0.6 ms 的时间。

Case 2：两个 Client 同时对 Ua Server 中的 10 万个 Node 分为 10 万次读取。

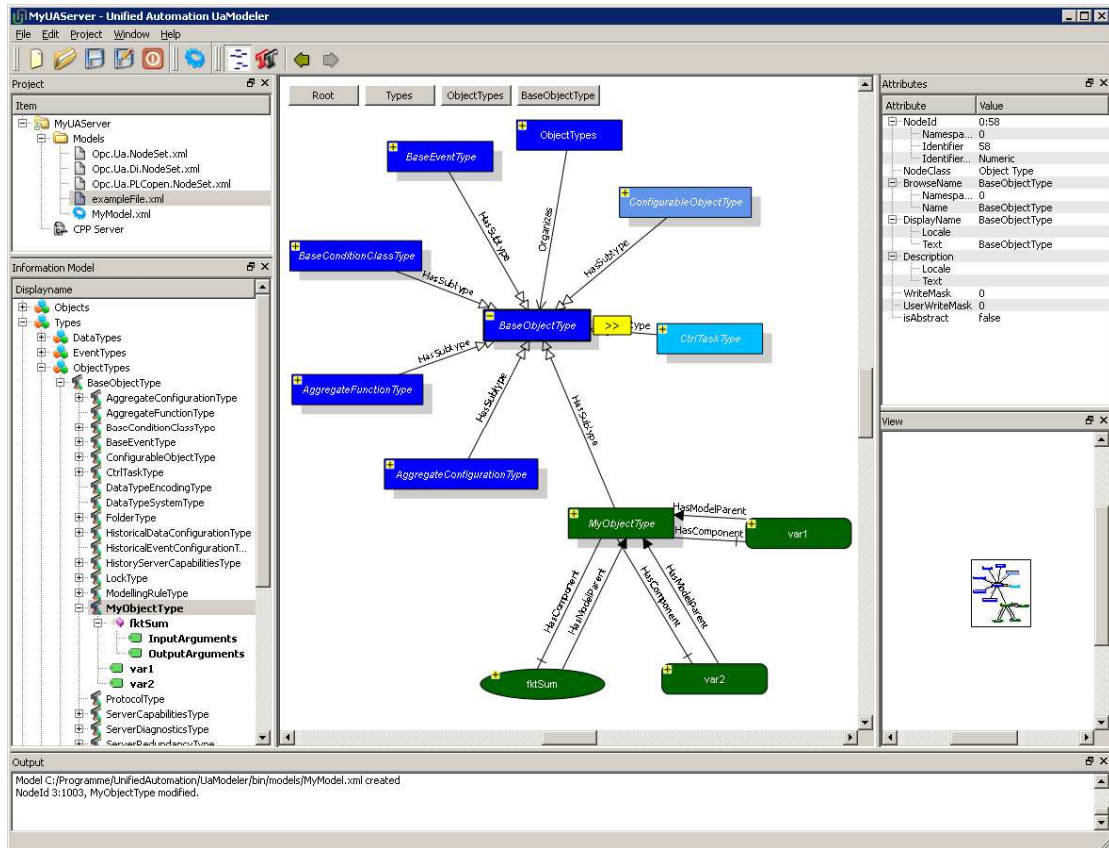
结果：两台机子分别用时 60 – 70 秒，即 0.6 – 0.7 ms/time。该数据与 Case 1 并无很大区别，这是由于两个 Client 并未对网络与服务器造成过高负载；当大量 Client 对 Server 进行访问时，系统负载变高时，等待应答的时间也将陡增。

Case 3：一个 Client 对 Ua Server 中的 10 万个 Node 分为 1 万次读取，每次请求 10 个 request。

结果：7-8 秒钟，即 0.7 ms/time

5. Unified Automation 工具

5.1 UA Modeler



The new release of the SDKs will include a specialized tool, the UaModeler, which not only simplifies the modeling of information itself, this tool generates the source code required to implement the designed model. The generated code exactly fits into the related SDK.

The UaModeler provides graphical design of address space, adding nodes and references and finally generates code by pressing a button, thus it not only speeds up the implementation, it also increases the software quality by producing well structured, error-free code. As implementation is reduced to “modeling and generating” even complex models can be “implemented” very quickly. This tool gives all the SDKs from Unified Automation the lead ahead on the OPC UA toolkit

market.

The UaModeler shows hierarchical and graphical representation of the designed model. The graphical representation follows the OPC UA notation and syntax.

5.2 UA Expert

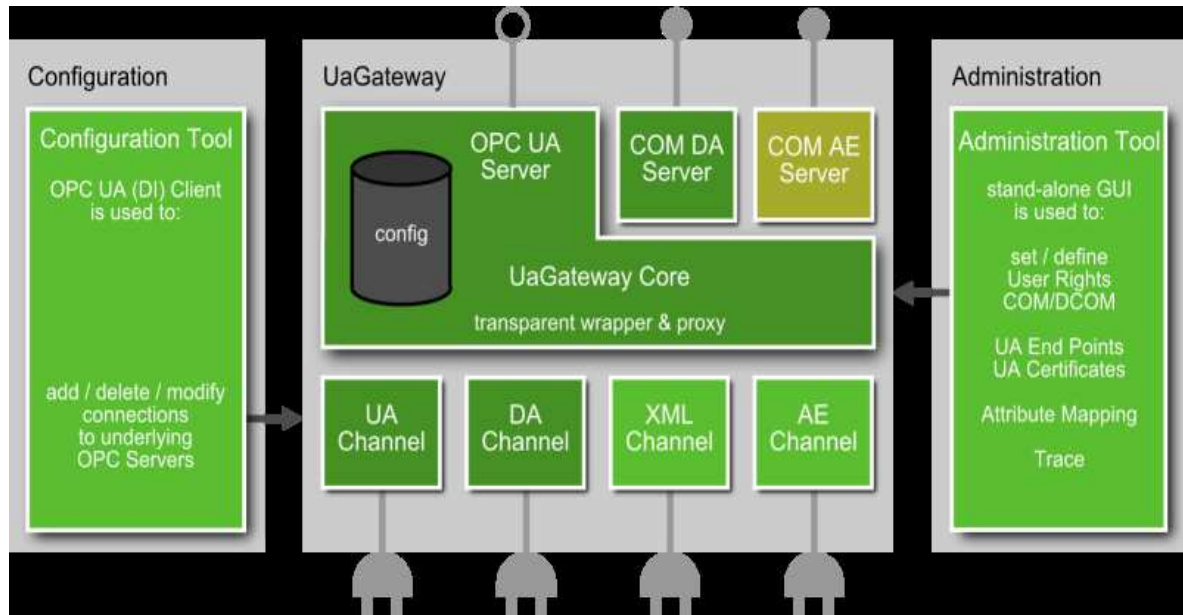
The UaExpert is a full featured OPC UA Client demonstrating the capabilities of our C++ OPC UA Client SDK/Toolkit. The UaExpert is designed as a general purpose test client supporting OPC UA features like DataAccess, Alarms & Conditions, Historical Access and calling of UA Methods. The UaExpert is a cross-platform OPC UA test client programmed in C++. It uses the sophisticated GUI library QT from Nokia (formaly Trolltech) forming the basic framework which is extendable by Plugins.

Free version of UaExpert comes with following Plugins:

- OPC UA Data Access View
- OPC UA Alarms&Conditions View
- OPC UA Historical Trend View
- DI Information Model Plugin
- OPC UA Performance Plugin

The UaExpert is available for Windows and Linux.

5.3 UA Gateway



The UaGateway is a product for migrating the 'classic' COM/DCOM based OPC Servers to the new OPC Unified Architecture. The UaGateway is a high performance Wrapper/Proxy programmed in C++. The UaGateway is based on our C++ OPC UA Server/Client SDK/Toolkit.

Solutions that can be obtained using the UaGateway:

- connecting UA Client to COM/ DCOM Server(s)
- connecting COM/DCOM Client to UA Server(s)
- tunneling COM/DCOM through secure UA connection
- data concentration on a single node; single point of access
- firewall-friendly, secure data access to 'classic' OPC

The new UaGateway is fully configurable through OPC UA Methods. It will comply with the OPC DI Specification (Device Integration) and will offer standardized diagnosis and configuration capabilities.

The UaGateway is a Windows 32Bit application; it can be installed on standard Windows (XP, Vista, Win7) and can also run on 64Bit Windows (WOW64). The UaGateway can connect multiple underlying COM DA Server and multiple underlying UA Server at the same time. Connections to underlying servers can be added during runtime without restarting the UaGateway.

5. 结论

鉴于 Unified Automation C++ SDK 良好的性能、完整的功能、合理的封装性与跨平台性，再加上其所提供的有效工具，建议选择该公司的 C++ SDK Professional。

注：UA C++ SDK Basic 仅支持 Standard UA Server 与 DataAccess Server Facet 两项，而 Professional 则支持 Basic Event Subscription Server Facet ,Method Server Facet , Node Management Server Facet , Auditing Server Facet , Client Redundancy Facet, Historical Access, Alarms&Conditions。