

# 【虎.无名】摘读空间

(学习、工作、生活) 网文阅读、摘要、评注

[主页](#)   [博客](#)   [相册](#) | [个人档案](#) | [好友](#)

查看文章

## WebService编程 (1:Axis; 2:Axis2/XFire; 3:CXF; 4:jdk6)

2009-08-15 10:22

【虎.无名：很早以前用过axis，只要把任意java类改名为jws部署到tomcat中，就可以发布为一个webservice，觉得很神奇。后来用MyEclipse通过XFire向导创建WebService也很方便。到了java6时代，在jdk中已经包含了wsgen和wsimport等命令，很容易发布和访问webservice，无需依赖任何框架或者库了。】

<http://www.javaeye.com/topic/128498> 如何正确地在Axis、Axis2和Apache CXF之间抉择？

新一代的 Web Services 框架如 Axis2、CXF 都是由现有的项目中逐渐演化而来的，Axis2 是由大家熟悉的 Axis 1.x 系列演化过来，而 Apache CXF 则是由 Celtix 和 XFire 项目整合而生，并且刚刚发布了 2.0.2 的最新版本，不过仍是 Apache 的一个孵化项目。

Axis2 是对 Axis 进行了彻底的重写的一个新项目了，它使用了新的模块化架构，更方便于功能性的扩展等等。

Apache CXF 则是由 XFire 和 Celtix 两个现有的项目进行了重组。

问题：如果现有的应用程序是基于 Axis 1.x、XFire 或者 Celtix 的话，那应该怎么办？都迁移到这些新的框架上去吗？但是即使是要迁移，那应该迁移到哪个框架上去呢？

如果是编写一个新的 Web Services 应用程序的话，就不存在迁移的问题了，但是哪个框架是你应当选择进行使用的呢？哪个比哪个更好呢？

对于现在的应用程序的迁移，如果你的应用程序是稳定而成熟的，并且在可预知的未来的情况下，只要很少的一些需求变更要做的话，那么保存你的体力，不要去做“劳民伤财”的迁移工作了。

如果你的现有应用程序BUG缠身，性能，功能等等都一片糟糕的话，那就要考虑迁移了，那选哪个框架呢？先比较一下它们的不同之处：

- 1、Apache CXF 支持 WS-Addressing、WS-Policy、WS-RM、WS-Security和WS-I BasicProfile
- 2、Axis2 支持 WS-Addressing、WS-RM、WS-Security和WS-I BasicProfile，WS-Policy将在新版本里得到支持
- 3、Apache CXF 是根据Spring哲学来进行编写的，即可以无缝地与Spring进行整合
- 4、Axis2 不是
- 5、Axis2 支持更多的 data bindings，包括 XMLBeans、JiBX、JaxMe 和 JaxBRI，以及它原生的 data binding (ADB)。
- 6、Apache CXF 目前仅支持 JAXB 和 Aegis，并且默认是 JAXB 2.0，与 XFire 默认是支持 Aegis 不同，XMLBeans、JiBX 和 Castor 将在 CXF 2.1 版本中得到支持，目前版本是 2.0.2
- 7、Axis2 支持多种语言，它有 C/C++ 版本。
- 8、Apache CXF 提供方便的Spring整合方法，可以通过注解、Spring标签式配置来暴露 Web Services和消费 Web Services

如何抉择：

- 1、如果应用程序需要多语言的支持，Axis2 应当是首选了；
- 2、如果应用程序是遵循 Spring 哲学路线的话，Apache CXF 是一种更好的选择，特别对嵌入式的 Web Services 来说；
- 3、如果应用程序没有新的特性需要的话，就仍是用原来项目所用的框架，比如 Axis1，XFire，Celtix 或 BEA 等等厂家自己的 Web Services 实现，就别劳民伤财了。

[http://www.diybl.com/course/3\\_program/java/javajs/20071022/79306.html](http://www.diybl.com/course/3_program/java/javajs/20071022/79306.html) XFire的下一代产品CXF的入门 (二)

<http://blog.csdn.net/pengchua/archive/2008/07/29/2731895.aspx> Axis1,Axis2,Xfire,CXF区别

## XFire VS Axis

**XFire**是与**Axis2** 并列的新一代**WebService**平台。之所以并称为新一代，因为它：

- 1、支持一系列 **Web Service**的新标准--**JSR181**、**WSDL2.0**、**JAXB2**、**WS-Security**等；
- 2、使用**Stax**解释**XML**,性能有了质的提高。**XFire**采用**Woodstox** 作**Stax**实现；
- 3、容易上手，可以方便快速地从**pojo**发布服务；
- 4、**Spring**的结合；
- 5、灵活的**Binding**机制，包括默认的**Aegis**,**xmlbeans**,**jaxb2**,**castor**。

**XFire**与**Axis1**性能的比较

- 1、**XFire**比**Axis1.3**快**2-6**倍
- 2、**XFire**的响应时间是**Axis1.3**的 **1/2**到 **1/5**

**XFire**与**Axis2**的比较

虽然**XFire**与**Axis2**都是新一代的**WebService**平台，但是**Axis2**的开发者太急于推出**1.0**版本，所以**1.0**还不是一个稳定的版本，它的开发者宣称**1.1**版本即将推出，希望**1.1**版本会是个稳定的版本。在**XFire**捐献给**apache**后有人认为**Axis2**将会灭亡。其实在很多人眼里，**Axis2**并不是**pojo**形式，**Dan Diephouse**证明了**XFire**比**Axis**更有市场，我也发现了有很多人开始从**Axis**转向**XFire**，包括我也在说服身边的人转向利用**XFire**进行**WebService**的开发，很典型的是我可以在几分钟之内教会我的团队实用**XFire**来发布一个他自己的**Web**服务。

本人倾向于**XFire**确实比**Axis2**简单很多

## AXIS VS CXF

在**SOA**领域，我们认为**Web Service**是**SOA**体系的构建单元（**building block**）。对于服务开发人员来说，**AXIS**和**CXF**一定都不会陌生。这两个产品都是**Apache**孵化器下面的**Web Service**开源开发工具。**Axis2**的最新版本是**1.3**。**CXF**现在已经到了**2.0**版本。

这两个框架 都是从已有的开源项目发展起来的。**Axis2**是从**Axis1.x**系列发展而来。**CXF**则是**XFire**和**Celtix**项目的结合产品。**Axis2**是从底层全部重新实现，使用了新的扩展性更好模块架构。**CXF**也重新深化了**XFire**和**Celtix**这两个开发工具。

新产品的退出导致了几个问题。是不是现有的使用**Axis 1.x**，**XFire**和**Celix**的应用需要迁移的新的版本上。如果一个开发人员确定要迁移它的应用到新的框架上，那么他应该选择哪一个呢？相反的，如果一个开发者决定从头开发一个新的**Web Service**，他应该使用哪个呢？这两个框架哪一个更好一些呢？对于系统迁移来说，也许迁移到新的框架并不难。**Axis**和**CXF**都提供了迁移的指导。能够给开发者一些迁移的技巧和经验。但是对于这样迁移，这两个开源项目都没有提供迁移的工具。对于这样的迁移工作，尽管很值得去寻找所有的可行方案。**Axis2**和**CXF**都有各自不同的**WebService**开发方法，每个方法都有相当数量拥护者。

通过一个比较矩阵来比较**Axis2**和**CXF**变得有现实的意义。这两个项目都开发不够成熟，但是最主要的区别在以下几个方面：

- 1.**CXF**支持 **WS-Addressing**，**WS-Policy**，**WS-RM**，**WS-Security**和**WS-I Basic Profile**。**Axis2**不支持**WS-Policy**，但是承诺在下面的版本支持。
2. **CXF**可以很好支持**Spring**。**Axis2**不能
3. **AXIS2**支持更广泛的数据并对，如**XMLBeans**，**JiBX**，**JaxMe**和**JaxBRI**和它自定义的数据绑定**ADB**。注意**JaxME**和**JaxBRI**都还是试验性的。**CXF**只支持**JAXB**和**Aegis**。在**CXF2.1**
4. **Axis2**支持多语言-除了**Java**,他还支持**C/C++**版本。

比较这两个框架的**Web Service**开发方法与比较它们的特性同样重要。从开发者的角度，两个框架的特性相当的不同。**Axis2**的开发方式类似一个小型的应用服务器，**Axis2**的开发包要以**WAR**的形式部署到**Servlet**容器中，比如**Tomcat**，通过这些容器可以对工作中的**Web Service**进行很好的监控和管理。**Axis2**的**Web administrion**模块可以让我们动态的配置**Axis2**。一个新的服务可以上载，激活，使之失效，修改**web**服务的参数。管理**UI**也可以管理一个或者多个处于运行状态的服务。这种界面化管理方式的一个弊端是所有在运行时修改的参数没有办法保存，因为在重新启动之后，你所做的修改就会全部失效。

**Axis2**允许自己作为独立的应用来发布**Web Service**，并提供了大量的功能和一个很好的模型，这个模型可以通过它本身的架构（**modular architecture**）不断添加新的功能。有些开发人员认为这种方式对于他们的需求太过于繁琐。这些开发人员会更喜欢**CXF**。

**CXF**更注重开发人员的工效 (**ergonomics**) 和嵌入能力 (**embeddability**)。大多数配置都可以API来完成, 替代了比较繁琐的XML配置文件, Spring的集成性经常的被提及, CXF支持Spring2.0和CXF's API和Spring的配置文件可以非常好的对应。CXF强调代码优先的设计方式 (code-first design), 使用了简单

的API使得从现有的应用开发服务变得方便。

不过你选择**Axis2**还是**CXF**, 你都可以从开源社区得到大量的帮助。这两个框架都有商业公司提供服务, **WSO2**提供**AXIS2**的支持, **Iona**提供**CXF**的支持。这两公司都有很活跃的开发社区。Axis2出现的时间较早, CXF的追赶速度快。我的建议是: 如果你需要多语言的支持, 你应该选择**AXIS2**。

如果你需要把你的实现侧重**JAVA**并希望和**Spring**集成, **CXF**就是更好的选择, 特别是把你的**Web Service**嵌入其他的程序中。如果你觉得这两个框架的新特性对于你并没有太大的用处, 你会觉得**Axis1**也是不错的选择, 你应该继续使用它知道你有充分的理由去更换它。

如何抉择:

- 1、如果应用程序需要多语言的支持, **Axis2** 应当是首选了;
- 2、如果应用程序是遵循 **Spring** 哲学路线的话, **Apache CXF** 是一种更好的选择, 特别对嵌入式的 **Web Services** 来说;
- 3、如果应用程序没有新的特性需要的话, 就仍是用原来项目所用的框架, 比如 **Axis1**, **XFire**, **Celtrix** 或 **BEA** 等等厂家自己的 **Web Services** 实现, 就别劳民伤财了。

[http://java.sun.com/developer/technicalArticles/J2SE/jax\\_ws\\_2/](http://java.sun.com/developer/technicalArticles/J2SE/jax_ws_2/) 用NetBean 5.5以上就可直接开发WebService

<http://www.cnitblog.com/sunnywang/archive/2009/04/14/56360.html> MyEclipse下开发web service (Java 5+ Eclipse 3.2 +MyEclipse 5.0.0. )

<http://www.blogjava.net/orangewhy/archive/2009/03/09/118485.html> Eclipse下使用Axis开发webservice

1.4. 使用java2wsdl生成wsdl

1.5. 使用wsdl2java生成stub

[http://www.ibm.com/developerworks/cn/edu/ws-dw-ws-eclipse-javase1.html?S\\_TACT=105AGX52&S\\_CMP=tec-cto](http://www.ibm.com/developerworks/cn/edu/ws-dw-ws-eclipse-javase1.html?S_TACT=105AGX52&S_CMP=tec-cto) 使用 Eclipse 和 Java SE 6 创建独立 **Web Services** 应用程序, 第 1 部分: **Web Services** 服务端应用程序

<http://chengda.javaeye.com/blog/378569> 用Eclipse在Java 6下开发Web Service

Java SE 6 对**Web Service**的开发提供了完全的支持。下面让我们利用Eclipse在Java 6下进行一个简单的**Web Service**的开发, 希望能给初学者带来好处。

1、 软件环境

1) 、 Eclipse

在<http://www.eclipse.org/downloads/>下载最新的Eclipse。

2) 、 JDK

在<http://java.sun.com/javase/downloads/>下载Java SE development kit 6

3) 、 Ant

使用Eclipse自带的Ant插件, 不用另外安装。

2、 开发服务器端程序

1) 、 创建一个Java项目, 命名为MyWebService\_server

2) 、 创建一个简单的类, 代码如下:

```
package test.ws;
import javax.ws.WebService;
@WebService
public class Hello {
    public String sayHello(String name) {
        return "Hello " + name + "!";
    }
}
```

这个类有一个方法 (sayHello), 参数是一个字符串, 代表一个人的名字, 返回值也是一个字符串, 意思是向这个人打招呼。这就是我们的**WebService**要提供的服务。

请注意类声明的前面有个annotation "**@WebService**", 这个注释的作用就是标明这个类是一个**WebService**。

2)、创建一个Ant脚本build.xml, 内容如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="wsngen">
<target name="wsngen">
<exec executable="D:\jdk1.6.0\bin\wsngen.exe">
<arg line="-cp ./bin -keep -s ./src -d ./bin test.ws.Hello" />
</exec>
</target>
</project>
```

请根据你自己的jdk安装路径对上面内容做适当修改

3)、运行ant脚本

4)、刷新项目, 查看生成的Java文件

5)、编写WebService启动程序

```
package test.ws;
import javax.xml.ws.Endpoint;
public class RunWebService {
public static void main(String[] args) {
Endpoint.publish("http://localhost:8000/hello", new Hello());
}
}
```

6)、启动WebService

7)、在IE中测试 (访问 <http://localhost:8000/hello?wsdl>)

3、开发客户端程序

1)、创建一个新的Java项目, 命名为MyWebservice\_client

2)、创建Ant脚本build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="wsimport">
<target name="wsimport">
<exec executable="D:\jdk1.6.0\bin\wsimport.exe">
<arg line="-s ./src -d ./bin http://localhost:8000/hello?wsdl" />
</exec>
</target>
</project>
```

同样, 请按您的jdk安装路径修改。

3)、运行ant脚本, 生成客户端程序, 刷新项目查看新生成的文件:

4)、编写测试程序, 测试客户端。

```
package test.ws.call;
import test.ws.Hello;
import test.ws.HelloService;
public class CallWebService {
public static void main(String[] args) {
Hello hello = new HelloService().getHelloPort();
String str = hello.sayHello("Mike");
System.out.println(str);
}
}
```

4、总结



一切都很顺利,“Hello Mike!”让我们着实兴奋了一阵子。我们用了很短的时间开发了一个简单的WebService及其客户端程序,显然这一切有赖于Java 6对WebService的完美支持。当然,这只是小试牛刀,读者可以自己尝试编写复杂的服务,比如:多参数、复杂类型参数、复杂类型返回值、抛出异常,等等。通过阅读不同情况下生成的wsdl你会理解WebService的内部机制。

WebService还有很多其他具体实现方法。比如apache的axis和axis2, weblogic也有自己的实现方法,等等。值得一提的是axis2,它也不需要开发者自己编写WSDL脚本,甚至不需要在服务器端生成任何额外的程序文件,只需把程序按规范部署就可以了,而且可以简单的部署到各种web服务器上。读者可以多尝试一下,根据实际情况取舍。

WebService做为SOA中的一项具体技术应用已经非常广泛,随着各种辅助工具的完善,我们的开发工作变得越来越容易和规范,看着清爽的代码,真的是一种享受。

【虎.无名】补充一些Axis2的资料,但就我个人不推荐使用。

<http://zhangjunhd.blog.51cto.com/113473/23690> 基于Tomcat5.0和Axis2开发Web Service应用实例

<http://www.ibm.com/developerworks/cn/webservices/ws-webaxis1/> 通过 Axis2 开发 Web 服务,第 1 部分:通过 Axis2 运行时部署和使用简单 Web 服务(2006 年 6 月 13 日)

<http://hideto.javaeye.com/blog/56835> 2007-03-01 **Axis2快速上手指南**

<http://blog.csdn.net/mudboy/archive/2006/09/08/1194535.aspx> 慎用AXIS2

AXIS2自带了Version.aar的示例,好象是返回一个STRING,可以运行,但用它的客户端工具自动生成的代码实在难看:

1、为什么非得封装一个\*\*\*Reponse, getMsg之类的参数和响应呢(使用ADB数据绑定)?在前面的版本中,生成的客户端使用起来就象是在本地一样(客户端代理完全实现了服务端接口/或说发布的方法签名完成一样),但现在却不是这样的,为了使用接口,你不得不生成一个封装类,再将你的参数设置进去,而返回时,又不得不从\*\*\*Response中显示的获取返回值(好像有个叫get\_return之类的方法),很是别扭。

2、生成客户端时提供好几种数据绑定方式,但我认为没有一种有以前的简洁,花哨不实用。

3、文档太简单了,真该花点时间写些像样的文档,最起码的,services.xml的DTD或Schema总得有一份呀!

接下来我们发现,使用自定义参数(无非是自己写的POJO)的接口发布后无法设用成功,返回的异常信息:reader is null (大概),开始还以为名称空间映射的问题,我们的自定义参数似下面的代码:

经过一番网上搜索和改配置(几乎试了所有的选项)/代码之类的操作,还是没有找到答案,错误还是发生。最好只好跟它的代码了,过程我就不多说了,只说结果吧。

1、错误发生在执行完要返回的路了,正准备将返回值序列化SOAP消息。

2、在ADB的代码中,有BeanUtils(不知道后面有没有S),里面有getFullParser方法,也就是,在序列化过程就,要通过该方法拿到一个Reader,但该方法返回了null。

3、该方法里面写得有些粗糙,很多可能的问题都没有想到,比如说,上面的POJO,在返回时有些为null,这应该是很正常的,但代码却假设任何一个属性都不为null(直接调用value.toString()方法),导致了NullpointException,发生该异常后,未记任何日志,而是直接返回了null,导致客户端跟据异常查错不太方便。

4、我们将我们的POJO改了改,对属性均做了初始化,使返回时每个属性都不为null,意外的时,我们还是没有成功。

5、这次异常还是因为getFullParser返回了null,难道不止一次调用了该方法?跟进去!发现代码是递归调用,即对于每个属性,也调用了该方法,代码里也是一堆if/else if之类的,对一些数据类型进行了处理,我发现在我们属性类型是Timestamp时返回了null,.....

6、它的序列化返序列化算法效率也有问题。

上面的过程很明显表现也AXIS2是那么的成熟不完善,代码BUG如此明显也是出人意料,本以为AXIS2应该有个飞跃(至少多实现了几个标准),但却没想到基石却如此不牢固,建议今后别在正式的项目中使用。

说到这,我又想起前段时间网上的一篇文章,大意也是说AXIS如何不好的话,说是一堆研究生写出来的,当时我并未在意,并一如既往的支持AXIS,直到昨天开始接触AXIS2,我的态度有了变化...

类别: [工作中学习](#) | [转帖](#)  | [添加到收藏](#) | [分享到i贴吧](#) | 浏览(409) | [评论](#) (4)

上一篇: [\(2009年8月3日\)时寒冰:依赖房地...](#) 下一篇: [开源Java项目: miniServer4J-0.1...](#)

相关文章:

- [JWS,Axis2,XFire,CXF](#)
- [Axis1,Axis2,Xfire,CXF区别](#)
- [Xfire客户端与CXF/Axis2服务器端...](#)

最近读者:



[登录](#)后, 您就出现在这里。



[wdl19861007](#)



[kingroc0928](#)



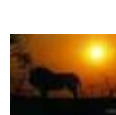
[feiyu1001](#)



[留·连](#)



[chengzq11](#)



[青春OF故事](#)



[www88485400](#)



[duckpia](#)

网友评论:

1



[milo1984cn](#)

2009-08-18 11:37 | [回复](#)

客户接口采用SOAP协议对外提供webservice服务, client如何用Axis2实现对客户接口的连接呢? 需要做一些什么样的配置?

2

网友:[nikeliu](#)

2009-08-25 09:59 | [回复](#)

使用axis2可以参考 (<http://hideto.javaeye.com/blog/56835> Axis2快速上手指南), 但从我个人来说不推荐使用。

4



[留·连](#)

2010-01-08 21:19 | [回复](#)

什么webservice Axis2 毕业论文里要用到, 可是什么都不会, 哎!  
MyEclipse6.5 + Axis2 + tomcat6.0 实现或扩充基于Apache Muse项目的原型系统 好复杂的东西

5



[nike\\_liu](#)

2010-01-12 21:23 | [回复](#)

The Apache Muse Project is a Java-based implementation of the WS-ResourceFramework (WSRF), WS-BaseNotification (WSN), and WS-DistributedManagement (WSDM) specifications.

所以, 你首先得读一下这三个基于WebService的规范。然后用Axis2发布一个小WebServer应用而已。没有太具体的需求, 所以还不算太难。

发表评论:

姓 名:  [注册](#) | [登录](#)

网址或邮箱:  (选填)

内 容:  插入表情

验证码:  请点击后输入四位验证码，字母不区分大小写