文章
Nicky Zhu · 四月 24, 2022 阅读大约需 10 分钟

在ObjectScript中调用Java程序 —— 一个国密算法的案例

尽管IRIS和HealthConnect拥有全面的互操作特性，但在实际工作中，还是有可能遇到需要使用遗留类库，dll SDK等方式与外部应用通信的情况。例如IRIS中并没有内嵌国密算法SM2、SM3和SM4，而开源社区中不乏通过Java、Python和C++等语言完成的具体实现。本文就将以调用SM4的Java实现为例展示ObjectScript程序与第三方语言通信的过程。

Github地址：https://github.com/LinZhuISC/javademo.SM4

跨编程环境调用设计要点

在开始实际操作之前，希望读者先针对整个调用过程中的主调方和被调方思考两个现象：

1. ObjectScript作为主调方，Java程序对它来说是个黑盒，它既不能直接访问Java虚拟机堆栈中的变量，也不能直接操纵被调代码的行为，例如限制内存使用、控制日志输出等。
2. Java程序作为被调方，其设计目的是通过Java容器运行或被其他Java程序调用，开发过程中通常并不会考虑其自身如何与另一个语言环境交互，因此不能确保异常信息能被主调方捕获与跟踪。

针对这两个现象，开发者需要思考，在哪一侧需要做什么样的一些工作以便调用过程能够顺畅进行。

在这个背景下，当我们需要让ObjectScript与Java相互通信和调用时，就不得不解决几个技术问题：

1. 两种体系结构间的数据类型如何相互转换：
   例如Java的int类型在ObjectScript中能顺利转为int吗？正如32位系统的Java int等同于64位系统的Java int吗？或者Java中的String与ObjectScript中的String等价吗？
   这是所有跨系统跨平台通信都会遇到的问题。比如32位的Java支持的int类型最大值比64位的Java要小得多，因此在大整型数的处理上，两者就会有不同。同理， ObjectScript中String的字符长度与Java中能支持的的String的最大长度也不同。因此，在构建跨技术平台调用时，需要有意识地进行类型转换和匹配。
2. 两种语言通信，传输的到底是什么？
   类型转换固然是一件繁琐而不怎么有趣的事，我们也可以从另一个方向思考。计算机通信的底层都是二进制数据流，并没有字符串和整型的区别。那么，在ObjectScript与Java进程通信的过程中，我们是否也可以直接进行二进制数据的传输，从而避免在一个运行时中去适配另一个运行时的数据类型？
3. 书写Java或.NET程序时不可避免地会引入大量的依赖（dependencies），如果不希望在ObjectScript这一侧来调试和验证classpath下包含所有依赖，那么在我们将Java程序打包时就需要处理这个问题——或把所有的依赖打在一个jar包里，或把所有依赖统一放在一个路径下供第三方调用。
4. Java程序有bug怎么办？
   事实上bug是不可避免的，那么当Java程序中的bug被触发的时候，开发者有没有办法从ObjectScript中来观察、定位和处理Java中的bug？参考上文主调方与被调方的特征，这显然不现实。因此，在Java程序中提供充足的异常捕获手段，在错误发生时记录日志以便在Java一侧还原、定位和解决问题，才能避免问题出现时被卡在两种语言之间不知从哪边着手的尴尬。

如上所述，大家在ObjectScript中应用Java程序时，通常不能期待对现存的Java程序不做任何改动即可顺利集成，而由于Java程序对于ObjectScript来说是一个黑盒，大量的准备工作都只能在Java一侧完成，使之能够适应跨运行时的调用。这些改动包括：

1. 重新打包并验证依赖的完整性。由于依赖不全导致的问题在实际工作中我们已多次遭遇，无论是dll依赖不全的.NET程序还是Java程序，发生问题时都极难调试和定位，因此更应在将发布包引入ObjectScript之前独立

验证。

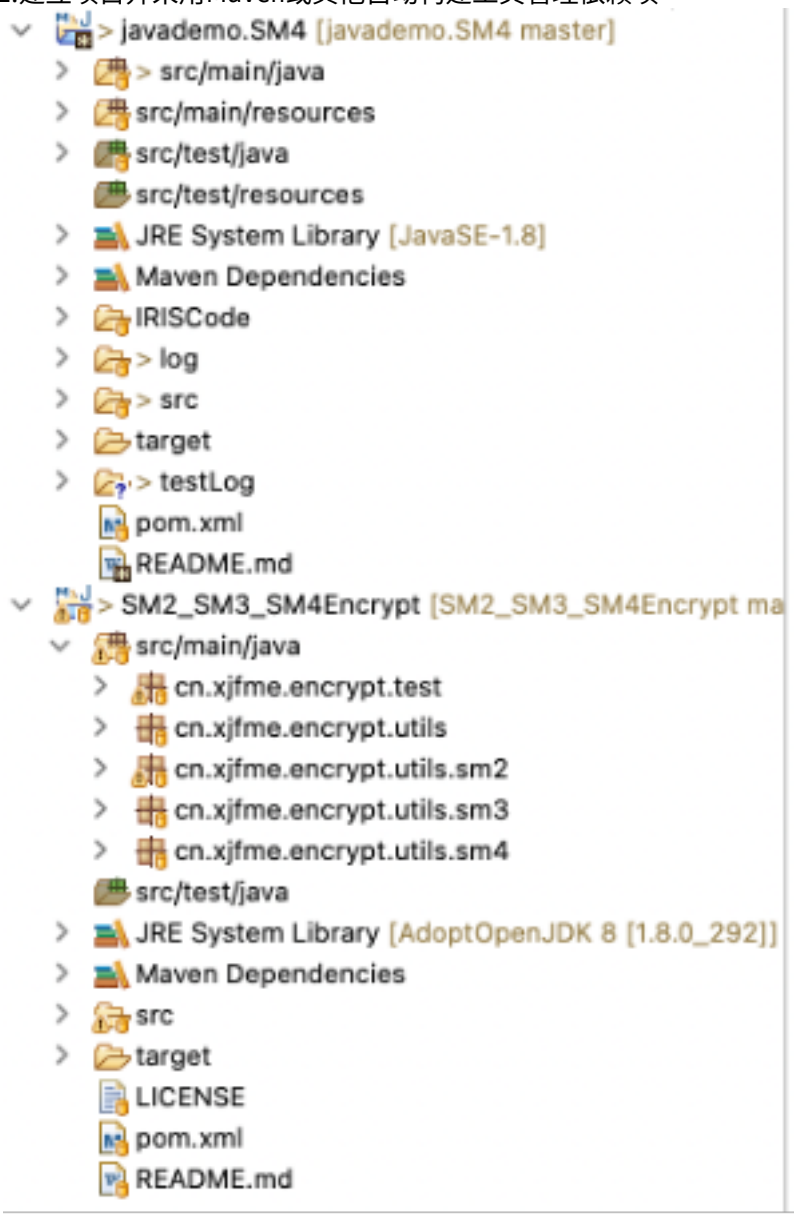
2. 对已有Java程序进行重构和包装，包括：

   1)．????????????????????????????
   2)．??

接下来我们来看实例，Java和Object Script代码都已上传到Github，读者可通过https://github.com/LinZhuISC/javademo.SM4获得。

Java程序改造

案例使用Eclipse IDE编写。
1.建立项目并采用Maven或其他自动构建工具管理依赖项



其中SM2SM3SM4Encrypt为从Github引用的国密算法开源项目(见https://github.com/xjfuuu/SM2SM3SM4Encrypt)，该项目同样使用Maven进行管理，但

在笔者使用的阿里云Maven存储库（http://maven.aliyun.com/nexus/content/groups/public
）中并没有记录，因此从Git引入工作空间后，开发者可通过maven install将安装在本地Maven库中再纳入管理。

如图所示，包括被引用的开源项目所依赖的Java包均已被解析并管理。

2.数据传输建模
已知引用的开源项目中ECB算法加密时需要两个输入，即密钥串和需要加密的文本或二进制内容。因此，对Java程序进行一次封装，将两个参数包裹在同一个POJO中，即com.intersystems.demo.SM4.Entity.EncTarget。如后续采用CBC算法，则还应加入CBC依赖的初始化向量iv。关于加密密钥和iv的格式需求，请查询SM4算法文档。

3.参数对象的序列化与反序列化
参数虽然以对象的方式组装，但在从ObjectScript向Java传输时可以被序列化为字符流，在Java中将该字符流反序列化为对象即可使用。本例中采用了alibaba的fastjson 1.2.79版本进行反序列化处理。

其中，传入的字符流应为按照EncTarget的结构序列化所得的JSON字符串，形如：
{"content":"This is the test content","secretKey":"9a3d2e6460b0483a84410a1fd14a9682"}

4.为Java程序添加日志
在异常发生时将传入内容记入日志中

本例中使用log4j2进行日志输出，配置文件见/src/main/java/log4j2.xml。由于已知log4j2存在若干漏洞，在生产环境构建时应更换为更安全的日志组件或更新版本。

5.书写命令行测试执行程序
在将Java包部署到IRIS侧之后，使用该程序验证相关依赖引用完整，加密解密程序可以正常运行。程序可见com.intersystems.demo.runner.CmdLineRunner。

6.将Java程序打包发布并验证
在使用maven的情况下，使用maven 打包即可，本例使用maven 的assembly插件打包。

其中的javademo.SM4-0.0.1-SNAPSHOT-jar-with-dependencies.jar为将所有依赖解包之后再重新汇集的产物，读者也可采用自己项目中常用的打包工具与策略。
在本例中，将项目打包并部署到IRIS服务器SM4Test目录后，从该服务器上用命令行执行Java程序，应得到如下结果，注意其中的异常是特意设计的用例，用于检测异常能够被抛出并被记录在日志中。

```
c:\SM4Test>java -classpath C:\SM4Test\* com.intersystems.demo.runner.CmdLineRunner
Test Succeed, decrypted String is This is the test content
java.lang.Exception: key error!
        at cn.xjfme.encrypt.utils.sm4.SM4.sm4_setkey_enc(SM4.java:207)
        at cn.xjfme.encrypt.utils.sm4.SM4Utils.encryptData_ECB(SM4Utils.java:41)
        at com.intersystems.demo.SM4.SM4WrapperWithJson.SM4Encrypt(SM4WrapperWithJson
.java:36)
        at com.intersystems.demo.SM4.SM4WrapperWithJson.SM4EncryptJson_ECB(SM4Wrapper
WithJson.java:61)
        at com.intersystems.demo.runner.CmdLineRunner.logCase(CmdLineRunner.java:51)
        at com.intersystems.demo.runner.CmdLineRunner.main(CmdLineRunner.java:15)
17:28:24.550 [main] WARN  com.intersystems.demo.SM4.SM4WrapperWithJson - Empty encryp
ted string for Content:This is the test content
17:28:24.566 [main] WARN  com.intersystems.demo.SM4.SM4WrapperWithJson - Using keySec
ret:9de7b44a-6cd7-48b8-a21a-5c0ab160eaa1
```

同时也可验证日志是否正常生成

在IRIS中引用Java程序

IRIS提供了数个工具使用Java或.NET程序，包括：
1. 代理类模式：在IRIS的ObjectScript程序中生成代理类。ObjectScript程序直接使用代理类，由代理类完成与Java虚拟机的交互。
2. 无状态服务模式：在Java一侧通过实现com.intersys.gateway.Service接口提供一个被调用的对象，在ObjectScript中可以通过直接访问由该接口定义的方法实现与Java的交互。

本例将以代理类模式为例构建对Java程序的调用。
1.在IRIS所在的服务器上安装和配置Java运行环境，并验证该环境可运行

2.配置Java Gateway
IRIS通过Java Gateway与JVM通信并驱动Java程序运行，可通过 系统 > 配置 > 对象网关 > 新建对象网关 - (配置设置) 配置Java Gateway

服务器名称/IP地址：Java Gateway可执行程序所在服务器的地址，通常为本地服务器，其IP为本地地址，即127.0.0.1
端口：选填未被占用且开放的网络端口即可
日志文件：可选配置，供Java虚拟机输出使用，如有需要长期使用Java Gateway，应为该日志所在的磁盘分配足够空间。
Java主目录：Java可执行文件所在目录，与JAVAHOME设置相同，通常为JRE或JDK根目录，即Java的bin目录的上级。如果Java典型配置（含JAVAHOME和Path等）在IRIS实例启动前已配置完成，也可以不填。
JVM自变量：选填，JVM启动参数，如设置Xms，Xmx对JVM进行调优等
类路径：除Java运行时相关的包外，被运行的Java程序及其所依赖的包应在此处引用。尽管ObjectScript也支持在程序运行过程中动态加载jar包，但效果不如预先加载稳定。在本例中，如果开发者也将所有依赖打在一个包中，那么也只需要引入该包即可（javademo.SM4-0.0.1-SNAPSHOT-jar-with-dependencies.jar）；如果采用程序和依赖包各自独立的打包，那么就需要将每个包的引用列出，以分号分隔。

3.生成Java代理类
用户可借助Studio解析jar包并生成代理类

在Studio的工具中选择Java 网关向导，

可导入整个jar中的所有类定义，也可以通过单个class文件导入类。
开发者也可使用ObjectScript代码指定从jar文件中导入特定的类，这样更灵活，要导入的类数量也最少，推荐使用。

4.调用代理类
在生成代理类之后，调用它就和调用一个典型的ObjectScript类一样

按照Java侧的设计，需要传入一个由对象转换出来的JSON字符流。我们可以同样建立一个参数对象，并应用ObjectScript提供的JSON包，将其转换为二进制流并通过代理类传递给JVM，即可实现功能。参数对象的定义可见Demo.Entity.EncPayload。
测试：

还应验证在异常发生时将生成Java的日志。在IRIS中Java程序执行的根路径为<IRIS安装路径>/mgr，可见生成的日志也在该目录下：

需要注意的是，在这个例子里我们通过应用序列化和反序列化去掉了对类型的依赖，ObjectScript与Java间传递的数据是二进制流。如果不对参数进行封装，直接从开源项目的方法生成代理类，用户就需要自行处理Java和ObjectScri

pt之间的类型转换。ObjectScript可支持的Java类型参见：
https://docs.intersystems.com/irisforhealth20212/csp/docbook/DocBook.UI....

上述ObjectScript代码也在Git代码中，大家可以通过Studio导入IRISCode目录下的JavaDemo.xml获得。

总结

如上所述，在从IRIS中调用Java代码时，需要有相当的工作量花在对Java代码进行重构和打包上，如果没有经过这些过程，那么Java程序中的bug以及两种语言交互中的类型转换等问题将很难被跟踪和解决。希望大家在工作中留意。
当然，即使使用这样的过程来优化调用，例如内存泄漏、虚拟机崩溃、Java程序死循环之类的问题仍然很难在生产环境中从ObjectScript这一侧被识别、跟踪和定位。因此，从IRIS中直接调用Java程序并不是最值得推荐和应用的办法。如果有可能，开发者还是应当考虑将需要依赖的Java程序封装为微服务，将IRIS与Java程序的运行时彻底隔离并打破代码级别的直接耦合。

最后，本例使用的SM4开源实现有严重的性能缺陷，在加密75K的文本时，耗时约10秒，在加密2M左右的文本时，耗时在10分钟以上，开发者应寻找替代开源实现或对其进行优化再用于生产环境。

#Eclipse #GitHub #Java #ObjectScript #InterSystems IRIS for Health

---

源URL:
https://cn.community.intersystems.com/post/%E5%9C%A8objectscript%E4%B8%AD%E8%B0%83%E7%94%A8java%E7%A8%8B%E5%BA%8F-%E2%80%94%E2%80%94-%E4%B8%80%E4%B8%AA%E5%9B%BD%E5%AF%86%E7%AE%97%E6%B3%95%E7%9A%84%E6%A1%88%E4%BE%8B