

图书分类号：_____
密 级：_____

毕业设计（论文）

题目：VMD开发与传统 P8 开发的比较分析

学 生 姓 名	_____
班 级	_____
学 院 名 称	_____
专 业 名 称	_____
指 导 教 师	_____

学位论文原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用或参考的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标注。

本人完全意识到本声明的法律结果由本人承担。

论文作者签名：_____ 日期：_____年____月____日

学位论文版权协议书

本人完全了解关于收集、保存、使用学位论文的规定，即：本校学生在学习期间所完成的学位论文的知识产权归所拥有。有权保留并向国家有关部门或机构送交学位论文的纸本复印件和电子文档拷贝，允许论文被查阅和借阅。可以公布学位论文的全部或部分内容，可以将本学位论文的全部或部分内容提交至各类数据库进行发布和检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

论文作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

VMD开发与传统 P8 开发的比较分析

摘 要

VMD，即面向业务的集成的可视化软件设计及开发一体化工具。通过 VMD 工具，可以将传统的详细设计过程模式化，避免了由详细设计文档转换为代码实现的重复劳动，同时将数据字典导入、数据模型导入、服务接口定义、构件开发实现、代码生成、编译、调试及部署等功能集成到同一界面中，并与 CC 配置管理等工具有效集成，支持了程序的可视化设计，代码及配置文件自动生成及管理的一体化实现。

使用 VMD 进行 P8 开发是我行 IT 发展战略要求，是新一代对各开发中心的要求，同时也是开发方式顺应时代潮流的转变。随着 VMD2.0，VMD3.0 版本的相继发布，VMD 在建行的推广和使用范围不断扩大，越来越多的项目组开始使用 VMD，VMD 也在大家的支持和监督下不断的完善，越来越好的满足用户的需求。在技术支持、测试、开发以及缺陷修复等工作的过程中，我们更加清楚的了解 VMD 与 P8 工程的匹配程度，VMD 在哪些方面符合 P8 开发的需求，有哪些方面还有改进和优化的空间和必要。

综上所述，本文旨在介绍使用 VMD 进行 P8 开发的方式和流程，从而将其与传统 P8 开发进行比较，得出使用 VMD 进行 P8 开发的优势，以及现存的问题，分析 VMD 下一步优化的方向。

关键词： VMD 可视化 P8 开发

目 录

1	绪论.....	3
1.1	研究背景.....	3
1.2	研究内容.....	3
1.3	研究意义.....	3
1.4	论文结构.....	4
2	P8 平台工作流程及技术理论综述.....	4
2.1	开放环境通用技术框架(JAVA)	4
2.2	P8 产品服务平台	6
2.3	应用组件开发.....	8
3	VMD P8 开发介绍	10
3.1	准备工作.....	10
3.1.1	导入 P8 工程.....	10
3.1.2	导入 VMD 模板工程.....	10
3.1.3	导入数据字典.....	12
3.1.4	关联工程.....	13
3.2	建立数据结构.....	13
3.2.1	导入报文数据结构.....	13
3.2.2	导入数据库数据结构.....	17
3.2.3	新建普通数据结构.....	19
3.2.4	数据结构添加后的结果.....	21
3.3	生成数据结构代码和 IBATIS 配置文件	22
3.3.1	配置源码输出路径.....	22
3.3.2	配置 IBATIS 模版，生成配置文件.....	25
3.4	发布构件.....	28
3.4.1	准备构件.....	28
3.4.2	发布构件.....	29
3.5	建立构件集.....	30
3.5.1	新建构件集.....	30
3.5.2	编辑构件.....	31
3.5.3	保存校验生成代码.....	32
3.6	建立联机服务和外呼服务.....	32
3.6.1	配置首选项.....	32
3.6.2	建立联机服务.....	34
3.6.3	建立外呼服务.....	38
4	VMD 开发与传统 P8 开发比较	41
4.1	VMD 与传统开发工具对比分析.....	41
4.2	VMD 开发特点	42
4.3	VMD 的设计目标	43
4.4	VMD 工具的设计方法	44

4.4.1	VMD 设计方法	44
4.4.2	VMD 功能模块	44
4.4.3	VMD 设计工具的平台层次.....	45
4.4.4	VMD 工具进行设计的规程.....	45
4.5	VMD 当前面临的问题	45
5	总结与展望.....	47
5.1	总结.....	47
5.2	展望及优化建议.....	47
5.1.1	原则.....	47
5.1.2	优化思路.....	48
5.1.3	下阶段优化内容.....	48

1 绪论

1.1 研究背景

业务需求能力的系统实现最终是由程序代码来完成的，程序编码设计和开发在整个应用软件开发过程中是一个非常重要的环节程序设计及编码开发方法、方式直接关系到开发效率和软件产品质量，对系统的安全稳定运行起到至关重要的作用。

随着系统的功能聚增和系统的组件化分层设计，系统的整体规模越来越大、复杂度越来越高，相应地对实现组件化的程序模块的设计开发要求也就越高，软件应用设计人员不仅要详细了解程序模块的业务功能和处理规则，而且还要熟悉相应应用平台开发框架、规范和编码语言等专业技术技能，因此如果能让软件应用设计人员在代码开发过程中只专注于业务处理流程设计和处理规则描述就能完成编码工作将是软件应用设计人员的理想追求。

在这样的现实背景下，可视化开发工具 VMD 应运而生。VMD（可视化开发工具）是一个客户端的开发工具，全称是面向业务的集成的可视化软件设计及开发一体化工具。VMD 是基于 IBM 的 RSA 平台的插件，旨在以流程图拖拽的方式代替手写 java 代码开发。VMD 的主要开发技术为 Eclipse 的插件开发技术，主要使用对象是各开发中心项目组的开发人员。20** 年 10 月中旬至今，我一直借调在新一代 P8VMD 项目组，工作以 VMD 开发为主，也会参与 VMD 的测试、需求分析、技术支持、配置管理等工作。

在新一代三期的开发阶段，将有更多的项目组和 P8 开发人员开始使用 VMD，因此 VMD 项目组将面临着更大的支持压力和更多的优化需求。

1.2 研究内容

首先，本文将介绍 P8 平台工作流程及技术理论；

其次，本文将介绍使用 VMD 进行 P8 开发的方式和流程，从而分析比较使用 VMD 进行 P8 开发和传统 P8 开发的区别，得出使用 VMD 进行 P8 开发的优势特点和现存问题；

最后，论文将进行总结，并分析 VMD 后续的优化原则、优化思路以及优化内容。

1.3 研究意义

使用 VMD 进行 P8 开发是我行 IT 发展战略要求，是新一代对各开发中心的要求，同时也是开发方式顺应时代潮流的转变。随着 VMD2.0，VMD3.0 版本的相继发布，VMD 在建行的推广和使用范围不断扩大，越来越多的项目组开始使用 VMD，VMD 也在大家的支持和监督下不断的完善，越来越好的满足用户的

需求。在技术支持、测试、开发以及缺陷修复的过程中，我们更加清楚的了解 VMD 与 P8 工程的匹配程度，VMD 在哪些方面符合 P8 开发的需求，有哪些方面还有改进和优化的必要。在新一代三期的开发阶段，将有更多的项目组 and P8 开发人员开始使用 VMD，因此 VMD 项目组将面临着更大的支持压力和更多的优化需求，因此，本文的研究是非常有必要的。

1.4 论文结构

本论文共分为五章。

第一章，绪论（即本章）。介绍论文的研究背景、研究意义、研究内容以及论文的组织结构。

第二章，P8 平台工作流程及技术理论综述。介绍建行北开 P8 平台开发工作流程现状及需要的技术理论基础。

第三章，详细介绍了使用 VMD 进行 P8 开发的具体步骤和流程。

第四章，将 VMDP8 开发与传统 P8 开发进行比较分析，介绍 VMD 的开发特点、设计目标 and 设计方法。

第五章，总结与展望，介绍 VMD 的下一步优化原则、优化思路 and 优化内容。

2 P8 平台工作流程及技术理论综述

2.1 开放环境通用技术框架(JAVA)

框架提供了 java 环境下的基本技术服务(与特有应用没有关系)，为各个使用 Java 环境的平台提供了基础开发支持。

目前框架的主要功能范围如下：

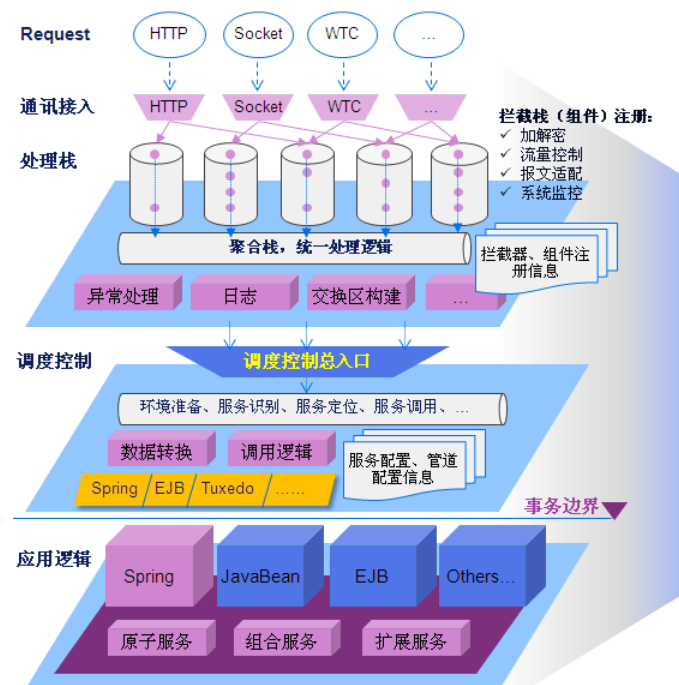
完成开放环境通用技术框架控制驱动程序组件（请求接入处理、调度控制、应用逻辑处理）的开发；

设计控制驱动程序组件调用功能处理程序组件的机制并实现；

完成典型的功能处理程序组件（通讯适配、报文适配、加解密处理、对外呼出、流量控制）的开发；

完成技术处理程序组件（缓存组件、日志输出、数据库访问、通用参数管理、

标准代码、文件访问、系统监控、异常处理、序号生成器、内部数据交换、界面展现) 的开发；



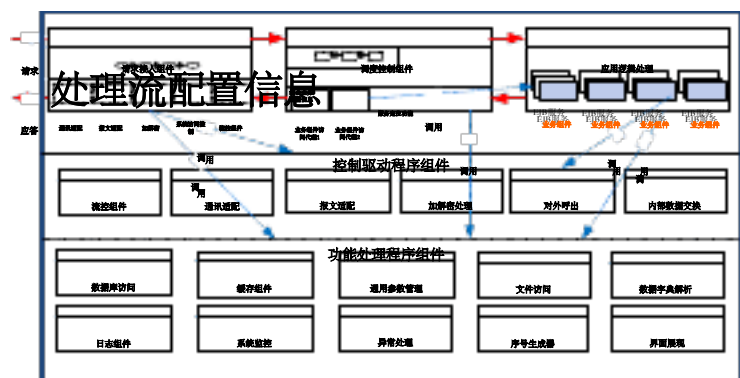
组件架构示意图

框架支持 HTTP、Socket 和 WTC 等多种方式的服务接入，通过新增适配器，可对接入方式进行扩展；系统可根据不同的接入方式和接入参数，为请求分配不同预处理栈（拦截栈），预处理栈由预处理组件构成，可完成如加解密、流量控制、报文适配和系统监控等功能。预处理组件都实现统一的接口，通过配置构成预处理栈，可以方便的扩展和插拔；

框架通过传递内部交换区发起对统一调度入口组件的调用，统一调度根据交换区内的请求数据、交易配置信息以及预处理管道配置信息，完成环境准备、服务识别、服务定位和服务调用等任务；

具体服务通过服务调用代理被调起，若需事务，也在服务调用时开启，可被识别和调用的服务包括 Spring、EJB、Java Bean 和组合服务等，应用系统可通过实现自己的服务代理达到调用特殊类型服务的目的。

组件层次示意图



框架组件库逻辑层次图

框架层由三种组件类型构成：控制驱动程序组件、功能处理程序组件、技术处理程序组件。

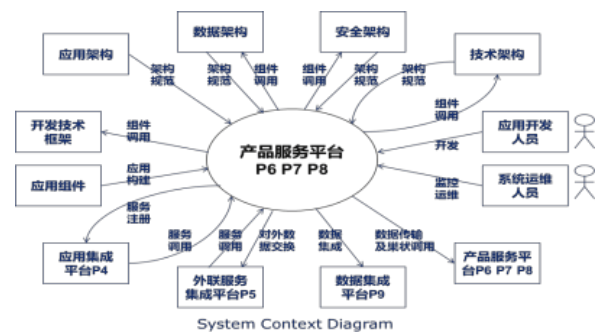
技术处理程序组件主要是可重用的技术组件，通过 API 方式由逻辑层次图中的其它组件使用；功能处理程序组件是围绕交易请求、应答数据的处理，如报文的

适配、加解密等，这部分的功能由框架根据配置信息由控制驱动程序组件调用。控制驱动程序组件组成框架的主控流程负责功能处理组件的调用,如果需要用到技术处理组件通过 API 的方式调用。

2.2P8 产品服务平台

总体架构

产品服务平台 P6、P7、P8 作为一个整体，与其他领域、平台、组织和人员的关系如下图所示：



其具体说明如下：

与应用架构的关系：应用架构对 P6-8 平台设计提供架构规范；

与数据架构的关系：数据架构对 P6-8 平台设计提供架构规范；P6-8 平台对数据架构提供的组件和构件进行调用；

与安全架构的关系：安全架构对 P6-8 平台设计提供架构规范；P6-8 平台对安全架构提供的组件进行调用；

与技术架构的关系：技术架构对 P6-8 平台设计提供架构规范；P6-8 平台对技术架构提供的组件进行调用、对工具软件进行集成；

与应用集成平台 P4 的关系：其他平台都通过 P4 平台调用 P6-8 平台的服务；P6-8 平台的服务都在 P4 平台中注册；

与外联服务集成平台 P5 的关系：银行外部系统通过 P5 平台调用 P6-8 平台的服务；P6-8 平台通过 P5 平台对银行外部系统进行数据交换；

与数据集成平台 P9 的关系：P6-8 平台采用准实时或定时批量模式将业务数据同步到 P9 平台；

P6、P7、P8 平台间的关系：P6-8 平台之间通过数据传输以实现某些跨组件的批量业务处理，而且必要时采用巢状调用以解决某些复杂的业务逻辑；

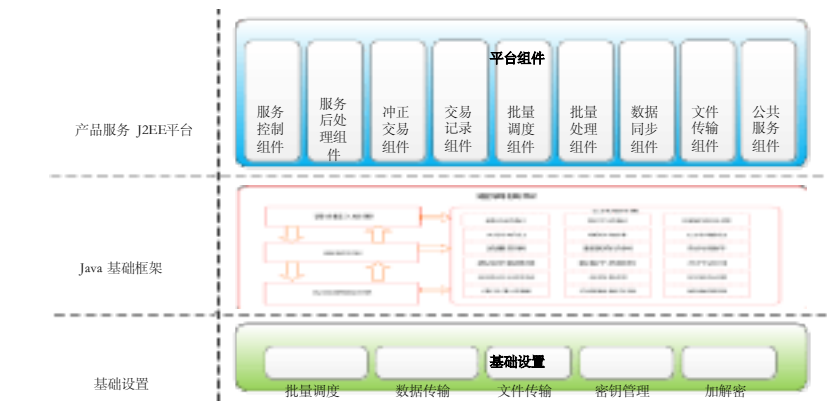
与开发技术框架的关系：P6-8 平台构建在对应的开发技术框架之上，并且对开发技术框架提供的组件进行调用；

与应用组件的关系：产品服务层的 73 个应用组件构建在 P6-8 平台之上，由平台进行驱动和调度；

与应用开发人员的关系：应用开发人员在 P6-8 平台之上开发应用组件；

与运行维护人员的关系：运行维护人员通过 P6-8 平台提供的功能对应用组件的运行状况进行监控。

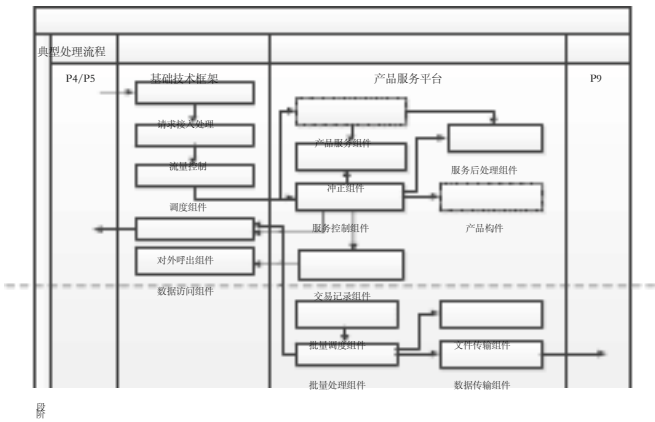
应用架构



P8 产品服务平台也叫做产品服务 J2EE 平台。产品服务 J2EE 平台依赖于 J2EE 基础技术框架和技术架构、数据架构、安全架构等基础设施组件。产品服务 J2EE 平台基于 J2EE 基础技术框架设计开发,是对 J2EE 基础技术框架的扩充,基本上是对框架功能处理组件的扩展,和框架功能处理组件处于对等的位置。

基础设施是包括数据架构、技术架构和安全架构提供的公共组件，依赖这些组件完成一些公共功能。比如：文件传输组件、数据传输组件、批量调度组件（Control-M）、密钥管理、加解密、短信服务、邮件服务等。

交易数据流模型



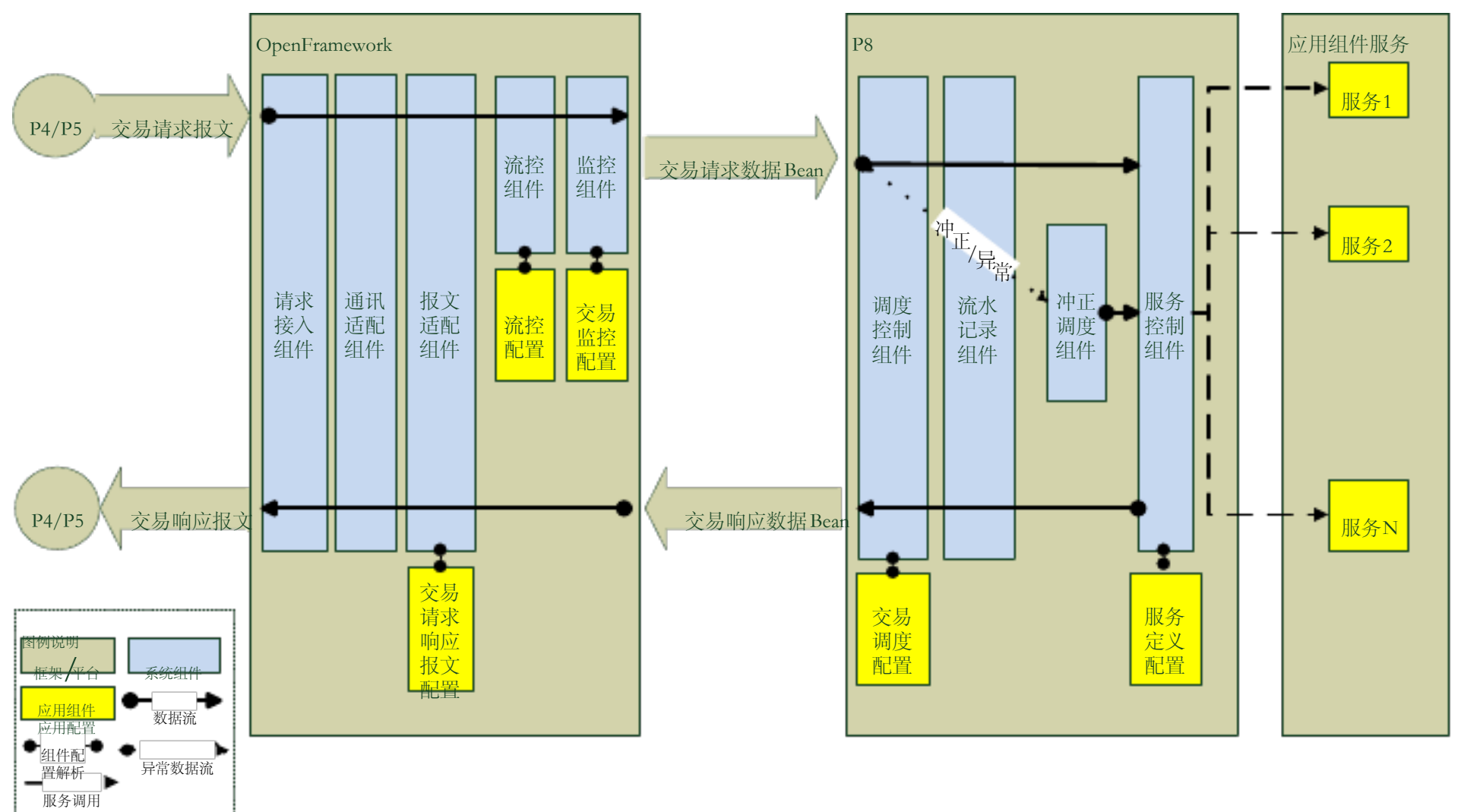
对于联机交易处理，产品服务平台通过基础技术框架提供的请求接入处理组件，接收来自 P4/P5 的交易服务请求。通过基础技术框架提供的流量控制组件、通讯适配、报文适配后，由基础技术框架的调度控制组件将请求交由平台的服务控制组件处理，调度控制组件负责根据服务配置，调用具体的产品服务。如果请求的产品服务，是由平台提供的构件组合配置形成的服务，则由服务控制组件负责根据配置调度具体的产品构件。

在服务处理过程中，如果出现异常，则由平台的冲正组件负责交易的冲正补偿调度，以保证交易的完整性和一致性。服务处理完成后，由服务后处理组件根据服务配置进行相应的处理。同时，采用平台提供的交易记录组件，通过基础技术框架提供的数据库访问组件，进行交易流水等痕迹记录的数据库操作。如果在服务处理过程中，需要访问其他平台的服务，则通过基础框架的对外呼出组件调用 P4/P5 上发布的服务。

对于批量交易处理，则由批量调度组件，根据批量任务配置信息，定时进行批量任务的调度处理，具体的任务处理，由批量处理组件完成，如果涉及批文件传输或数据同步等任务，则调用文件传输组件或数据传输组件完成。

2.3 应用组件开发

下图黄色区域标注就是需要应用组件自己定义开发的。



- 1、流控配置、交易监控配置为统一管理配置，为应用组件共享。
- 2、交易响应报文配置为交易输入报文和响应报文配置。
- 3、交易调度配置定义了每个交易的处理流程。
- 4、服务定义配置定义了交易所需要用到的各种服务。
- 5、服务1...N为各种Java 代码或者是外部引用服务。

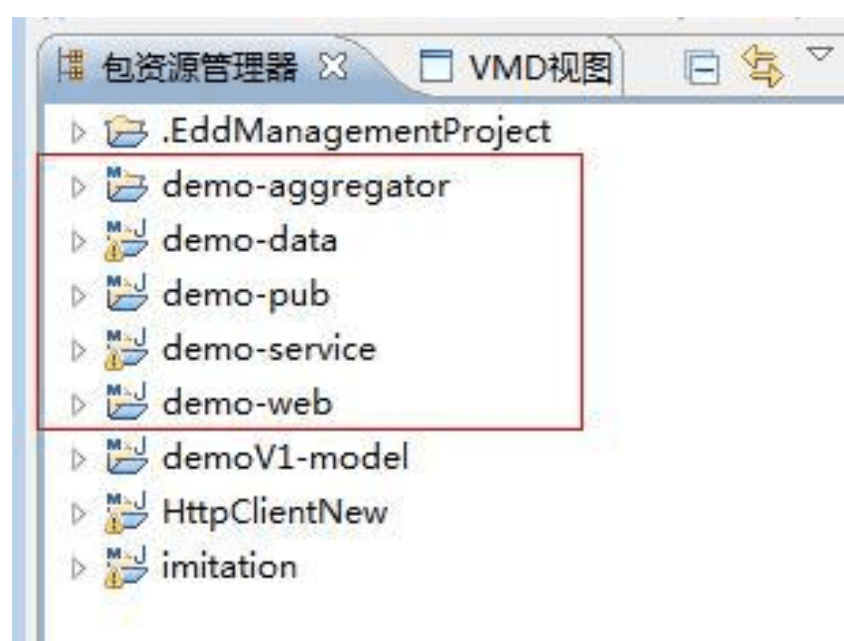
3 VMD P8 开发介绍

下面以现金项目组的“维护机构额度”交易为例,介绍使用VMD进行P8开发的具体步骤流程。

3.1 准备工作

3.1.1 导入 P8 工程

第一步,启动VMD,将P8工程导入到工作空间中。导入后,在包资源管理器下可以看到P8工程,如下图所示。

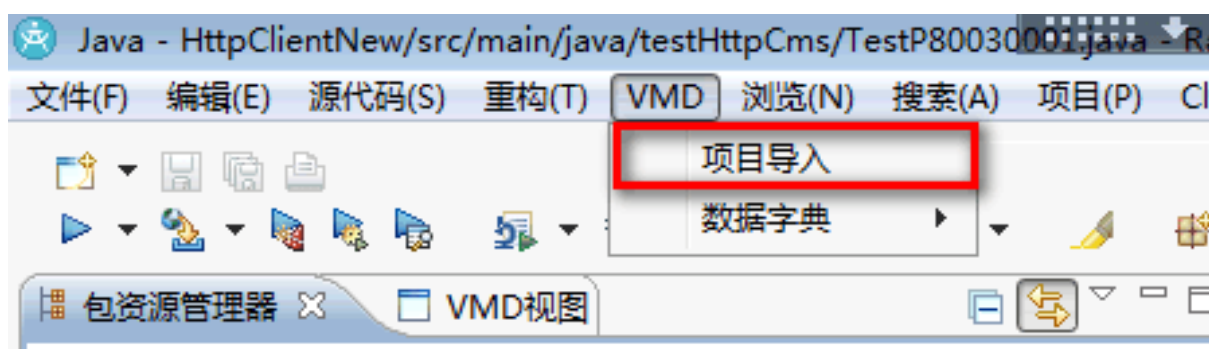


3.1.2 导入 VMD 模板工程

在云桌面的浏览器中输入地址<http://128.194.2.134:8080/CCBVMD/versions/>,进入“VMD自测版本”,之后进入“优化版本工程模板”,下载最新模板工程,如图所示。



在菜单栏点击VMD,选择项目导入。



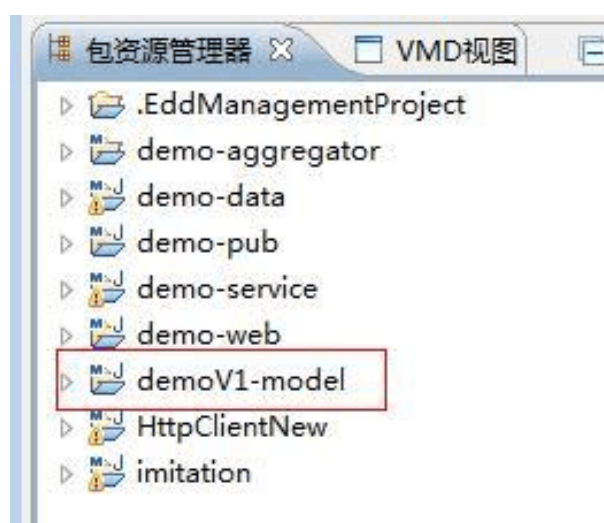
在弹出框中，选择刚才下载的模版文件，并为项目命名，指定中文名，同时勾选“是否创建EDD项目”。



编辑完成之后，点击确定。



导入完成之后，包资源管理器和VMD视图分别如下所示。

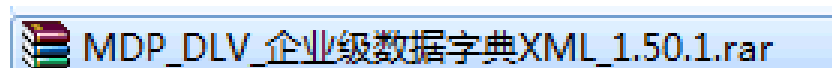




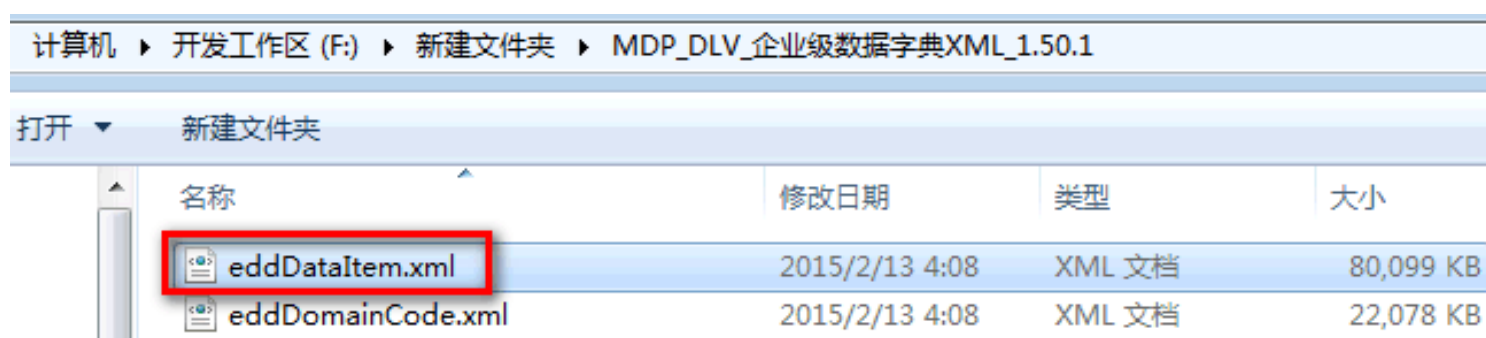
3.1.3 导入数据字典

从云上flpm下载最新的全量数据字典xml格式文件并导入。

下载的数据字典文件如下图所示。



解压后选择eddDataItem.xml。



在菜单栏选择VMD下的数据字典，点击导入。

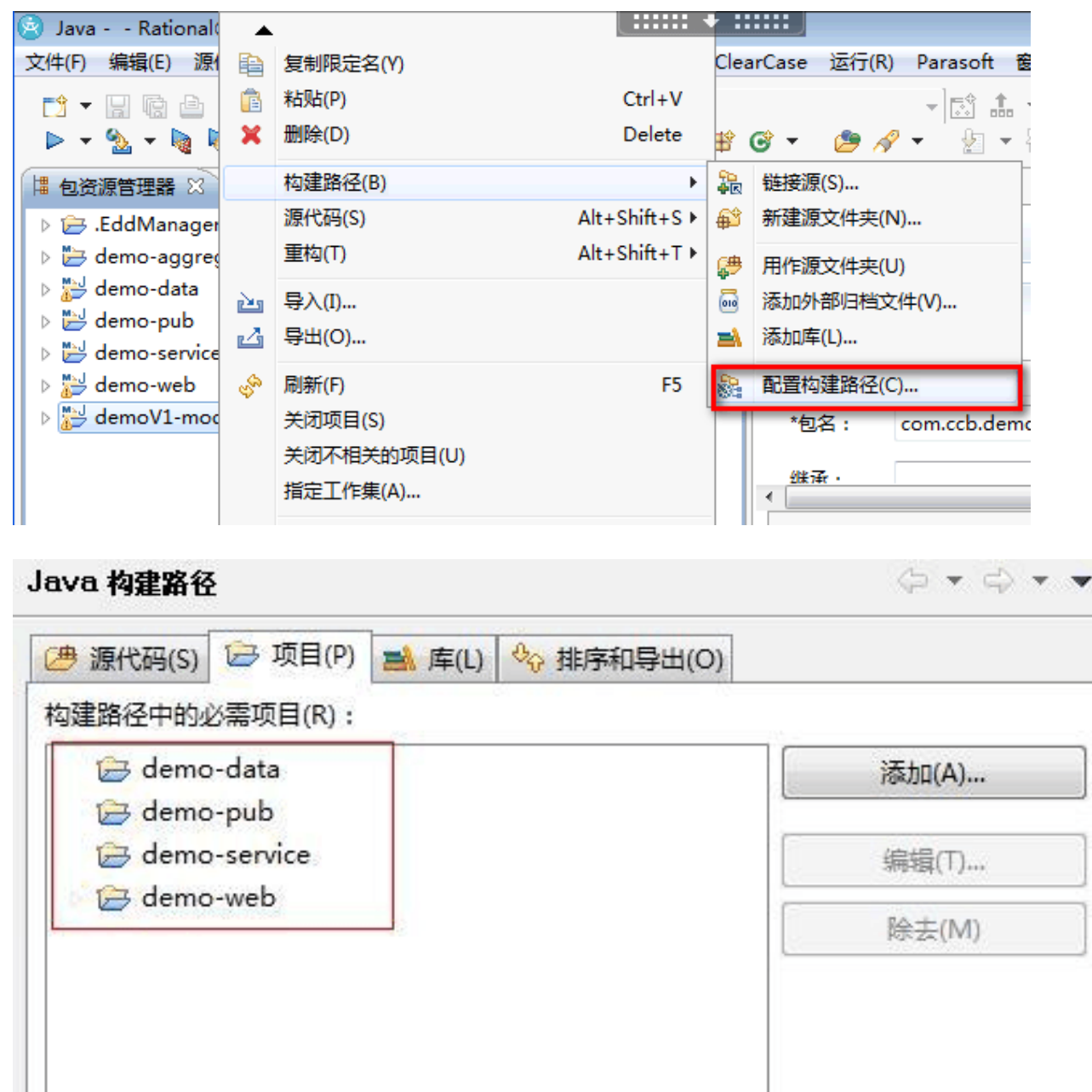


选择之前下载的文件后，点击确定，等待导入完成！



3.1.4 关联工程

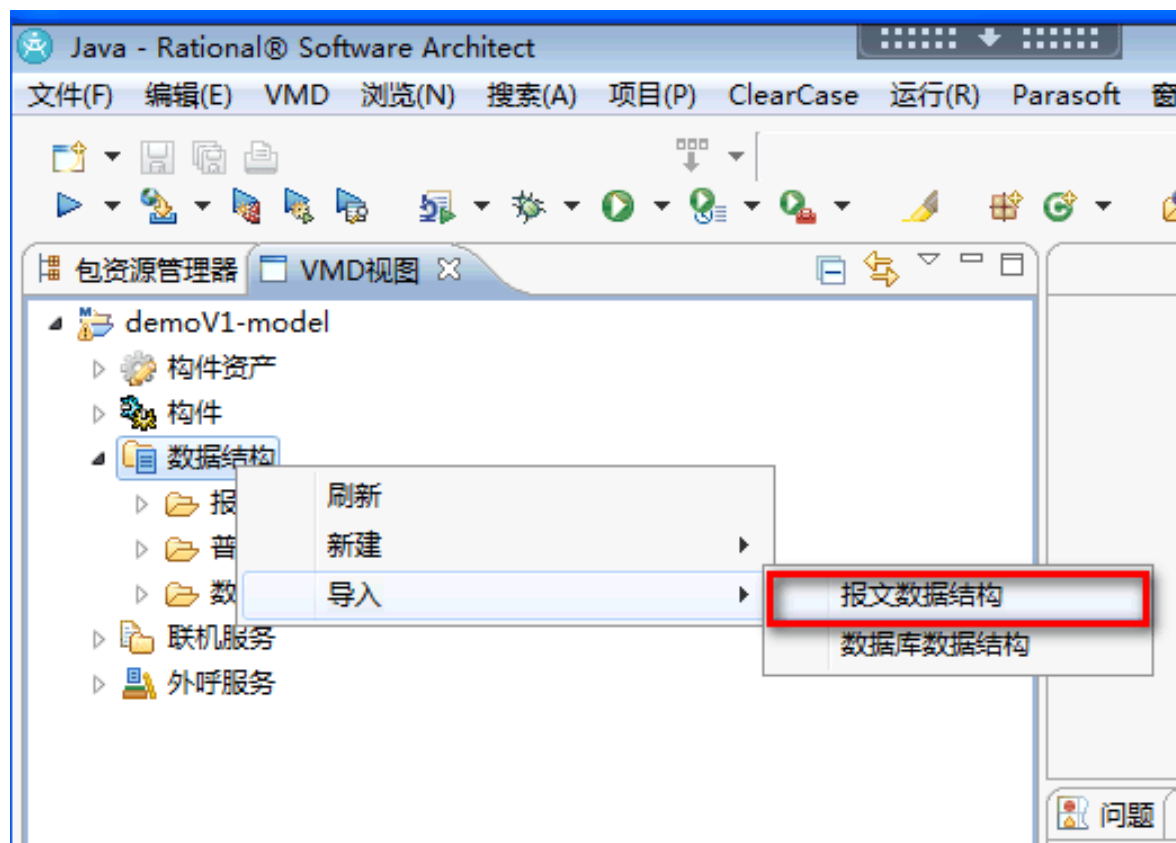
右键项目，选择构建路径，配置关联P8工程，如下图所示。



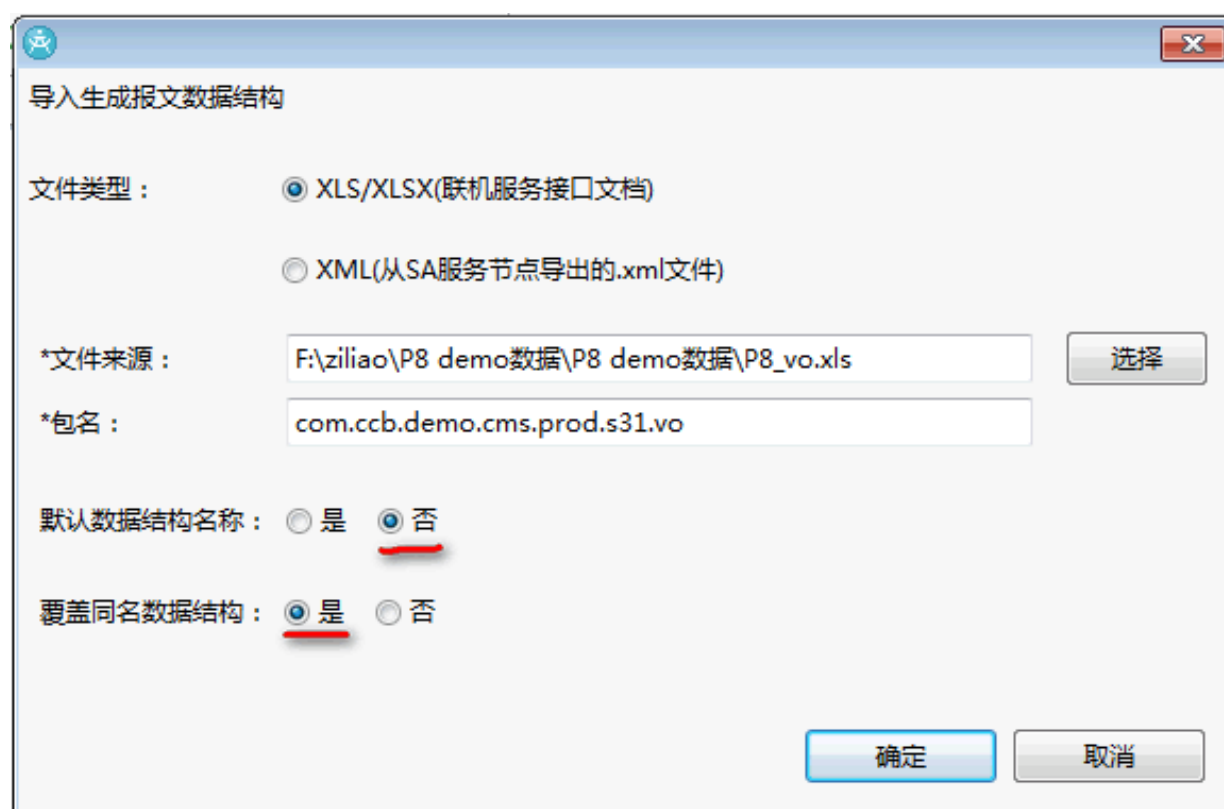
3.2 建立数据结构

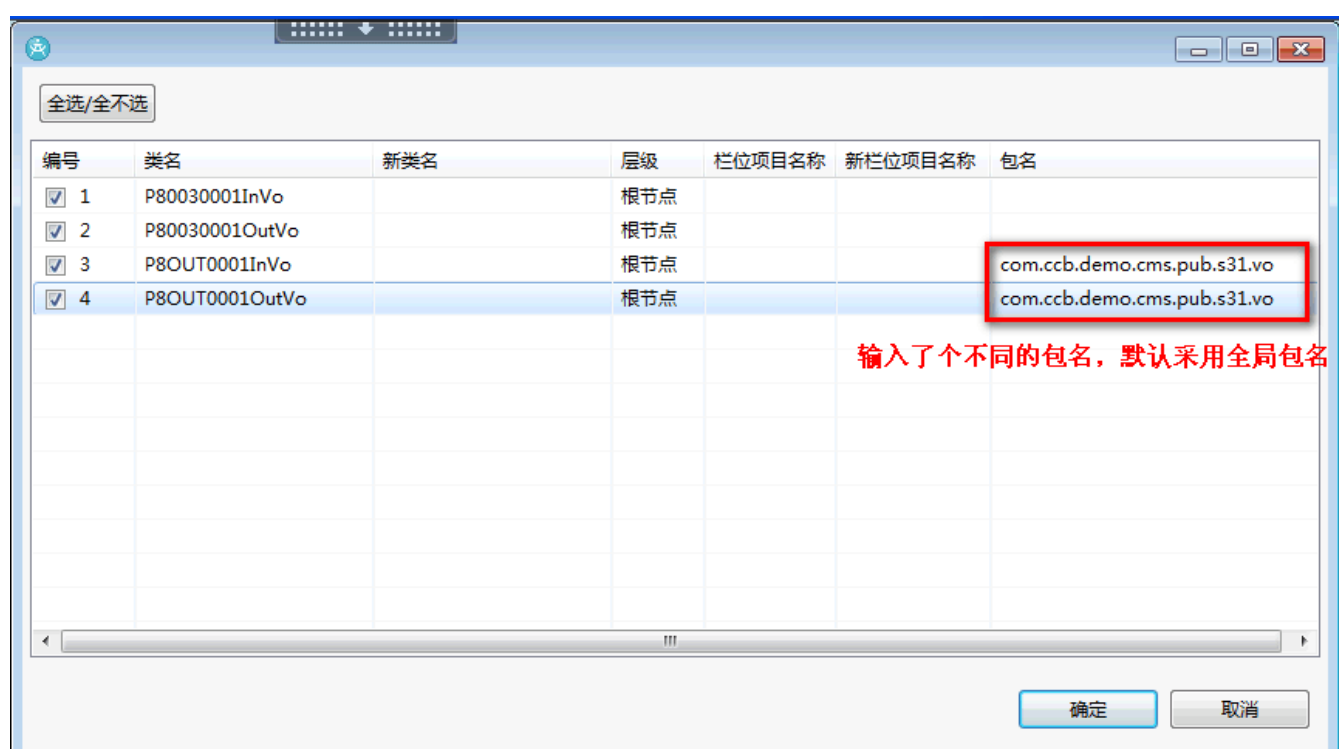
3.2.1 导入报文数据结构

在vmd视图的数据结构节点，右键选择导入，首先导入报文格式的数据结构，如下图所示。



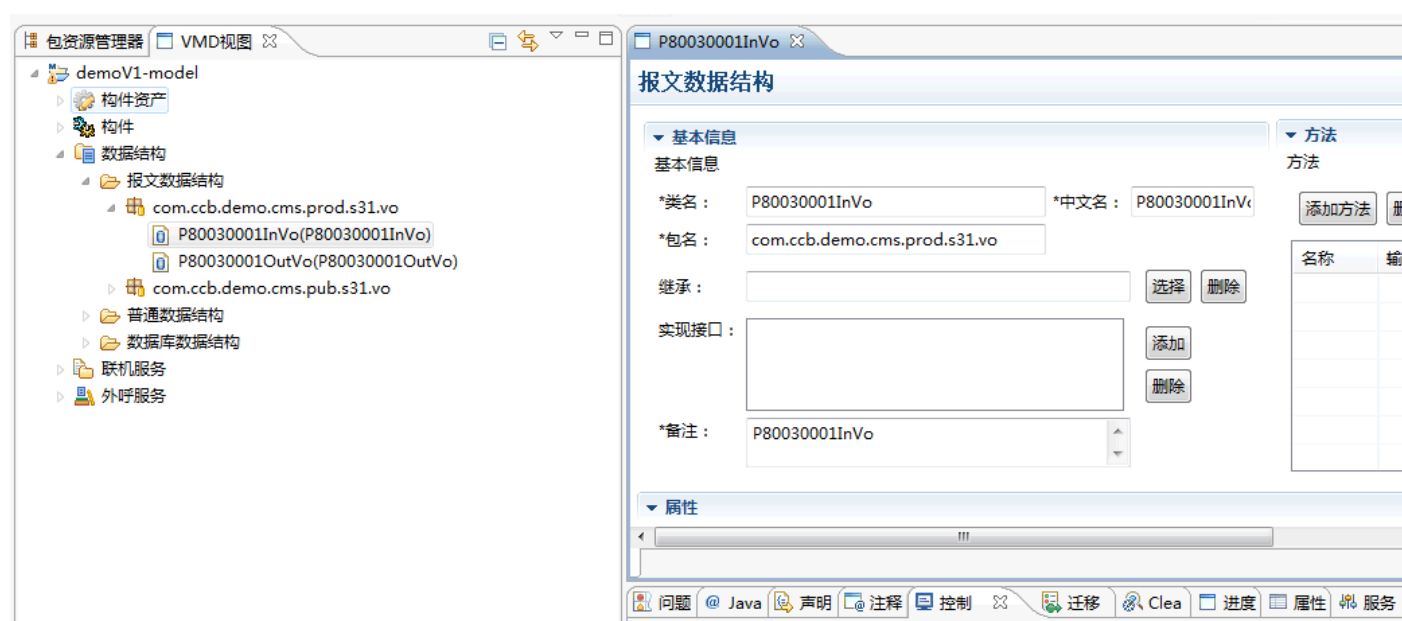
开始选择文件（P8 demo数据中的xls 文件），并输入包名（VO类的全路径）。



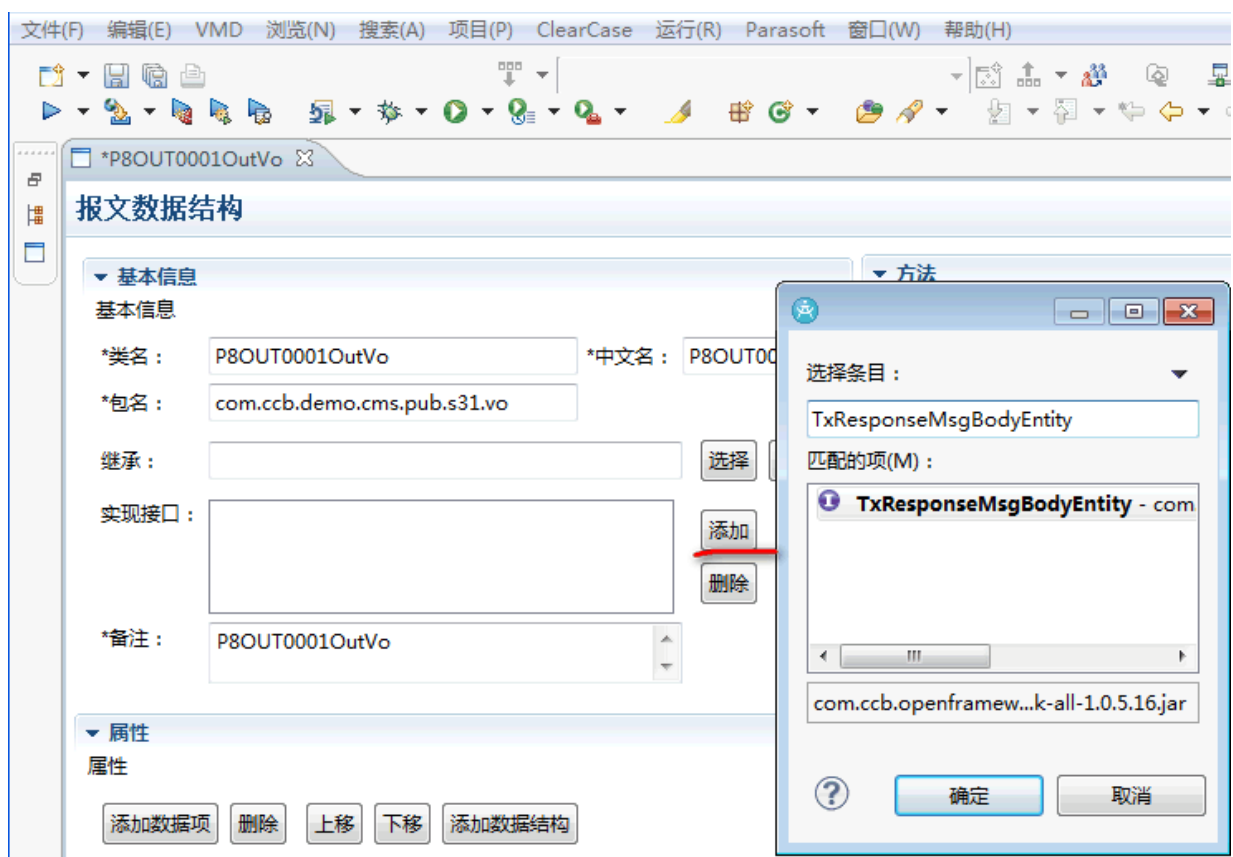
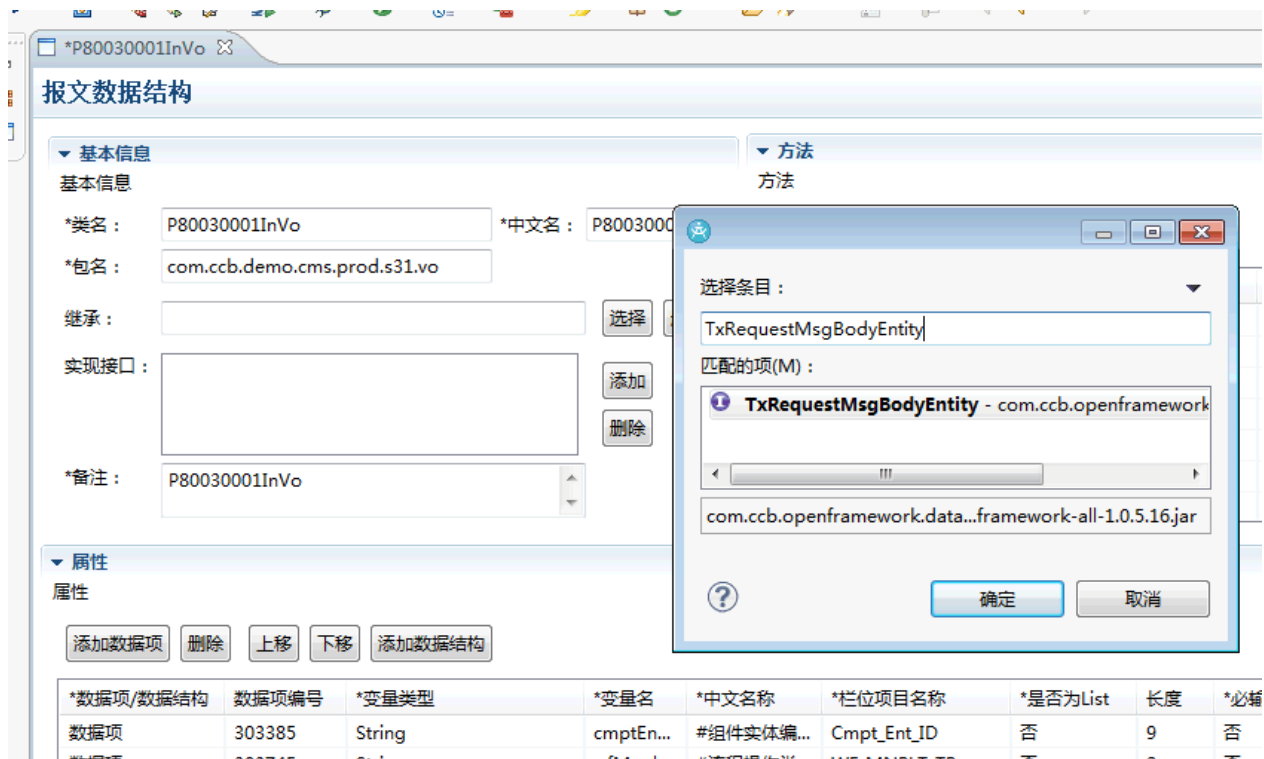


打开导入的报文数据结构文件并编辑。

依次双击数据结构文件打开。



为每个invo、outvo添加实现接口。invo添加TxRequestMsgBodyEntity 接口，outvo添加TxResponseMsgBodyEntity 接口，并保存。



修改各个数据结构的数据项中的变量类型，如果有两种类型皆可的情况（如：String或java.sql.Date），请修改为其中一种类型，确保每个数据项类型唯一。

添加数据项

删除

上移

下移

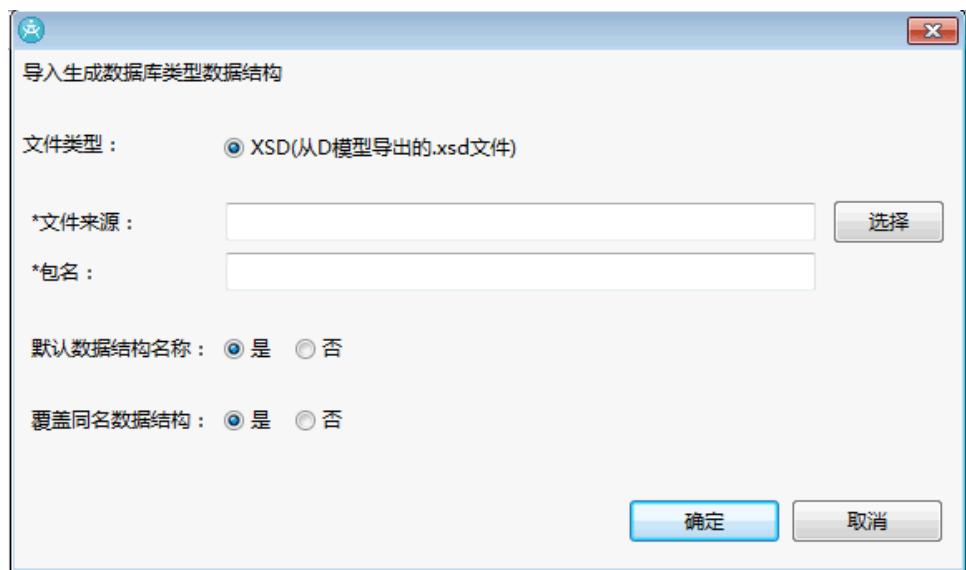
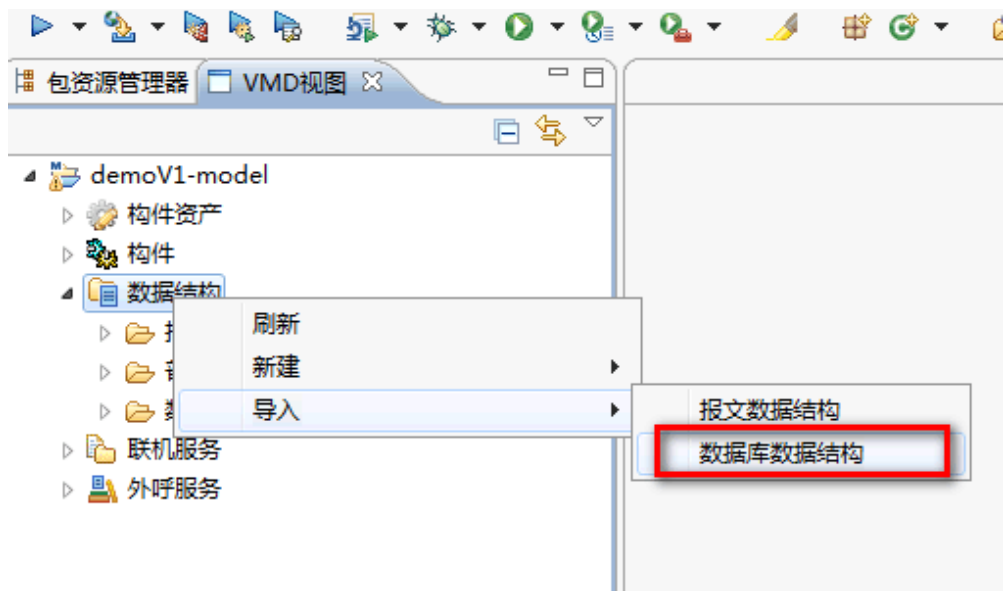
添加数据结构

*数据项/数据结构	数据项编号	*变量类型	*变量名	*中文名称	*栏位项目	*是否为List	长度	*必填
数据项	303385	String	cmptEn...	#组件实体编...	Cmpt_Ent...	否	9	否
数据项	303745	String	wfMnpl...	#流程操作类...	WF_MNPL...	否		否
数据项	119756	String	ccsttrId	现金客户树...	CCstTrID	否		否
数据项	119742	String	ccsttrnd...	现金客户树...	CCstTrNdID	否		否
数据项	303176	String	trgtCcst...	目标现金客...	Trgt_CCstTrNdID	否		否
数据项	120048	String	inraccB...	内部账业务...	InrAcc_Bsn_TpCd	否		否
数据项	120027	String	inraccL...	内部账额度...	InrAcc_Lmt_Char_...	否		否
数据项	600384	String	ccyCd	货币代码	Ccy_Cd	否		否
数据项	303155	java.math....	lmtQot	额度限额	Lmt_Qot	否		否
数据项	120055	short	inraccC...	内部账周期值	InrAcc_CycVal	否		否
数据项	120053	String	inraccC...	内部账周期...	InrAcc_Cyc_Unit	否		否
数据项	120145	String	cycRvlC...	周期循环类...	Cyc_Rvl_CgyCd	否		否

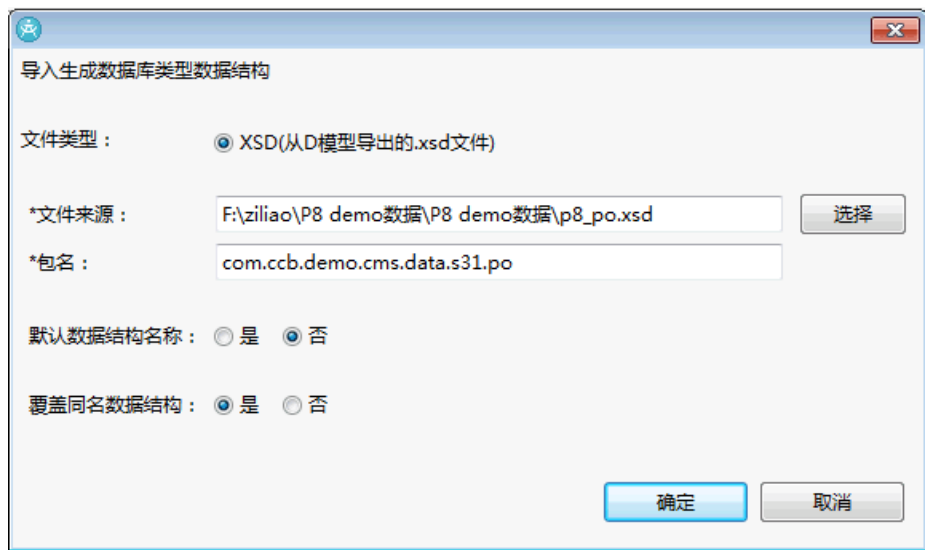
确保唯一

确保唯一

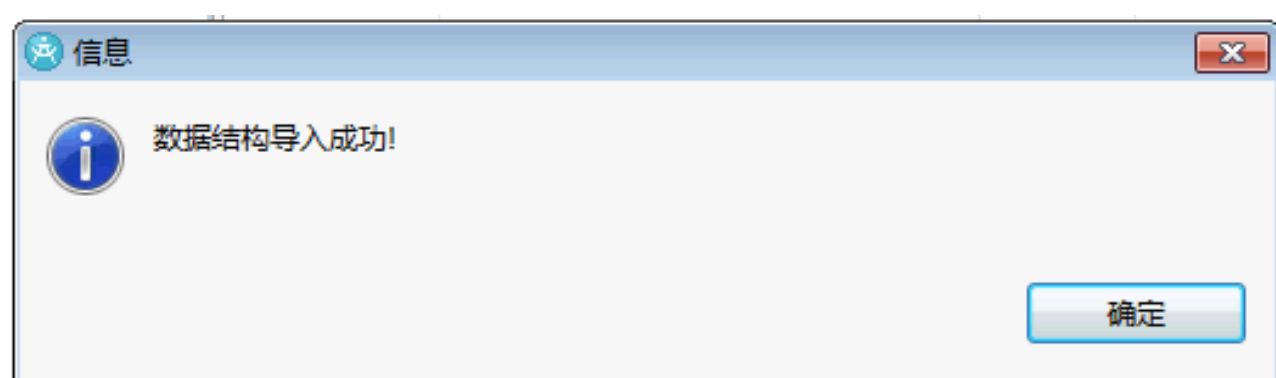
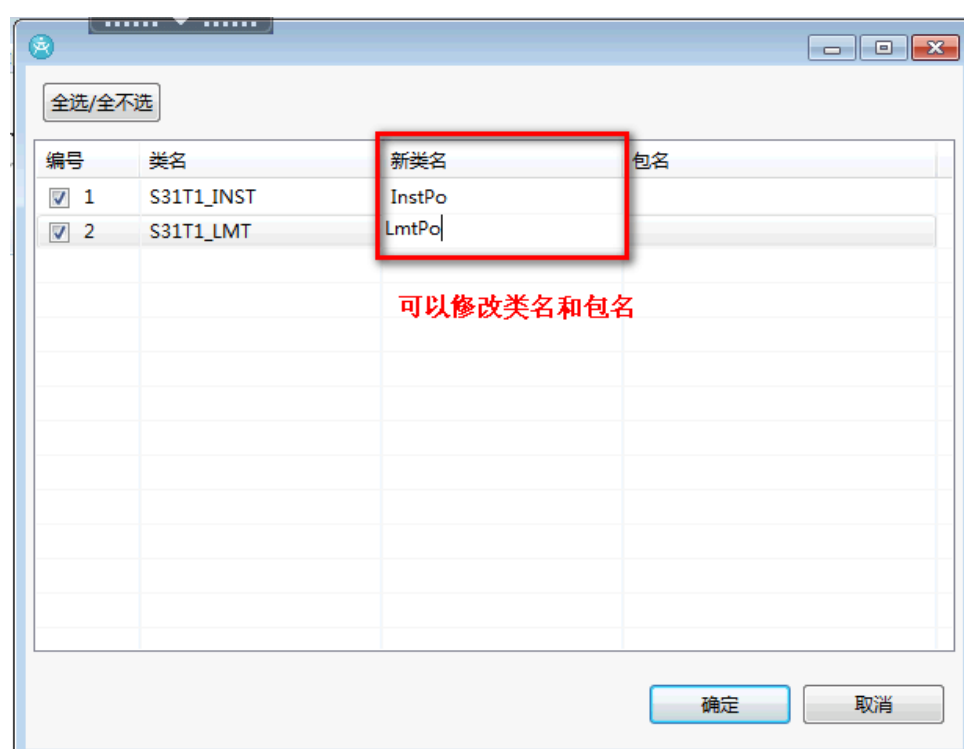
3.2.2 导入数据库数据结构



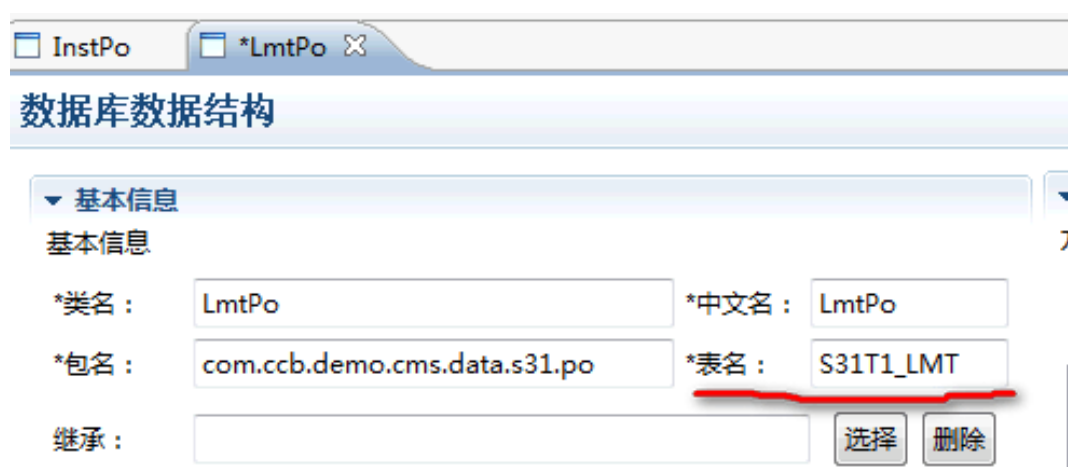
点击导入数据库数据结构，选择xsd文件，并输入包名；在“默认数据结构名称”，选“否”，进入修改窗口。



可以编辑设置“新类名”、“包名”。



导入成功后，依次打开导入的数据库数据结构文件，修改数据结构的表名。

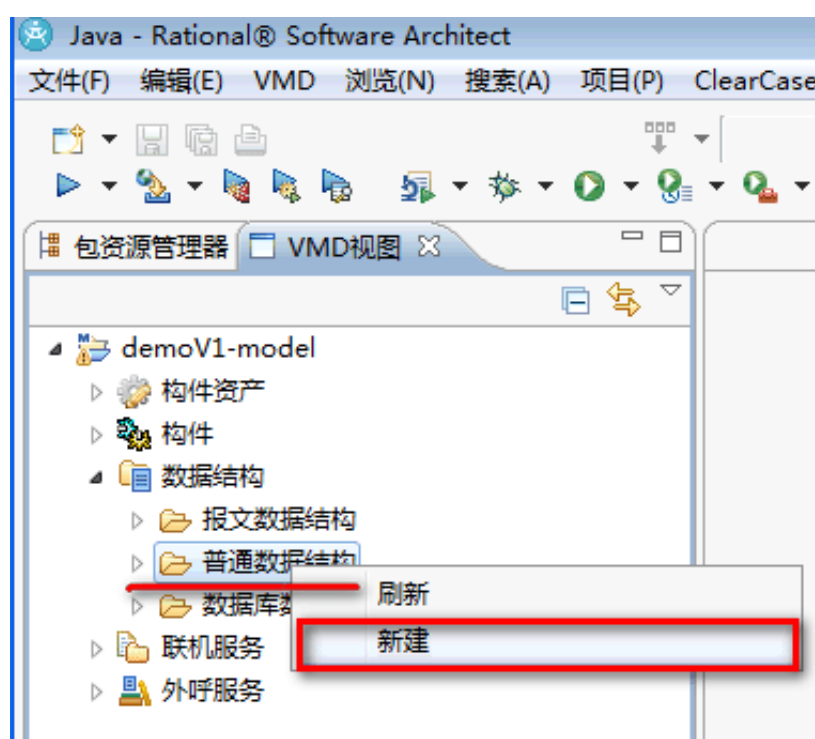


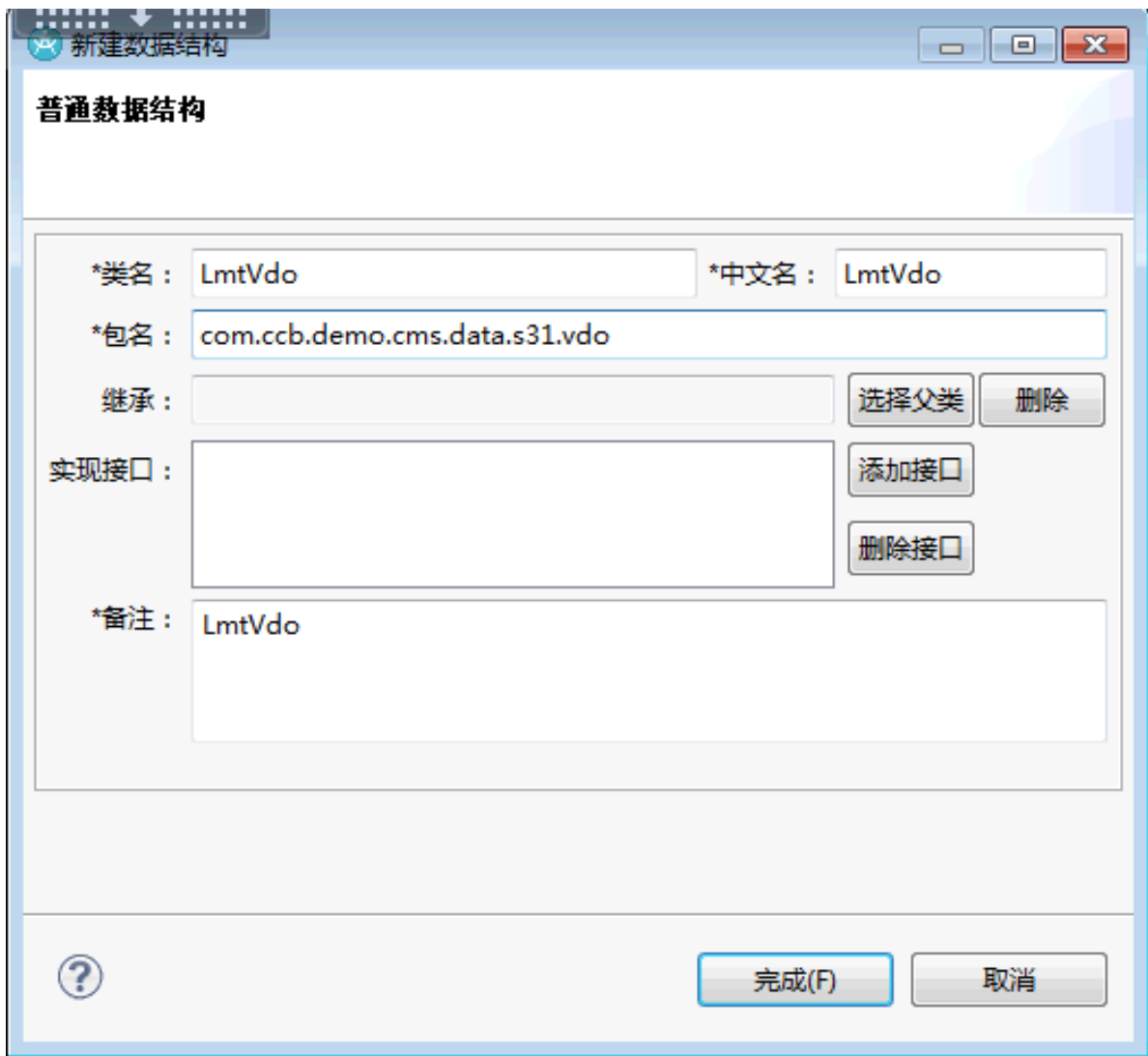
修改数据项的变量类型、Oracle数据格式，去掉或的情况，保留一种类型，并设置主键（可以根据建表语句，确定哪些字段为主键）。

InstPo						
*LmtPo						
删除						
*备注: S31T1_LMT						
属性						
属性						
添加数据项 删除 上移 下移						
*数据项/数据结构	数据项编号	*变量类型	*变量名	*中文名称	*Oracle数据格式	*主键
数据项	101673	String	exdat	到期日期	DATE	否
数据项	301521	String	multiTe...	#多实体标识	CHAR(5)	是
数据项	120028	java.math....	inraccL...	内部账额度值	NUMBER(19,2)	否
数据项	303665	String	inraccL...	#内部账额度...	CHAR(1)	否
数据项	120055	short	inraccC...	内部账周期值	NUMBER(4)	否
数据项	120144	java.math....	cinspls...	周期内剩余...	NUMBER(19,2)	否
数据项	120145	String	cycRvlC...	周期循环类...	CHAR(1)	否
数据项	102869	String	aspdEcd	可售产品编码	CHAR(8)	否
数据项	303378	String	dacVerf	#DAC校验	CHAR(16)	否
数据项	303383	String	rsrvFld...	#预留字段描...	CLOB	否
数据项	119756	String	ccsttrId	现金客户树...	CHAR(13)	是
数据项	119742	String	ccsttrnd...	现金客户树...	CHAR(25)	是

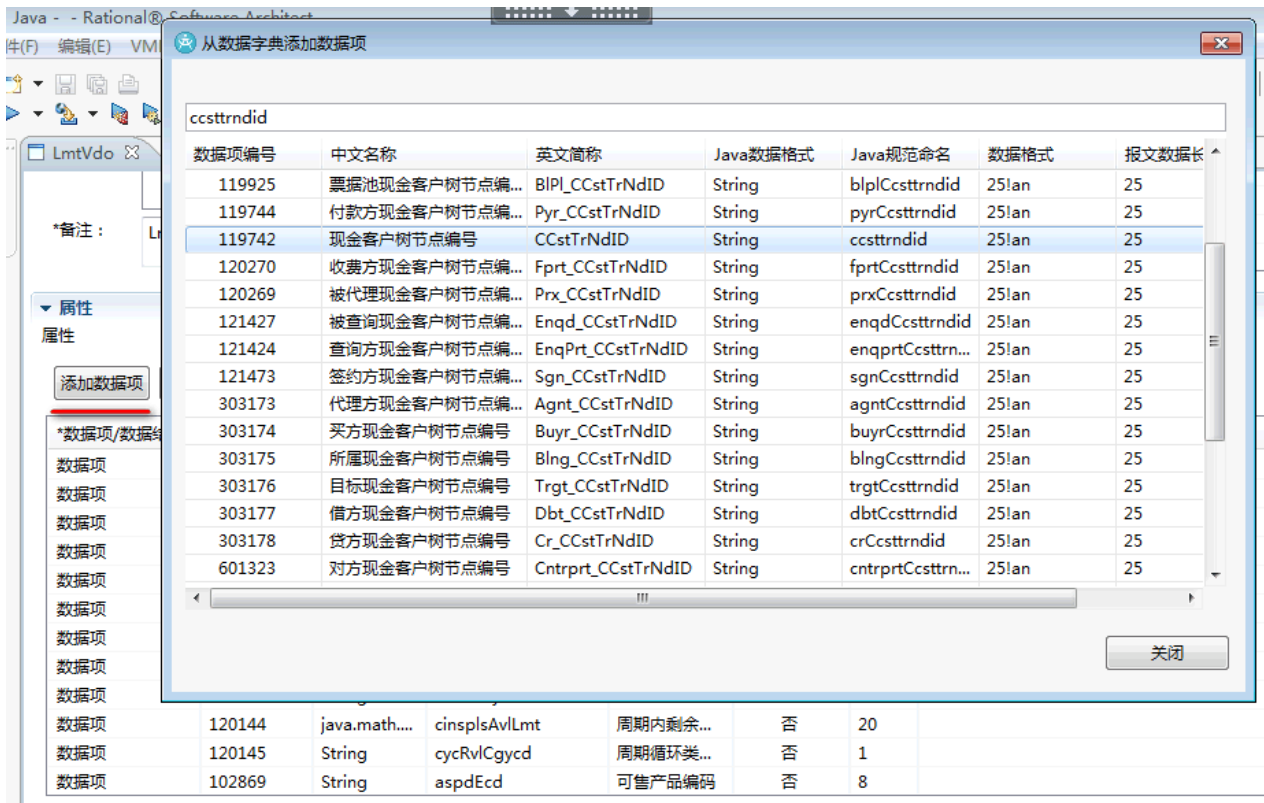
InstPo						
删除						
*备注: S31T1_INST						
属性						
属性						
添加数据项 删除 上移 下移						
*数据项/数据结构	数据项编号	*变量类型	*变量名	*中文名称	*Oracle数据格式	*主键
数据项	119742	String	ccsttrnd...	现金客户树...	CHAR(25)	是
数据项	302587	String	ccsttrnd...	#现金客户树...	CHAR(1)	否
数据项	119756	String	ccsttrId	现金客户树...	CHAR(13)	是
数据项	301521	String	multiTe...	#多实体标识	CHAR(5)	是

3.2.3 新建普通数据结构





打开新建的“普通数据结构”，点击“添加数据项”，显示如下窗口：



数据项的搜索条件有“数据项编号”、“中文名称”、“英文简称”、“Java规范命名”，双击“数据项”进行添加。

添加完数据项后，如下图所示。

LmtVdo

删除

*备注：

LmtVdo

属性

属性

添加数据项

删除

上移

下移

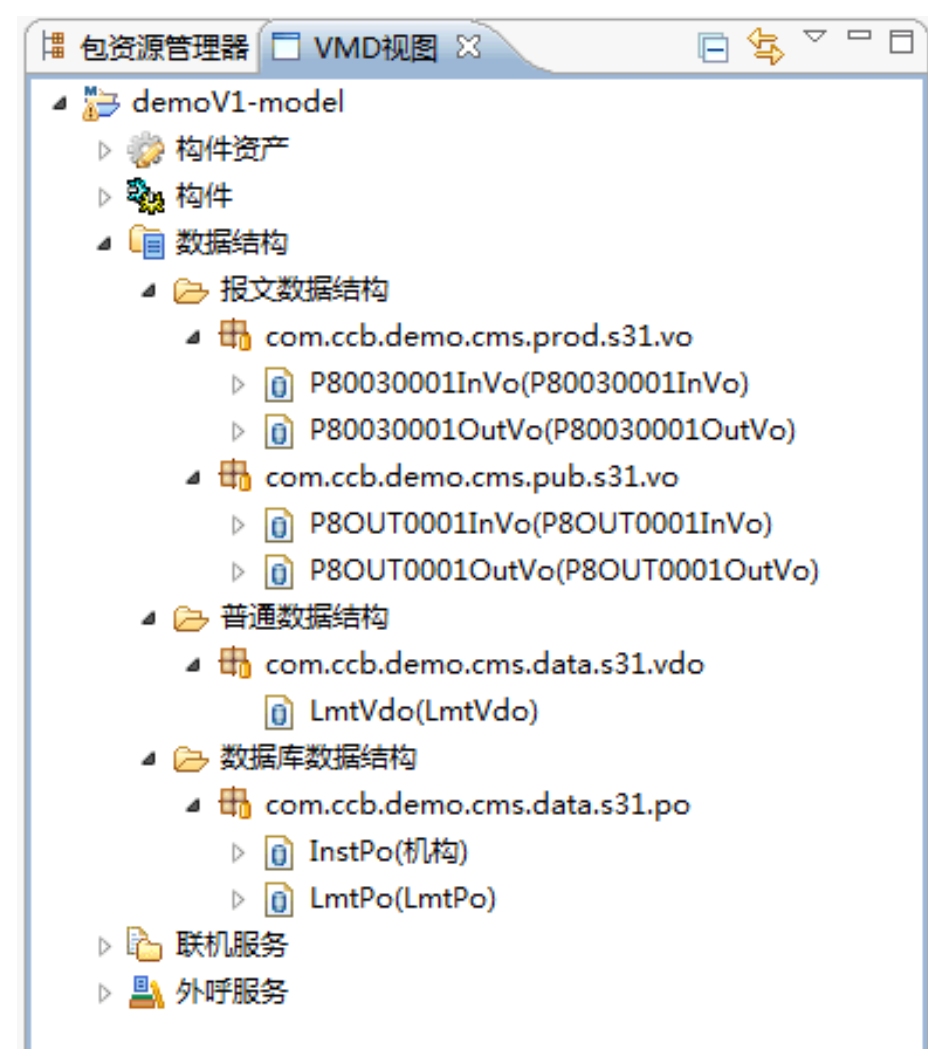
添加数据结构

*数据项/数据结构	数据项编号	*变量类型	*变量名	*中文名称	*是否为List	长度
数据项	119756	String	ccsttrId	现金客户树...	否	13
数据项	119742	String	ccsttrndid	现金客户树...	否	25
数据项	103891	String	ccycd	币种代码	否	3
数据项	301521	String	multiTenancyId	#多实体标识	否	5
数据项	120027	String	inraccLmtCharCd	内部账额度...	否	1
数据项	120028	java.math.BigDecimal	inraccLmtVal	内部账额度值	否	20
数据项	303665	String	inraccLmtSt	#内部账额度...	否	1
数据项	120055	Short	inraccCycval	内部账周期值	否	4
数据项	120053	String	inraccCycUnit	内部账周期...	否	2
数据项	120144	java.math.BigDecimal	cinsplsAvlLmt	周期内剩余...	否	20
数据项	120145	String	cycRvlCgycd	周期循环类...	否	1
数据项	102869	String	aspdEcd	可售产品编码	否	8

数据项	120048	String	inraccBsnTpcd	内部账业务...	否	2
数据项	303383	String	rsrvFldDsc	#预留字段描...	否	6000
数据项	302521	String	inraccBsnJrnlNo	#内部账业务...	否	32
数据项	119766	String	ccsttrndnm	现金客户树...	否	600

3.2.4 数据结构添加后的结果

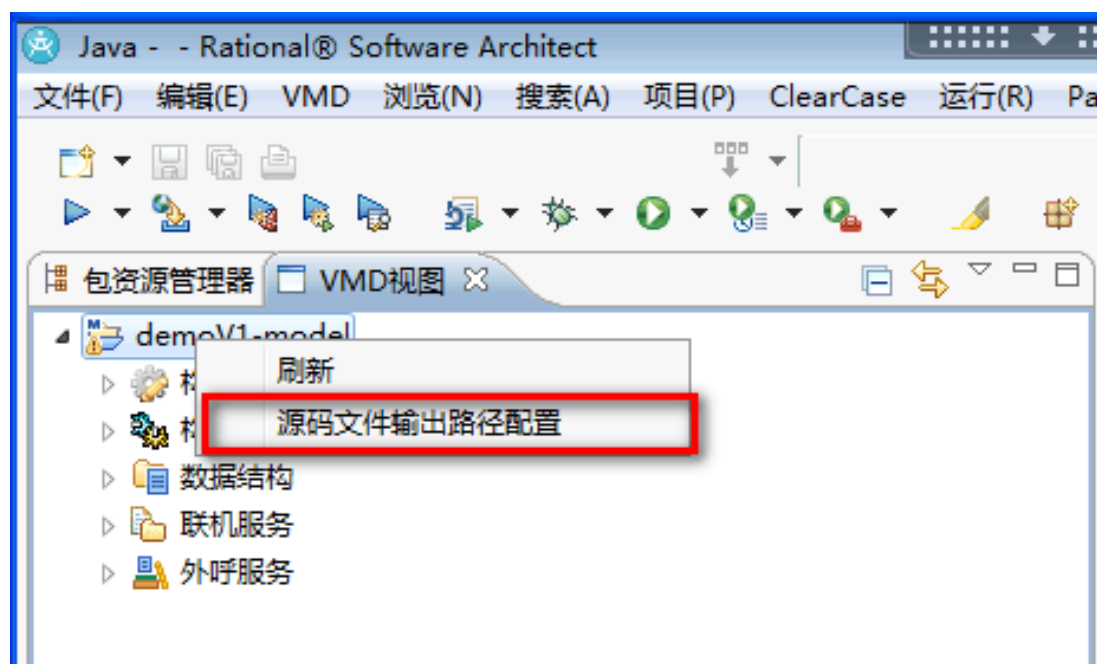
最后数据结构新建完毕的结果如下：



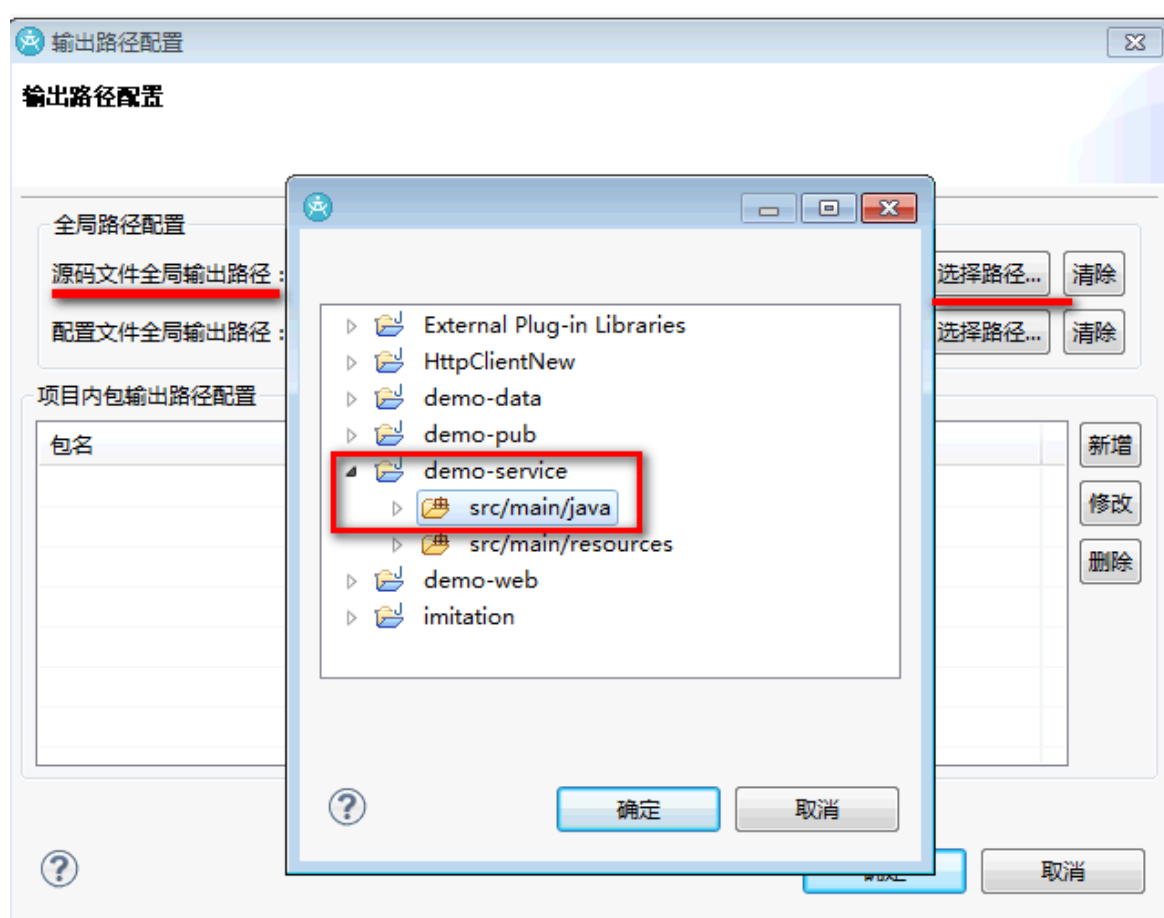
3.3 生成数据结构代码和 IBATIS 配置文件

3.3.1 配置源码输出路径

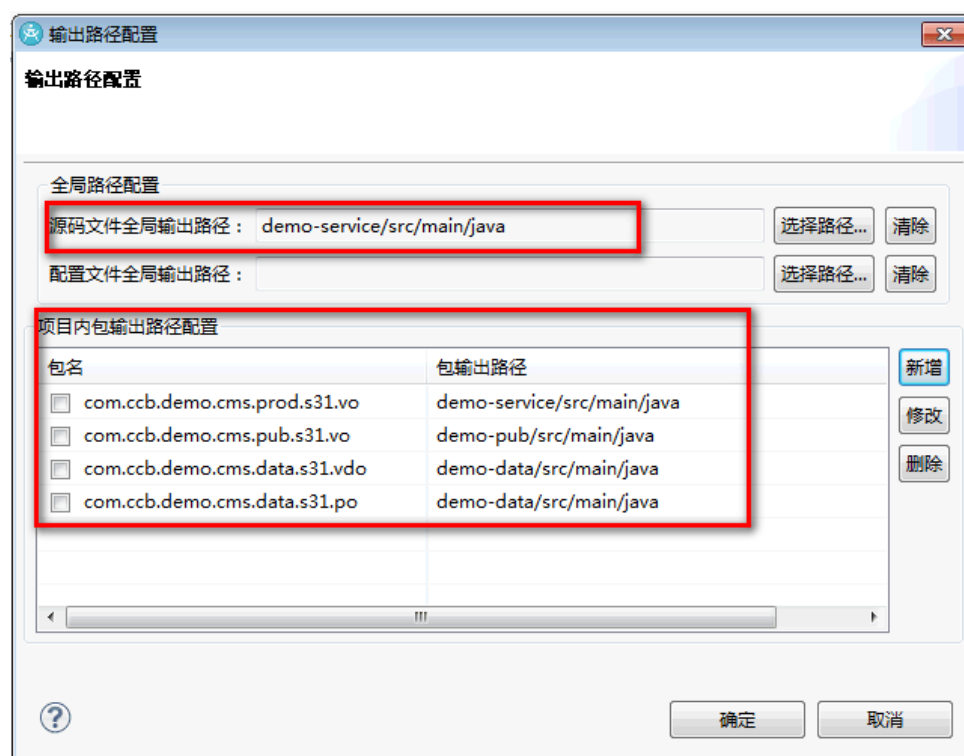
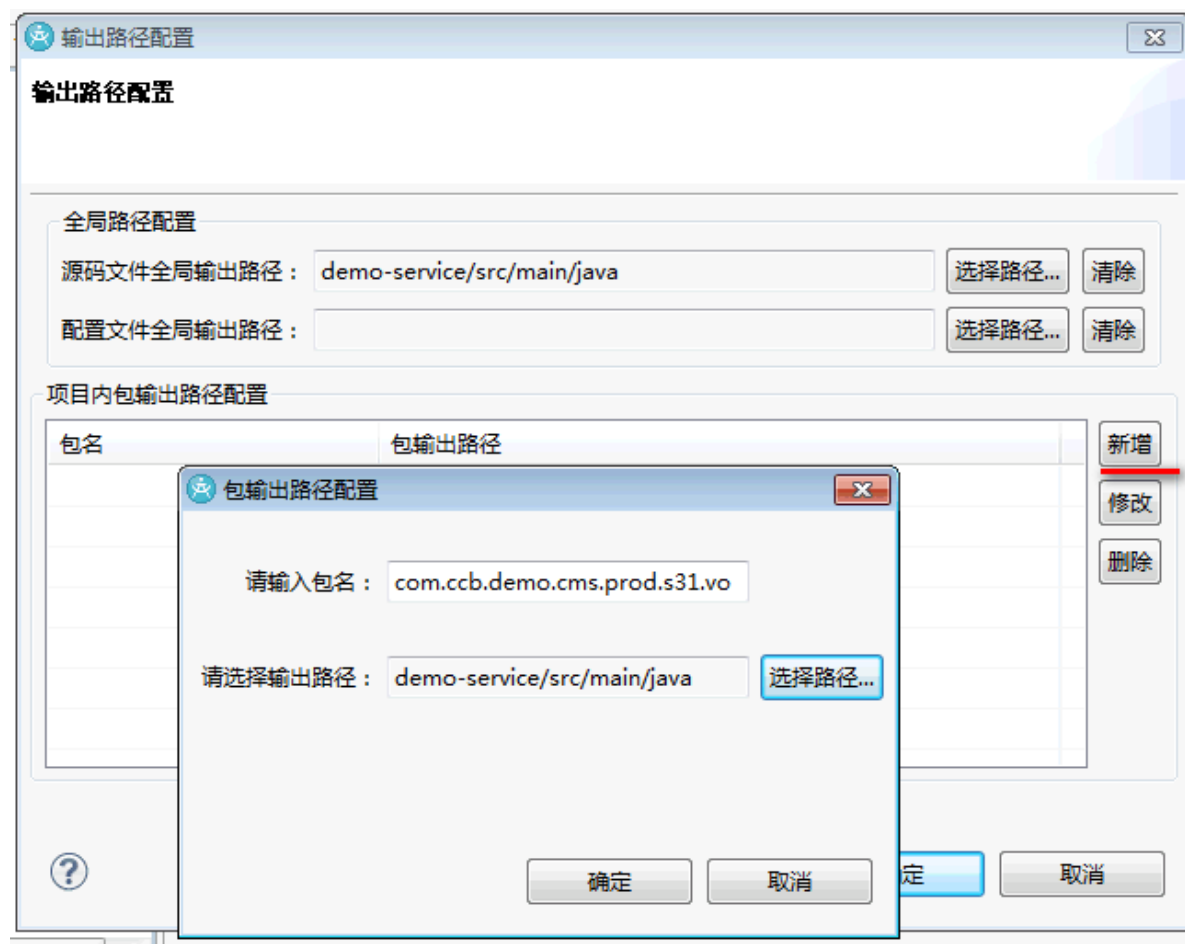
VMD项目右键，选择源码文件输出路径配置。



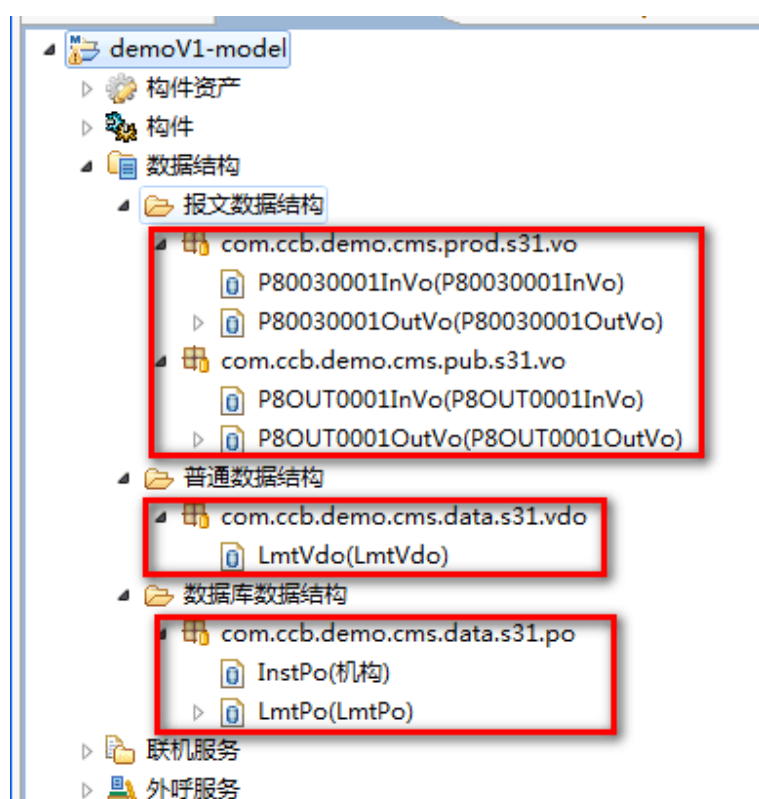
为项目配置全局输出路径，为：demo-service/src/main/java。

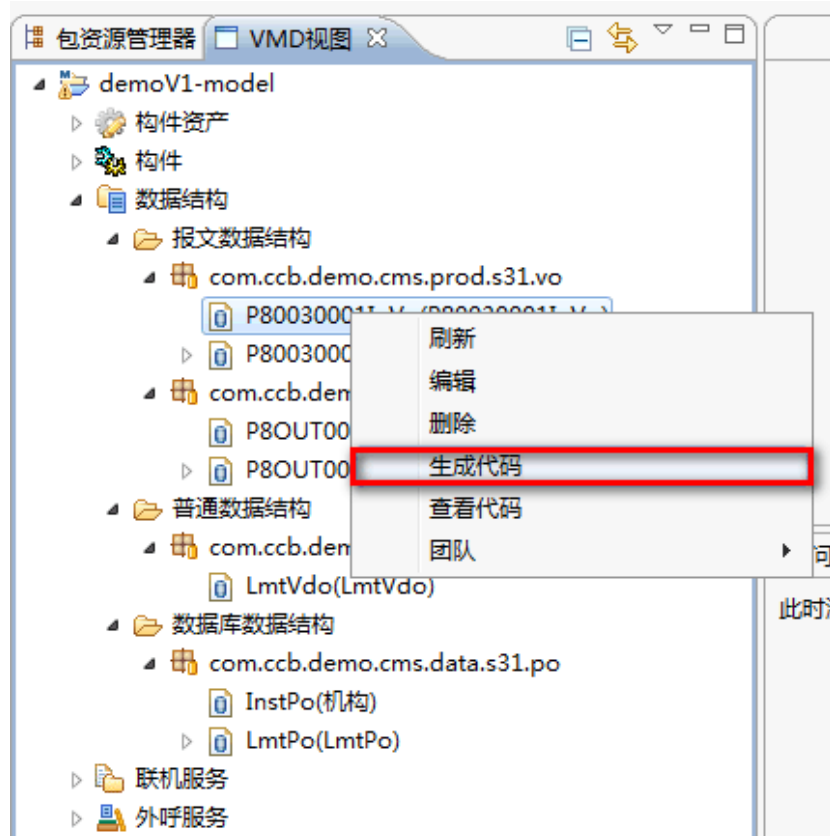


为项目内的不同包配置专属的输出路径。

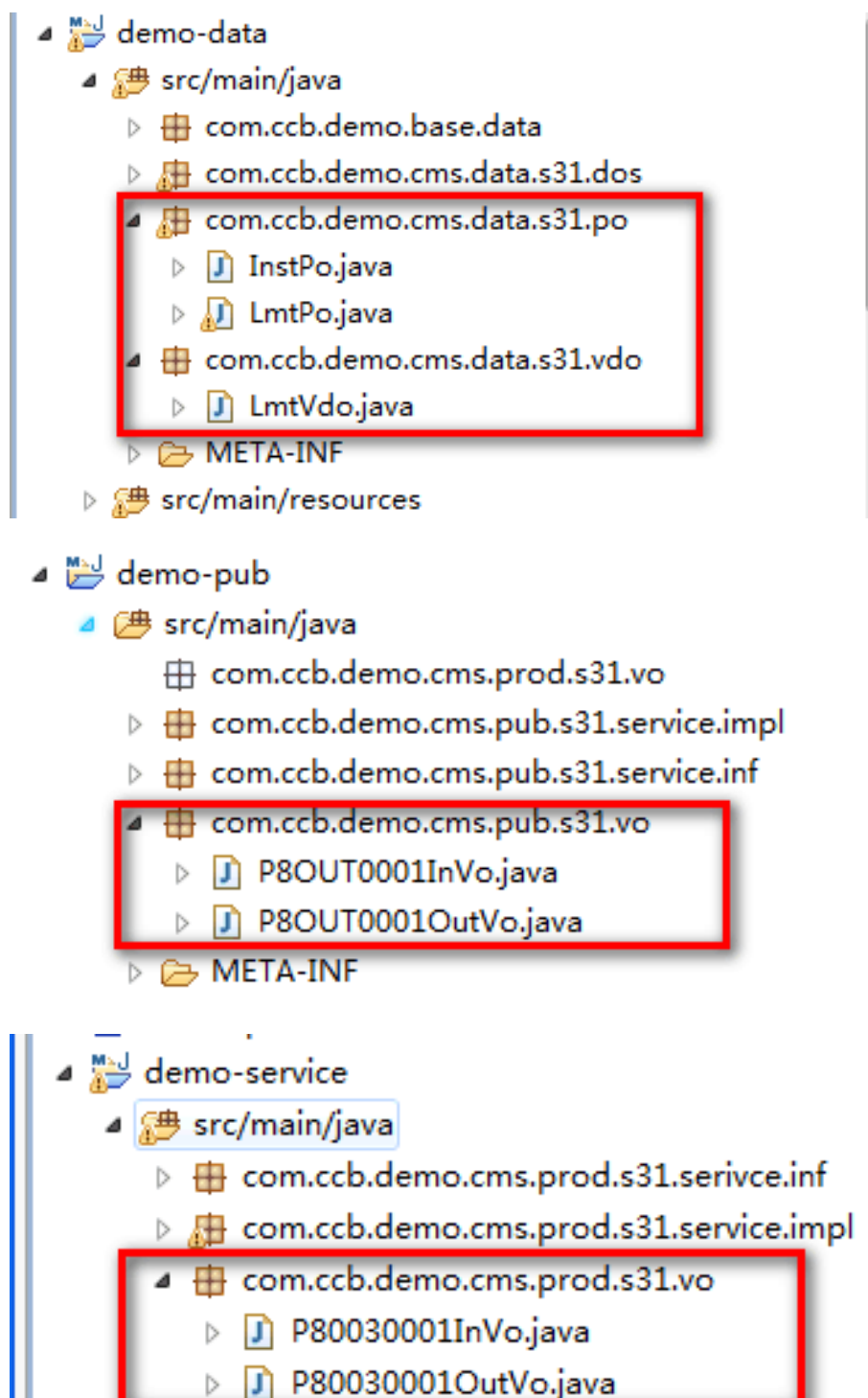


将以下的数据结构右键生成代码，点击右键，选择“生成代码”。





分别执行生成代码之后，查看生成代码。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/396231115212010230>