



Informatica®
10.1.1

Web 服务指南

Informatica Web 服务指南

10.1.1

2016 年 12 月

© 版权所有 Informatica LLC 2009, 2018

本软件和文档仅根据包含使用与披露限制的单独许可协议提供。未事先征得 Informatica LLC 同意，不得以任何形式、通过任何手段（电子、影印、录制或其他手段）复制或传播本文档的任何部分。

Informatica 和 Informatica 标志是 Informatica LLC 在美国和世界其他许多司法管辖区的商标或注册商标。欲获得 Informatica 商标的最新列表，请访问 <https://www.informatica.com/trademarks.html>。其他公司和产品名称可能是其各自所有者的商业名称或商标。

本软件和/或文档的某些部分受第三方版权制约，包括但不限于：版权所有 DataDirect Technologies。保留所有权利。版权所有 (C) Sun Microsystems。保留所有权利。版权所有 (C) RSA Security Inc. 保留所有权利。版权所有 (C) Ordinal Technology Corp. 保留所有权利。版权所有 (C) Aandacht c.v. 保留所有权利。版权所有 Genivia, Inc. 保留所有权利。版权所有 Isomorphic Software。保留所有权利。版权所有 (C) Meta Integration Technology, Inc. 保留所有权利。版权所有 (C) Intalio。保留所有权利。版权所有 (C) Oracle。保留所有权利。版权所有 (C) Adobe Systems Incorporated。保留所有权利。版权所有 (C) DataArt, Inc. 保留所有权利。版权所有 (C) ComponentSource。保留所有权利。版权所有 (C) Microsoft Corporation。保留所有权利。版权所有 (C) Rogue Wave Software, Inc. 保留所有权利。版权所有 (C) Teradata Corporation。保留所有权利。版权所有 (C) Yahoo! Inc. 保留所有权利。版权所有 (C) Glyph & Cog, LLC。保留所有权利。版权所有 (C) Thinkmap, Inc. 保留所有权利。版权所有 (C) Clearpace Software Limited。保留所有权利。版权所有 (C) Information Builders, Inc. 保留所有权利。版权所有 (C) OSS Nokalva, Inc. 保留所有权利。版权所有 Edifecs, Inc. 保留所有权利。版权所有 Cleo Communications, Inc. 保留所有权利。版权所有 (C) International Organization for Standardization 1986。保留所有权利。版权所有 (C) ej-technologies GmbH。保留所有权利。版权所有 (C) Jaspersoft Corporation。保留所有权利。版权所有 (C) International Business Machines Corporation。保留所有权利。版权所有 (C) yWorks GmbH。保留所有权利。版权所有 (C) Lucent Technologies。保留所有权利。版权所有 (C) University of Toronto。保留所有权利。版权所有 (C) Daniel Veillard。保留所有权利。版权所有 (C) Unicode, Inc. 版权所有 IBM Corp. 保留所有权利。版权所有 (C) MicroQuill Software Publishing, Inc. 保留所有权利。版权所有 (C) PassMark Software Pty Ltd. 保留所有权利。版权所有 (C) LogiXML, Inc. 保留所有权利。版权所有 (C) 2003-2010 Lorenzi Davide。保留所有权利。版权所有 (C) Red Hat, Inc. 保留所有权利。版权所有 (C) The Board of Trustees of the Leland Stanford Junior University。保留所有权利。版权所有 (C) EMC Corporation。保留所有权利。版权所有 (C) Flexera Software。保留所有权利。版权所有 (C) Jinfonet Software。保留所有权利。版权所有 (C) Apple Inc. 保留所有权利。版权所有 (C) Telerik Inc. 保留所有权利。版权所有 (C) BEA Systems。保留所有权利。版权所有 (C) PDFlib GmbH。保留所有权利。版权所有 (C) Orientation in Objects GmbH。保留所有权利。版权所有 (C) Tanuki Software, Ltd. 保留所有权利。版权所有 (C) Ricebridge。保留所有权利。版权所有 (C) Sencha, Inc. 保留所有权利。版权所有 (C) Scalable Systems, Inc. 保留所有权利。版权所有 (C) jQWidgets。保留所有权利。版权所有 (C) Tableau Software, Inc. 保留所有权利。版权所有 (C) MaxMind, Inc. 保留所有权利。版权所有 (C) TMate Software s.r.o. 保留所有权利。版权所有 (C) MapR Technologies Inc. 保留所有权利。版权所有 (C) Amazon Corporate LLC。保留所有权利。版权所有 (C) Highsoft。保留所有权利。版权所有 (C) Python Software Foundation。保留所有权利。版权所有 (C) BeOpen.com。保留所有权利。版权所有 (C) CNRI。保留所有权利。

本产品包括由 Apache Software Foundation (<http://www.apache.org/>) 开发的软件 and/或在不同 Apache 许可证版本（以下简称“许可证”）下许可的其他软件。您可从 <http://www.apache.org/licenses/> 获取这些许可证的副本。除非适用法律要求或者有相应书面协议，否则依据这些“许可证”分发的软件以“原样”提供，不附带任何明示或暗示的担保或条件。请参阅“许可证”中规定的具体语言管理权限和限制。

本产品包括由 Mozilla (<http://www.mozilla.org/>) 开发的软件、由 JBoss Group, LLC 开发的软件（版权所有 JBoss Group, LLC 保留所有权利）、由 Bruno Lowagie 和 Paulo Soares 开发的软件（版权所有 (C) 1999-2006 Bruno Lowagie 和 Paulo Soares）以及在 <http://www.gnu.org/licenses/lgpl.html> 网站上的不同版本 GNU Lesser General 公共许可协议下许可的软件。这些材料由 Informatica 按“原样”免费提供，不附带任何明示或暗示的担保，包括但不限于适销性和特定用途适用性的暗示担保。

本产品包括 ACE(TM) 和 TAO(TM) 软件，这些软件版权归 Douglas C. Schmidt 及其在华盛顿大学、加利福尼亚大学欧文分校以及范德堡大学的研发团队所有（版权所有 (C) 1993-2006，保留所有权利）。

本产品包括由 OpenSSL Project 开发并在 OpenSSL Toolkit（版权所有 OpenSSL Project。保留所有权利）中使用的软件，该软件的再分发受 <http://www.openssl.org> 和 <http://www.openssl.org/source/license.html> 上规定条款之制约。

本产品包括 Curl 软件，版权所有 1996-2013, Daniel Stenberg <daniel@haxx.se>。保留所有权利。有关该软件的权限和限制受 <http://curl.haxx.se/docs/copyright.html> 上规定条款之制约。允许出于任何目的以免费或收费形式使用、复制、修改和分发该软件，但前提是所有副本均应注明上述版权声明以及本许可声明。

本产品包括由 MetaStuff, Ltd. 开发的软件，版权所有 2001-2005 ((C)) MetaStuff, Ltd. 保留所有权利。有关该软件的权限和限制受 <http://www.dom4j.org/license.html> 上规定条款之制约。

本产品包括由 Dojo Foundation 开发的软件，版权所有 (C) 2004-2007, Dojo Foundation。保留所有权利。有关该软件的权限和限制受 <http://dojotoolkit.org/license> 上规定条款之制约。

本产品包括 ICU 软件，版权所有 International Business Machines Corporation 和其他方。保留所有权利。有关该软件的权限和限制受 <http://source.icu-project.org/repos/icu/icu/trunk/license.html> 上规定条款之制约。

本产品包括由 Per Bothner 开发的软件，版权所有 (C) 1996-2006 Per Bothner。保留所有权利。<http://www.gnu.org/software/kawa/Software-License.html> 上的许可证中规定了您使用这些材料的权利。

本产品包括 OSSP UUID 软件，版权所有 (C) 2002 Ralf S. Engelschall，版权所有 (C) 2002 OSSP Project，版权所有 (C) 2002 Cable & Wireless Deutschland。有关该软件的权限和限制受 <http://www.opensource.org/licenses/mit-license.php> 上规定条款之制约。

本产品包括由 Boost (<http://www.boost.org/>) 开发的软件或在 Boost 软件许可证下许可的软件。有关该软件的权限和限制受 http://www.boost.org/LICENSE_1_0.txt 上规定条款之制约。

本产品包括由 University of Cambridge 开发的软件，版权所有 (C) 1997-2007 University of Cambridge。有关该软件的权限和限制受 <http://www.pcre.org/license.txt> 上规定条款之制约。

本产品包括由 The Eclipse Foundation 开发的软件，版权所有 (C) 2007 The Eclipse Foundation。保留所有权利。有关该软件的权限和限制受 <http://www.eclipse.org/org/documents/epl-v10.php> 和 <http://www.eclipse.org/org/documents/edl-v10.php> 上规定条款之制约。

本产品包括在 <http://www.tcl.tk/software/tcltk/license.html>、<http://www.bosrup.com/web/overlib/?License>、<http://www.stlport.org/doc/license.html>、<http://asm.ow2.org/license.html>、<http://www.cryptix.org/LICENSE.TXT>、<http://hsqldb.org/web/hsqldbLicense.html>、<http://htpunittest.sourceforge.net/doc/license.html>、<http://jung.sourceforge.net/license.txt>、http://www.zip.org/zlib/zlib_license.html、<http://www.openldap.org/software/release/license.html>、<http://www.libssh2.org>、<http://slf4j.org/license.html>、<http://www.sente.ch/software/OpenSourceLicense.html>、<http://fusesource.com/downloads/licenses-agreements/fuse-message-broker-v-5-3-license-agreement>、<http://antlr.org/license.html>、<http://aopalliance.sourceforge.net/>、<http://www.bouncycastle.org/licence.html>、<http://www.jgraph.com/jgraphdownload.html>、<http://www.jcraft.com/jsch/LICENSE.txt>、http://jotm.objectweb.org/bsd_license.html、<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>、<http://www.slf4j.org/license.html>、<http://nanoxml.sourceforge.net/orig/copyright.html>、<http://www.json.org/en/jcalendar/license.html>、<http://forge.ow2.org/projects/jaservice/>、<http://www.postgresql.org/about/licence.html>、<http://www.sqlite.org/copyright.html>、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.jaxen.org/faq.html>、<http://www.jdom.org/docs/faq.html>、<http://www.slf4j.org/license.html>、<http://www.iodbc.org/dataspace/iodbc/wiki/ODBC/License>、<http://www.keplerproject.org/md5/license.html>、<http://www.toedter.com/en/jcalendar/license.html>、<http://www.edankert.com/bounce/index.html>、<http://www.net-snmp.org/about/license.html>、<http://www.openmdx.org/#FAQ>、http://www.php.net/license/3_01.txt、<http://srp.stanford.edu/license.txt>、<http://www.schneier.com/blowfish.html>、<http://www.jmock.org/license.html>、<http://xsom.java.net>、<http://benalman.com/about/license/>、<http://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>、<http://www.h2database.com/html/license.html#summary>、<http://jsoncpp.sourceforge.net/LICENSE>、<http://jdbc.postgresql.org/license.html>、<http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>、<https://github.com/>

rantav/hector/blob/master/LICENSE; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>、<http://jibx.sourceforge.net/jibx-license.html>、<https://github.com/lyokato/libgeohash/blob/master/LICENSE>、<https://github.com/hjiang/jsonxx/blob/master/LICENSE>、<https://code.google.com/p/lz4/>、<https://github.com/jedisct1/libsodium/blob/master/LICENSE>、<http://one-jar.sourceforge.net/index.php?page=documents&file=license>、<https://github.com/EsotericSoftware/kryo/blob/master/license.txt>、<http://www.scala-lang.org/license.html>、<https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>、<http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>、<https://aws.amazon.com/asl/>、<https://github.com/twbs/bootstrap/blob/master/LICENSE> 和 <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt> 下许可的软件。

本产品包括在 Academic 免费许可证 (<http://www.opensource.org/licenses/afl-3.0.php>)、通用开发和分发许可证 (<http://www.opensource.org/licenses/cddl1.php>)、通用公共许可证 (<http://www.opensource.org/licenses/cpl1.0.php>)、Sun Binary Code 许可协议补充许可条款、BSD 许可证 (<http://www.opensource.org/licenses/bsd-license.php>)、新 BSD 许可证 (<http://opensource.org/licenses/BSD-3-Clause>)、MIT 许可证 (<http://www.opensource.org/licenses/mit-license.php>)、Artistic 许可证 (<http://www.opensource.org/licenses/artistic-license-1.0>) 以及原始开发者公共许可证版本 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>) 下许可的软件。

本产品包括由 Joe Walnes 和 XStream Committers 开发的软件，版权所有 (C) 2003-2006 Joe Walnes，2006-2007 XStream Committers。保留所有权利。有关该软件的权限和限制受 <http://xstream.codehaus.org/license.html> 上规定条款之制约。本产品包括由 Indiana University Extreme! Lab 开发的软件。有关详细信息，请访问 <http://www.extreme.indiana.edu/>。

本产品包括软件版权所有 (c) 2013 Frank Balluffi 和 Markus Moeller。保留所有权利。有关此软件的权限和限制受 MIT 许可证上规定条款之制约。

请参阅位于以下位置的专利：<https://www.informatica.com/legal/patents.html>。

免责声明：Informatica LLC 以“原样”提供本文档，不附带任何明示或暗示的担保，包括但不限于非侵权、适销性或特定用途适用性的暗示担保。Informatica LLC 不保证本软件和文档中没有错误。本软件或文档中提供的信息可能包括技术上的不准确性或排字错误。本软件和文档中包含的信息随时可能更改，恕不另行通知。

声明

本 Informatica 产品（以下称“软件”）包括由 Progress Software Corporation 的运营公司 DataDirect Technologies（以下称“DataDirect”）提供的某些驱动程序（以下称“DataDirect 驱动程序”），受以下条款和条件制约：

1. DataDirect 驱动程序以“原样”提供，不附带任何明示或暗示的担保，包括但不限于适销性、特定用途适用性以及非侵权的暗示担保。
2. 在任何情况下，DataDirect 或其第三方供应商均不对最终用户客户承担因使用 ODBC 驱动程序而引起的任何直接、间接、偶发、特殊、继发或其他损害赔偿的责任，无论是否已提前告知该种损害的可能性。这些限制适用于所有诉因，包括但不限于违反合同、违反担保、过失、严格责任、虚假陈述以及其他侵权行为。

本文档中的信息如有更改，恕不另行通知。如果您发现本文档中存在任何问题，请以书面形式将问题报告给我们，邮寄地址是 Informatica LLC 2100 Seaport Blvd.Redwood City, CA 94063。

INFORMATICA LLC 按“原样”提供本文档中的信息，无任何明示或暗示的担保，包括但不限于任何适销性和特定用途适用性担保，也没有任何非侵权担保或条件。

发布日期: 2018-07-18

目录

前言	11
Informatica 资源	11
Informatica Network	11
Informatica 知识库	11
Informatica 文档	11
Informatica 产品可用性矩阵	12
Informatica Velocity	12
Informatica Marketplace	12
Informatica 全球客户支持部门	12
第 1 章：Web 服务.....	13
Web 服务概览	13
REST Web 服务和 SOAP Web 服务之间的差异	14
Web 服务过程	14
Web 服务使用者转换流程	14
第 2 章：SOAP Web 服务	15
SOAP Web 服务组件	15
操作	15
WSDL	16
SOAP	16
开发 SOAP Web 服务	16
SOAP Web 服务示例	17
第 3 章：WSDL 数据对象.....	19
WSDL 数据对象概览	19
WSDL 数据对象概览视图	20
WSDL 数据对象高级视图	20
导入 WSDL 数据对象	20
WSDL 同步	21
同步 WSDL 数据对象	21
证书管理	22
Informatica Developer 证书属性	22
将证书添加到 Informatica Developer	22
第 4 章：架构对象.....	23
架构对象概览	23
架构对象概览视图	23
架构文件	24
架构对象架构视图	24

命名空间属性.	24
元素属性.	25
简单类型属性.	26
复杂类型属性.	27
特性属性.	28
架构对象高级视图.	28
创建架构对象.	29
架构更新.	29
架构同步.	30
架构文件编辑.	30
证书管理.	32
Informatica Developer 证书属性.	33
将证书添加到 Informatica Developer.	33
第 5 章：如何创建 SOAP Web 服务.	34
创建 SOAP Web 服务概览.	34
类型和元素.	34
Web 服务概览视图.	35
Web 服务 WSDL 视图.	37
从 WSDL 数据对象创建 Web 服务.	37
步骤 1。从 WSDL 数据对象创建 Web 服务.	37
步骤 2.将操作添加到 Web 服务.	38
将 WSDL 数据对象与 Web 服务关联.	38
手动创建 SOAP Web 服务.	39
步骤 1. 手动创建 Web 服务.	39
步骤 2. 创建操作.	39
步骤 3. 创建元素.	42
步骤 4. 创建预定义故障.	43
步骤 5. 创建表头.	43
第 6 章：操作映射.	45
操作映射概览.	45
操作映射常规选项卡.	46
操作映射操作选项卡.	46
操作映射高级选项卡.	46
输入转换.	46
输入转换端口选项卡.	47
将操作输入映射到端口的规则和指南.	47
配置输入转换.	47
输出转换.	48
输出转换端口选项卡.	49
输出转换高级选项卡.	49
将端口映射至操作输出的规则和准则.	49

配置输出转换.	49
故障转换.	51
故障转换端口选项卡.	51
故障转换高级选项卡.	52
将端口映射到操作故障的规则和指导.	52
创建故障转换.	52
配置故障转换.	53
故障处理.	54
系统定义的故障.	55
预定义故障.	56
通用故障.	56
测试操作映射.	57
测试操作映射.	57
自定义视图选项.	57
 第 7 章： 解析 Web 服务 SOAP 消息.	 59
解析 Web 服务 SOAP 消息概览.	59
转换用户界面.	59
多次出现输出配置.	60
规范化的关系输出.	61
生成的键.	61
非规范化的关系输出.	62
已转换的关系输出.	62
解析 anyType 元素.	62
解析派生类型.	63
解析 QName 元素.	64
解析置换组.	65
解析 SOAP 消息中的 XML 构造.	65
选项元素.	65
列表元素.	65
联合元素.	65
 第 8 章： 生成 Web 服务 SOAP 消息.	 66
生成 Web 服务 SOAP 消息概览.	66
转换用户界面.	67
输入端口区域.	67
操作区域.	68
端口和层次结构级别关系.	68
键.	69
映射端口.	70
映射端口.	71
映射组.	71
映射多个端口.	71

转换多次出现的端口	72
映射非规范化数据	73
派生类型和元素置换	74
生成派生类型	74
生成 anyType 元素和属性	75
生成置换组	75
生成 SOAP 消息中的 XML 构造	75
选项元素	76
列表元素	76
联合元素	77
第 9 章：Web 服务使用者转换	78
Web 服务使用者转换概览	78
SOAP 消息	78
WSDL 文件	79
操作	79
Web 服务安全	79
WSDL 选择	80
Web 服务使用者转换端口	81
HTTP 表头输入端口	81
其他输入端口	82
Web 服务使用者转换输入映射	82
将输入端口映射到节点的规则和准则	83
自定义视图选项	83
将输入端口映射到操作输入	84
Web 服务使用者转换输出映射	85
将节点映射到输出端口的规则和准则	86
将 SOAP 消息映射为 XML	86
自定义视图选项	86
将操作输出映射到输出端口	87
Web 服务使用者转换高级属性	87
Web 服务错误处理	89
消息压缩	90
并发	90
筛选器优化	91
启用通过 Web 服务使用者转换执行早期选择优化	91
通过 Web 服务使用者转换执行推入优化	91
创建 Web 服务使用者转换	93
Web 服务使用者转换示例	94
输入文件	94
逻辑数据对象模型	94
逻辑数据对象映射	95
Web 服务使用者转换	95

第 10 章： REST Web 服务	97
REST Web 服务概览.	97
REST Web 服务过程.	98
Web 服务使用者转换流程.	98
REST Web 服务资源.	98
REST Web 服务架构视图.	100
数据对象同步.	100
资源键.	101
资源映射.	101
默认资源映射.	101
自定义资源映射.	102
REST Web 服务输出转换.	103
REST 输出转换中多次出现的数据.	104
请求消息.	104
筛选资源映射中的数据.	105
按键搜索.	106
响应消息格式.	106
响应数据预览.	107
第 11 章： 如何创建 REST Web 服务	109
创建 REST Web 服务.	109
如何手动创建 REST Web 服务.	110
示例 REST Web 服务.	110
步骤 1. 创建 REST Web 服务资源.	110
创建 REST Web 服务资源.	110
步骤 2. 定义资源映射.	113
定义资源映射.	114
步骤 3. 配置输出映射.	116
配置输出映射.	117
步骤 4. 在“数据查看器”视图中测试映射.	118
按资源 ID 筛选输出.	118
按筛选条件筛选输出.	119
步骤 5. 部署应用程序.	120
部署应用程序.	120
步骤 6. 从浏览器查询 Web 服务.	121
查询 Web 服务.	122
如何从数据对象创建 REST Web 服务.	123
如何将数据对象部署为 REST Web 服务.	126
第 12 章： REST Web 服务使用者转换.....	129
REST Web 服务使用者转换概览.	129
REST Web 服务使用者转换流程.	130

REST Web 服务使用者转换配置.	130
消息配置.	131
资源标识.	131
HTTP 方法.	132
HTTP Get 方法.	132
HTTP Post 方法.	133
HTTP Put 方法.	133
HTTP Delete 方法.	134
REST Web 服务使用者转换端口.	134
输入端口.	135
输出端口.	135
传递端口.	135
参数端口.	135
URL 端口.	135
HTTP 表头端口.	136
Cookie 端口.	136
输出 XML 端口.	136
响应代码端口.	136
REST Web 服务使用者转换输入映射.	136
将输入端口映射到元素的规则和指南.	137
将输入端口映射到方法输入.	137
REST Web 服务使用者转换输出映射.	138
将元素映射到输出端口的规则和准则.	139
自定义视图选项.	139
将方法输出映射到输出端口.	139
REST Web 服务使用者转换高级属性.	140
REST Web 服务使用者转换创建.	141
创建 REST Web 服务使用者转换.	141
 第 13 章：管理 REST Web 服务和 SOAP Web 服务.	 142
Web 服务管理概览.	142
Web 服务属性配置.	142
Web 服务属性.	143
Web 服务操作和资源属性.	145
Web 服务结果集缓存.	145
Web 服务安全管理.	146
Web 服务权限.	147
SOAP 请求中的用户名标志.	148
Web 服务日志.	150
Web 服务跟踪级别.	150
Web 服务监视.	150
Web 服务的“属性”视图.	151
Web 服务的“报告”视图.	151

REST 或 SOAP Web 服务的“操作”视图.	151
Web 服务的“请求”视图.	151
附录 A: 数据类型兼容性.	153
数据类型引用概览.	153
XML 和转换数据类型.	153
小数.	155
索引.	156

前言

《Informatica Web 服务指南》是面向数据质量和数据服务开发人员编写的。本指南假定您已了解 Web 服务的概念。

Informatica 资源

Informatica Network

Informatica Network 囊括了 Informatica 全球客户支持部门、Informatica 知识库和其他产品资源。要访问 Informatica Network，请访问 <https://network.informatica.com>。

成员可以执行以下操作：

- 在一个位置访问您的所有 Informatica 资源。
- 在知识库中搜索文档、常见问题和最佳实践等产品资源。
- 查看产品可用性信息。
- 查看支持案例。
- 查找当地的 Informatica 用户组网络并与您的伙伴进行协作。

Informatica 知识库

使用 Informatica 知识库可在 Informatica Network 中搜索文档、入门知识文章、最佳实践和 PAM 等产品资源。

要访问知识库，请访问 <https://kb.informatica.com>。如果您对知识库有任何疑问、意见或建议，请与 Informatica 知识库团队联系，电子邮件地址为 KB_Feedback@informatica.com。

Informatica 文档

要获取有关产品的最新文档，请浏览 Informatica 知识库，网址为 https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx。

如果您对此文档有任何疑问、意见或建议，请与 Informatica 文档团队联系，电子邮件地址为 infa_documentation@informatica.com。

Informatica 产品可用性矩阵

产品可用性矩阵 (PAM) 指明了产品版本支持的操作系统版本、数据库以及其他类型的数据源和目标。如果您是 Informatica Network 成员，您可以访问 PAM，网址为 <https://network.informatica.com/community/informatica-network/product-availability-matrices>。

Informatica Velocity

Informatica Velocity 收集了 Informatica 专业服务开发的一系列提示和最佳实践。Informatica Velocity 基于数以百计的数据管理项目的实际经验而开发，汇集了我们曾在世界各地组织就职的顾问在成功规划、开发、部署和维护数据管理解决方案方面的知识。

如果您是 Informatica Network 成员，您可以访问 Informatica Velocity 资源，网址为 <http://velocity.informatica.com>。

如果您对 Informatica Velocity 有任何疑问、意见或建议，请通过 ips@informatica.com 与 Informatica 专业服务联系。

Informatica Marketplace

Informatica Marketplace 是一个论坛，该论坛中提供的解决方案可补充、扩展或增强您的 Informatica 实现。您可以利用 Informatica 开发人员和合作伙伴提供的数以百计解决方案中的任何方案，提高生产率，加快项目的实现时间。您可以访问 Informatica Marketplace，网址为 <https://marketplace.informatica.com>。

Informatica 全球客户支持部门

您可以通过电话或 Informatica Network 上的联机支持与全球支持中心联系。

要查找您当地的 Informatica 全球客户支持部门电话号码，请访问 Informatica 网站，链接为：
<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>。

如果您是 Informatica Network 成员，您可以使用联机支持，网址为 <http://network.informatica.com>。

第 1 章

Web 服务

本章包括以下主题：

- [Web 服务概览, 13](#)
- [REST Web 服务和 SOAP Web 服务之间的差异, 14](#)
- [Web 服务过程, 14](#)
- [Web 服务使用者转换流程, 14](#)

Web 服务概览

Web 服务客户端可以连接到 Informatica Web 服务以访问、转换或交付数据。外部应用程序或 Web 服务使用者转换可作为 Web 服务客户端连接到 Web 服务。可以在 Developer tool 中创建 Informatica Web 服务。

Web 服务可以处理信息请求、数据更新请求或任务执行请求。例如，Web 服务客户端发送了运行 Web 服务操作的请求。该 Web 服务客户端在请求中传递了一个客户 ID。Web 服务将检索客户信息和订单信息，并在响应中将信息返回到客户端。

Informatica Web 服务使用简单对象访问协议 (Simple Object Access Protocol, SOAP) 或具象状态传输 (Representational State Transfer, REST) 消息传递协议与 Web 服务客户端通信。

可以在 Developer tool 中创建以下类型的 Web 服务或 Web 服务客户端：

SOAP Web 服务

使用 SOAP 协议的 Web 服务。Web 服务客户端请求和 Web 服务响应都属于 SOAP 消息。Web 服务描述语言 (Web Service Description Language, WSDL) 是一种基于 XML 的接口定义语言，用于描述 Web 服务的功能。WSDL 文件包含有关如何调用 Web 服务、Web 服务需要哪些参数以及 Web 服务将返回哪些数据结构的说明。可以从 WSDL 文件创建 Informatica SOAP Web 服务。

SOAP Web 服务使用者转换

作为 Web 服务客户端连接到 Web 服务，以访问或转换位于映射中游的数据。可以从 WSDL 创建 SOAP Web 服务使用者转换。

REST Web 服务

接收 HTTP 请求以执行 Web 服务操作的 Web 服务。Informatica REST Web 服务可以接收 HTTP 请求以执行 GET 操作。Informatica REST Web 服务可以在 JSON 文件或 XML 文件中返回响应。

REST 使用者转换

作为 Web 服务客户端连接到 REST Web 服务，以访问或转换位于映射中游的数据。REST Web 服务使用者转换通过在转换、HTTP 连接或 HTTPS 连接中定义的 URL 连接到 Web 服务。请求消息和响应消息包含 XML 或 JSON 数据。

REST Web 服务和 SOAP Web 服务之间的差异

可以在 Informatica Developer tool 中创建 REST Web 服务或 SOAP Web 服务。

REST Web 服务和 SOAP Web 服务具有以下差异：

请求消息格式

SOAP 消息是结构化的 XML。SOAP Web 服务可解析该 XML 以确定 Web 服务必须执行的操作。REST 请求是一个简单的 URI 字符串，其中包含一个查询。

响应消息格式

SOAP Web 服务会按照 WSDL 中的定义以 XML 格式返回响应。

Informatica REST Web 服务会返回 JavaScript 对象表示法 (JavaScript Object Notation, JSON) 或 XML 响应消息。响应消息格式不由 WSDL 或架构定义。可以在定义 Informatica REST Web 服务时定义输出格式。

Web 服务映射格式

Informatica SOAP Web 服务包含操作映射。SOAP 操作映射包含一个输入转换，该输入转换将解析来自请求消息的 XML。必须向 Web 服务映射添加转换，以按照客户端请求的要求处理数据。

Informatica REST Web 服务包含资源映射。该资源映射不会读取请求查询。REST 资源映射包含读取转换而非输入转换。读取转换将读取模型存储库中的数据对象，以检索数据并将其返回到客户端。默认情况下，不必添加用于根据客户端查询来检索数据的筛选器转换或查找转换。REST Web 服务会在映射返回数据之后筛选输出数据。

Web 服务过程

Web 服务接收来自 Web 服务客户端的请求。

以下过程描述了数据集成服务如何处理来自 Web 服务客户端的 Web 服务请求：

1. 数据集成服务接收来自 Web 服务客户端的请求。
2. 数据集成服务的 Web 服务模块（或数据集成服务的 REST Web 服务模块）通过运行映射来处理请求。
3. Web 服务模块或 REST Web 服务模块向 Web 服务客户端发送响应。

Web 服务使用者转换流程

外部应用程序或 Web 服务使用者转换可作为 Web 服务客户端连接到 Web 服务。

以下过程描述 Web 服务使用者转换如何发送请求以及接收来自 Web 服务的响应：

1. Web 服务使用者转换生成请求，并通过连接对象连接到 Web 服务。
2. Web 服务使用者转换收到来自 Web 服务的响应。
3. Web 服务使用者转换从该响应提取数据，并在转换输出端口中返回数据。

第 2 章

SOAP Web 服务

本章包括以下主题：

- [SOAP Web 服务组件, 15](#)
- [开发 SOAP Web 服务, 16](#)
- [SOAP Web 服务示例, 17](#)

SOAP Web 服务组件

SOAP Web 服务组件定义 Web 服务的用途以及 Web 服务客户端如何与 Web 服务通信。

Web 服务具有以下组件：

操作

一个 Web 服务可以具有一个或更多操作。每个操作都与 Web 服务中的一个操作相对应。

Web 服务描述语言 (WSDL)

WSDL 是一个 XML 文档，用于描述 Web 服务操作的协议、格式和签名。

简单对象访问协议 (SOAP)

SOAP 是 Web 服务所使用的通信协议。

操作

对于自身支持的每个行为，Web 服务都包含一项操作。

例如，Web 服务可以包含名为 getcustomerid 的操作，用于接收客户名称并使用客户详细信息进行响应。操作输入包含客户名称的元素。操作输出包含基于客户名称的客户详细信息的元素。

在 Developer tool 中定义操作后，可以定义操作组件。操作具有以下组件：

操作输入和输出

操作输入为操作定义了 SOAP 请求中的元素。操作输出为操作定义了 SOAP 响应中的元素。

操作输入和操作输出可以包含表头。表头接收或发送 SOAP 消息中的数据。表头定义了 SOAP 请求或 SOAP 响应的表头中的元素。

操作故障

操作故障定义了可能作为操作结果的输出的错误消息的消息格式。可以为一个操作定义多个操作故障。

必须为每个操作配置操作映射。操作输入、操作输出和每个操作故障均与操作映射中的转换相对应。

WSDL

WSDL 是一种 XML 架构，用于描述 Web 服务操作的协议、格式和签名。

WSDL 包含要传递给 Web 服务的数据的说明，以便服务请求的发送方和接收方均能理解所交换的数据。WSDL 的元素包含要对数据执行的操作的说明，以便消息的接收者知道如何处理操作。WSDL 的元素还包含协议或传输绑定，以便消息的发送方知道该如何发送。

可以在 Developer tool 或 Administrator 工具中查看 Web 服务的 WSDL。向数据集成服务部署 Web 服务后，查看 WSDL URL 或者可以将 WSDL 下载到文件中。访问 Administrator 工具所显示的 WSDL URL 时，可以看到 WSDL 的内容。

SOAP

SOAP 是 Web 服务所使用的通信协议。其定义 Web 服务请求、响应和故障消息的格式。数据集成服务可以通过文档/文字编码处理 SOAP 1.1 和 SOAP 1.2 消息。

SOAP 消息包含以下各部分：

SOAP 信封

信封定义消息的框架、消息的内容以及处理消息的对象。

SOAP 表头

表头标识发送 SOAP 消息的实体。其包括身份验证信息。还包括 SOAP 消息的处理方法的信息。

SOAP 主体

主体是在客户端与 Web 服务提供程序之间相互传递的数据的容器。

SOAP 消息为 XML。如果 SOAP 消息包含多次出现的元素，则元素组将形成 XML 层次结构中的级别。如果一个级别嵌套在另一个级别内，则这些组都相互关联。

SOAP 请求消息可以包含层次结构数据。例如，客户端发送一个将客户订单添加到销售数据库中的请求。客户端在 SOAP 请求消息中传递两组数据。一组数据包含客户 ID 和名称，另一组数据包含订单信息。订单信息将出现多次。

SOAP 响应消息可以包含层次结构数据。例如，Web 服务客户端生成一条关于客户订单的 SOAP 请求。Web 服务将在 SOAP 响应中返回订单表头和多次出现的订单详细信息元素。

开发 SOAP Web 服务

开发 SOAP Web 服务，以提供一个 Web 服务客户端可用来执行操作的界面。Web 服务客户端可以是外部 Web 服务客户端或 Web 服务使用者转换。例如，Web 服务客户端可以连接到 Web 服务，以便根据客户名称或客户 ID 来查看客户详细信息。

完成以下步骤来开发 Web 服务：

1. 创建 Web 服务。

- 从 WSDL 数据对象创建 Web 服务。导入 WSDL 文件以创建 WSDL 数据对象。WSDL 文件为 Web 服务定义了操作输入、操作输出和操作故障。
- 手动创建 Web 服务。配置操作输入、操作输出和操作故障。可以使用架构对象中的元素和类型来定义操作组件。可以使用可重用 Mapplet、可重用转换和可重用逻辑数据对象来为操作定义操作输入和操作输出的元素。

2. 配置操作映射。
配置数据集成服务如何在 SOAP 消息与输入转换和输出转换端口之间提取数据。此外，配置操作映射逻辑并测试每个操作映射。
3. 将 Web 服务部署到数据集成服务。
将 Web 服务添加到应用程序中，并将该应用程序部署到数据集成服务。部署包含已在数据集成服务上运行的 Web 服务的应用程序时，数据集成服务会将一个数字附加到 Web 服务的服务名称中。
4. 完成 Web 服务的管理任务。
在 Administrator 工具中配置 Web 服务属性和安全。

SOAP Web 服务客户端可以连接到正在数据集成服务上运行的 SOAP Web 服务。Web 服务客户端使用 WSDL 的内容连接到 Web 服务。可以将 Web 服务使用者转换配置为使用 Web 服务连接对象连接到 Web 服务。

SOAP Web 服务示例

可以创建用于访问客户数据的 Web 服务，也可以创建用于验证客户地址数据的 Web 服务。

访问客户数据

Hypostores 客户服务代表希望通过网络从 Los Angeles 和 Boston 办事处访问客户数据。该客户服务代表想要根据客户名或客户 ID 查看客户详细信息。公司策略要求通过网络访问的数据必须安全。

开发人员和管理员完成以下步骤以访问客户服务所需的数据：

1. 在 Developer tool 中，开发人员通过以下操作创建 Web 服务：
 - getCustomerDetailsByName
操作输入包含客户名称的元素。操作输出包含基于客户名称的客户详细信息的元素。
 - getCustomerDetailsById
操作输入包含客户 ID 的元素。操作输出包含基于客户 ID 的客户详细信息的元素。
2. 开发人员使用以下组件配置每个操作的操作映射：
 - 输入转换和输出转换。
 - 从 Los Angeles 和 Boston 办事处对定义客户数据的单个视图的逻辑数据对象执行查找的查找转换。
3. 开发人员向数据集成服务部署 Web 服务。
4. 在 Administrator 工具中，管理员将 Web 服务配置为使用传输层安全和消息层安全，以便能够使用 HTTPS URL 接收授权请求。
5. 管理员向客户服务发送 WSDL URL 以使他们能够连接到 Web 服务。

验证客户地址数据

Hypostores 执行部门希望在最终确定订单之前对地址数据进行验证。地址验证器转换将输入地址数据与地址引用数据相比较，以确定输入地址的准确性并修复这些地址中的错误。

开发人员和管理员完成以下步骤，以对执行部门提供地址验证功能：

1. 在 Developer tool 中，开发人员使用地址验证转换创建一个 mapplet，可接收地址数据作为输入，并将验证后的地址作为输出返回。
2. 开发人员创建一个 Web 服务，并使用 mapplet 创建 Web 服务操作。
可以使用**创建 Web 服务**向导通过可重用对象创建操作。
3. 开发人员向数据集成服务部署 Web 服务。
4. 管理员向执行部门发送 WSDL URL，以使他们能够连接到 Web 服务。

Web 服务接受地址作为输入，并返回验证后的地址作为输出。

第 3 章

WSDL 数据对象

本章包括以下主题：

- [WSDL 数据对象概览, 19](#)
- [WSDL 数据对象概览视图, 20](#)
- [WSDL 数据对象高级视图, 20](#)
- [导入 WSDL 数据对象, 20](#)
- [WSDL 同步, 21](#)
- [证书管理, 22](#)

WSDL 数据对象概览

WSDL 数据对象是使用 WSDL 文件作为源的物理数据对象。可以使用 WSDL 数据对象创建 Web 服务或 Web 服务使用者转换。导入 WSDL 文件以创建 WSDL 数据对象。

导入 WSDL 数据对象后，可以在**概览**和**高级**视图中编辑常规属性和高级属性。**WSDL** 视图将显示 WSDL 文件内容。

导入 WSDL 时，请考虑以下准则：

- WSDL 文件必须与 WSDL 1.1 兼容。
- WSDL 文件必须有效。
- 要在 Web 服务或 Web 服务使用者转换中包含的操作必须使用文档/文字编码。如果 WSDL 文件中的所有操作都使用除文档/文字之外的编码类型，则 WSDL 导入将失败。
- Developer 工具必须能够访问 WSDL 文件所引用的任何架构。
- 如果 WSDL 文件包含架构或者具有外部架构，Developer 工具将在 WSDL 数据对象内创建嵌入式架构。
- 如果一个 WSDL 文件导入另一个 WSDL 文件，Developer 工具会将这两个 WSDL 组合在一起创建 WSDL 数据对象。
- 如果一个 WSDL 文件定义多个操作，Developer 工具将在 WSDL 数据对象中包含所有操作。从 WSDL 数据对象创建 Web 服务时，可以选择包含一个或多个操作。

WSDL 数据对象概览视图

WSDL 数据对象的概览视图将显示有关 WSDL 以及其中的操作的常规信息。

下表介绍了可为 WSDL 数据对象配置的常规属性：

属性	说明
名称	WSDL 数据对象的名称。
说明	WSDL 数据对象的说明。

下表介绍了在 WSDL 数据对象中定义的操作的列：

属性	说明
操作	WSDL 在其中定义操作的消息格式和协议的位置。
输入	与输入操作相关联的 WSDL 消息名称。
输出	与输出操作相关联的 WSDL 消息名称。
故障	与操作故障相关联的 WSDL 消息名称。

WSDL 数据对象高级视图

WSDL 数据对象的高级视图将显示 WSDL 数据对象的高级属性。

下表介绍了 WSDL 数据对象的高级属性：

属性	说明
连接	Web 服务使用者转换的默认 Web 服务连接。
文件位置	WSDL 文件所在的位置。

导入 WSDL 数据对象

可以从 WSDL 文件或指向 WSDL 位置的 URI 来导入 WSDL 数据对象。可以从包含 SOAP 1.1 或 SOAP 1.2 绑定操作之一或同时包含这两者的 WSDL 文件导入 WSDL 数据对象。

1. 单击**文件 > 新建 > 数据对象**。
2. 选择**WSDL 数据对象**，然后单击**下一步**。
此时将显示**新建 WSDL 数据对象**对话框。
3. 单击**WSDL**选项旁的**浏览**，然后输入 WSDL 的位置。然后单击**确定**。

输入 WSDL 的位置时，可以浏览到 WSDL 文件，也可以输入指向 WSDL 的 URI。

注意: 如果 URI 包含非英文字符，则导入可能会失败。将 URI 复制到任意浏览器的地址栏中。将该位置从浏览器复制回来。Developer 工具将从浏览器接受编码后的 URI。

4. 输入 WSDL 的名称。
5. 单击位置选项旁的**浏览**，以选择要在其中导入 WSDL 数据对象的项目或文件夹位置。
6. 单击**下一步**以查看 WSDL 中的操作。
7. 单击**完成**。

数据对象显示在**对象浏览器**视图的项目或文件夹中的**物理数据对象**下。

WSDL 同步

可以在 WSDL 文件更改时同步 WSDL 数据对象。同步 WSDL 数据对象时，Developer 工具将从 WSDL 文件重新导入对象元数据。

可以使用 WSDL 数据对象创建 Web 服务或 Web 服务使用者转换。更新 WSDL 数据对象时，Developer 工具将更新引用 WSDL 的对象，并在打开这些对象时将这些对象标记为已更改。Developer 工具将新的 WSDL 与旧的 WSDL 进行比较时，会通过名称属性标识 WSDL 组件。

如果没有名称属性发生更改，Developer 工具将更新引用 WSDL 组件的对象。例如，可以编辑 WSDL 文件并将简单元素“CustID”的类型从 xs:string 改为 xs:integer。

如果名称属性发生更改，Developer 工具将在您打开引用 WSDL 组件的对象时将这些对象标记为已更改。

Developer 工具将先验证 WSDL 文件，然后再更新 WSDL 数据对象。如果 WSDL 文件包含错误，Developer 工具不会导入文件。

同步 WSDL 数据对象

在 WSDL 文件更改时同步 WSDL 数据对象。

1. 在**对象浏览器**视图中右键单击 WSDL 数据对象，然后选择**同步**。

此时将显示**同步 WSDL 数据对象**对话框。

2. 单击 **WSDL** 字段旁的**浏览**，然后输入 WSDL 的位置。然后单击**确定**。

输入 WSDL 的位置时，可以浏览到 WSDL 文件，也可以输入指向 WSDL 的 URI。

注意: 如果 URI 包含非英文字符，则导入可能会失败。将 URI 复制到任意浏览器的地址栏中。将该位置从浏览器复制回来。Developer 工具将从浏览器接受编码后的 URI。

3. 验证 WSDL 名称和位置。
4. 单击**下一步**以查看 WSDL 中的操作。
5. 单击**完成**。

Developer 工具将更新引用 WSDL 的对象，并在您打开这些对象时将它们标记为已更改。

证书管理

Developer 工具必须使用证书从需要客户端身份验证的 URL 导入 WSDL 数据对象和架构对象。

默认情况下，当托管 URL 的服务器使用受信任证书时，Developer 工具将从需要客户端身份验证的 URL 导入对象。如果托管 URL 的服务器使用非信任证书，请将此非信任证书添加到 Developer 工具中。如果不将非信任证书添加到 Developer 工具中，Developer 工具无法导入对象。向服务器管理员请求要从中导入对象的 URL 的证书文件和密码。

添加到 Developer 工具中的证书适用于您在 Developer 工具计算机上执行的导入。Developer 工具不在模型存储库中存储证书。

Informatica Developer 证书属性

如果要从需要使用非信任证书进行客户端身份验证的 URL 导入对象，请将证书添加到 Developer 工具中。

下表介绍了证书属性：

属性	说明
主机名	托管 URL 的服务器的名称。
端口号	URL 的端口号。
证书文件路径	客户端证书文件的位置。
密码	客户端证书文件的密码。

将证书添加到 Informatica Developer

添加证书时，可以配置当从需要使用非信任证书进行客户端身份验证的 URL 导入对象时 Developer 工具所使用的证书属性。

1. 单击**窗口 > 首选项**。
2. 选择 **Informatica > Web 服务 > 证书**。
3. 单击**添加**。
4. 配置证书属性。
5. 单击**确定**。

第 4 章

架构对象

本章包括以下主题：

- [架构对象概览, 23](#)
- [架构对象概览视图, 23](#)
- [架构对象架构视图, 24](#)
- [架构对象高级视图, 28](#)
- [创建架构对象, 29](#)
- [架构更新, 29](#)
- [证书管理, 32](#)

架构对象概览

架构对象是您导入模型存储库的层次结构架构。导入架构后，您可以在 Developer tool 中查看该架构的组件。您可以导入 Avro、Parquet、XML 或 JSON 架构。Developer tool 会在模型存储库中将架构转换为 .xsd 文件。

在创建 SOAP Web 服务时，您可以根据层次结构架构定义 Web 服务的结构。创建不包含 WSDL 的 Web 服务后，可以根据架构定义的类型和元素来定义操作、输入、输出和故障特征。

导入架构后，可以在**概览**视图中编辑常规架构属性。在**高级**视图中编辑高级属性。在**架构**视图中查看架构文件内容。

架构对象概览视图

选择**概览**视图可更新架构名称或架构说明、查看命名空间以及管理架构文件。

概览视图显示架构的名称、说明和目标命名空间。您可以编辑架构名称和说明。目标命名空间显示架构组件所属的命名空间。如果未显示任何目标命名空间，则架构组件不属于命名空间。

架构位置区域列出了架构文件和命名空间。可以添加多个根 .xsd 文件。如果架构文件包含或导入其他架构文件，Developer 工具会在架构中包含子 .xsd 文件。

架构文件

可以向架构对象添加多个根级 .xsd 文件。也可以从架构对象中删除根级 .xsd 文件。

添加架构文件后，Developer 工具会导入由您添加的文件导入的、或由该文件包含的所有 .xsd 文件。Developer 工具会根据属于架构对象的文件验证您添加的文件。如果文件与属于架构对象的文件冲突，Developer 工具不会允许您添加该文件。

例如，架构对象包含根架构文件“BostonCust.xsd”。您想要向该架构对象添加根架构文件“LACust.xsd”。这两个架构文件都包含相同的目标命名空间，并且定义名为“Customer”的元素。您尝试向架构对象添加架构文件 LACust.xsd 时，Developer 工具会提示您是保留 BostonCust.xsd 文件还是用 LACust.xsd 文件将其覆盖。

可以删除任何根级架构文件。如果您删除架构文件，Developer 工具会将架构文件定义的元素类型更改为 xs:string。

要添加架构文件，请选择**概览**视图，单击**架构位置**列表旁边的**添加**按钮。然后选择架构文件。要删除架构文件，请选择该文件并单击**删除**按钮。

架构对象架构视图

架构视图显示架构中的组、元素、类型、特性组和特性的字母顺序列表。在**架构**视图中选择组、元素、类型、特性组或特性后，属性会显示在右侧面板中。还可以在**架构**视图中查看每个 .xsd 文件。

架构视图提供架构对象中的命名空间和 .xsd 文件列表。

在**架构**视图中可执行以下操作：

- 要查看架构构造列表，请展开**指令**文件夹。要查看命名空间、前缀和位置，请从列表中选择架构构造。
- 要查看命名空间前缀、生成的前缀和位置，请选择命名空间。可以更改生成的前缀。
- 要查看架构对象的 .xsd 文件，请选择**源**。如果架构对象包含其他架构，您可以选择要查看的 .xsd 文件。
- 要查看架构的每个命名空间中的组、元素、类型、特性组和特性的字母顺序列表，请选择**设计**。可以在**名称**字段中输入一个或多个字符，以按名称筛选组、元素、类型、特性组和特性。
- 要查看元素属性，请选择组、元素、类型、特性组或特性。Developer 工具会根据您选择的对象在右侧面板中显示不同的字段。

查看类型时，可以看到某种类型是否派生自另一种类型。该界面会显示父类型。该界面还会显示子元素是否按限制或扩展继承值。

命名空间属性

命名空间视图显示选定命名空间的前缀和位置。

与每个架构文件关联的命名空间会区分来源不同但名称相同的元素。统一资源标识符 (URI) 引用定义包含元素和属性名称的文件的位置。

当您导入一个包含多个命名空间的架构时，Developer tool 会将这些命名空间添加到架构对象中。架构文件包含其他架构时，这些架构的命名空间也会一并添加。

Developer 工具会为每个命名空间创建生成的前缀。当架构不包含前缀时，Developer tool 会生成命名空间前缀 tns0，并为每个增加的命名空间前缀递增前缀号。Developer 工具会保留命名空间前缀 xs。如果您导入的架构包含命名空间前缀 xs，Developer tool 将创建生成的前缀 xs1。架构包含生成的前缀值时，Developer 工具会增加前缀号。

例如，Customer_Orders.xsd 包含命名空间。该架构包含另一个架构 Customers.xsd。Customers 架构包含不同的命名空间。Developer 工具会为 Customer_Orders 命名空间分配前缀 tns0，为 Customers 命名空间分配前缀 tns1。

要查看命名空间位置和前缀，请在**架构**视图中选择命名空间。

从多个架构对象创建 Web 服务时，每个命名空间都必须具有唯一前缀。您可以为每个命名空间修改生成的前缀。

元素属性

元素为简单类型或复杂类型。复杂类型包含其他类型。在**架构**视图选择元素后，Developer 工具会在屏幕的右侧面板中列出子元素和属性。

下表介绍了选择元素时将显示的元素属性：

属性	说明
名称	元素名称。
说明	类型说明。
类型	元素类型。

下表介绍了选择元素时将显示的子元素属性：

属性	说明
名称	元素名称。
类型	元素类型。
最少出现次数	元素在某个时刻在一个实例中可以出现的最少次数。
最多出现次数	元素在某个时刻在一个实例中可以出现的最多次数。
说明	元素说明。

要查看其他子元素属性，请单击“说明”列中的双箭头以展开该窗口。

下表介绍了展开“说明”列时将显示的其他子元素属性：

属性	说明
固定值	不会更改的特定元素值。
可空	元素可以包含空值。空元素包含元素标记，但不包含任何值和内容。
抽象	元素为抽象类型。实例必须包含该类型派生的类型。如果没有派生的元素类型，则抽象类型不是有效类型。
最小值	实例中元素的最小值。
最大值	实例中元素的最大值。

属性	说明
最小长度	元素的最小长度。 长度取决于元素类型，以字节、字符或项目为单位。
最大长度	元素的最大长度。 长度取决于元素类型，以字节、字符或项目为单位。
枚举	元素的所有合法值的列表。
模式	定义有效元素值的表达式模式。

高级元素属性

要查看元素的高级属性，请在**架构视图**中选择该元素。 单击**高级**。

下表介绍了元素的高级属性：

属性	说明
抽象	元素为抽象类型。 SOAP 消息必须包含该类型派生的类型。 如果没有派生的元素类型，则抽象类型不是有效类型。
块	阻止派生的元素代替此元素显示在层次结构中。块值可以包含“#all”或扩展、限制或置换的列表。
最后	阻止架构扩展或限制简单类型作为派生类型。
置换组	要置换为该元素的元素的名称。
可空	元素可以包含空值。 空元素包含元素标记，但不包含任何值和内容。

简单类型属性

简单类型元素是指包含非结构化文本的元素。在**架构视图**中选择简单类型元素后，有关该简单类型元素的信息会显示在右侧面板中。

下表介绍了您可以查看的简单类型的属性：

属性	说明
类型	元素名称。
说明	元素说明。
种类	定义简单类型是否为联合类型、列表类型、任意类型或原子类型。 原子元素不包含任何其他元素或特性。
成员类型	联合构造中的类型列表。
项类型	元素类型。
基本	原子元素的基本类型，例如整数或字符串。
最小长度	元素的最小长度。 长度取决于元素类型，以字节、字符或项目为单位。
最大长度	元素的最大长度。 长度取决于元素类型，以字节、字符或项目为单位。

属性	说明
折叠空格	去除前导空格和尾随空格。将多个空格折叠成单个空格。
枚举	将类型限制为合法值列表。
模式	将类型限制为模式表达式定义的值。

简单类型高级属性

要查看简单类型的高级属性，请在**架构**视图中选择该简单类型。单击**高级**。

高级属性显示在简单类型属性的下方。

下表介绍了简单类型的高级属性：

属性	说明
最后	阻止架构扩展或限制简单类型作为派生类型。

复杂类型属性

复杂类型是指包含其他元素和属性的元素。复杂类型包含属于简单或复杂类型的元素。在**架构**视图中选择复杂类型后，Developer 工具会在屏幕的右侧面板中列出子元素和子元素属性。

下表介绍了复杂类型属性：

属性	说明
名称	类型名称。
说明	类型说明。
继承自	父类型的名称。
继承者	限制或扩展。复杂类型派生自父类型。复杂类型可能减少父级的元素或特性。或者，它也可能添加元素和特性。

要查看属于复杂类型的每个元素的属性，请单击“说明”列中的双箭头以展开该窗口。

复杂类型高级属性

要查看复杂类型的高级属性，请在**架构**视图选择该元素。单击**高级**。

下表介绍了复杂元素或类型的高级属性：

属性	说明
抽象	元素为抽象类型。SOAP 消息必须包含该类型派生的类型。如果没有派生的元素类型，则抽象类型不是有效类型。
块	阻止派生的元素代替此元素显示在架构中。块值可以包含“#all”或扩展、限制或置换的列表。

属性	说明
最后	阻止架构扩展或限制简单类型作为派生类型。
置换组	要置换为该元素的元素的名称。
可空	元素可以包含空值。空元素包含元素标记，但不包含任何值和内容。

特性属性

特性为简单类型。元素和复杂类型包含特性。全局特性显示为架构的一部分。在**架构**视图中选择全局特性后，Developer 工具会在屏幕的右侧面板中列出特性属性和相关类型属性。

下表介绍了特性属性：

属性	说明
名称	特性名称。
说明	特性说明。
类型	特性类型。
值	特性类型的值。指示特性类型的值是固定值还是默认值。如果未定义任何值，该属性会显示 default=0。

下表介绍了类型属性：

属性	说明
最小长度	类型的最小长度。长度取决于类型，以字节、字符或项目为单位。
最大长度	类型的最大长度。长度取决于类型，以字节、字符或项目为单位。
折叠空格	去除前导空格和尾随空格。将多个空格折叠成单个空格。
枚举	将类型限制为合法值列表。
模式	将类型限制为模式表达式定义的值。

架构对象高级视图

查看架构对象的高级属性。

下表介绍了架构对象的高级属性：

名称	值	说明
elementFormDefault	限定或非限定	确定元素是否必须包含命名空间。架构通过前缀或目标命名空间声明来限定元素。非限定值表示元素不需要命名空间。

名称	值	说明
attributeFormDefault	限定或非限定	确定本地声明的特性是否必须包含命名空间。架构通过前缀或目标命名空间声明来限定特性。非限定值表示特性不需要命名空间。
文件位置	到 .xsd 文件的完整路径	导入 .xsd 文件时该文件的位置。

创建架构对象

您可以导入层次结构架构文件或示例文件，用来在存储库中创建架构对象。

1. 在**对象浏览器**视图选择一个项目或文件夹。
2. 单击**文件 > 新建 > 架构**。
此时将显示**新建架构**对话框。
3. 要导入架构文件，选择**从架构创建**，然后浏览到并选择一个层次结构架构文件。
您可以输入 URI 或文件系统上的某个位置来浏览文件。Developer tool 会验证您选择的架构。查看验证消息。您可以选择 Avro、Parquet、JSON 或 .xsd 架构文件。
注意: 如果 URI 包含非英文字符，则导入可能会失败。将 URI 复制到任意浏览器的地址栏中。将该位置从浏览器复制回来。Developer tool 将从浏览器接受编码后的 URI。
4. 要从示例文件创建架构，选择**从示例文件创建**，然后浏览到并选择一个层次结构文件。
您可以选择 Avro、Parquet、JSON 或 XML 文件。
注意: 如果选择的文件具有不同的扩展名，但是包含 Avro、Parquet、JSON 或 XML 内容，向导会识别文件内容。
5. （可选）更改架构名称。
6. 单击**下一步**查看架构中的元素和类型的列表。
7. 单击**完成**导入架构。
架构显示在**对象浏览器**视图中的“架构对象”之下。Developer tool 将架构存储为 .xsd 文件。
8. 要更改为架构命名空间生成的前缀，请在**对象浏览器**视图选择该命名空间。在**命名空间**视图中更改**生成的前缀**属性。

架构更新

元素、特性、类型或其他架构组件发生更改时，您可以更新架构对象。更新架构对象时，Developer 工具会更新使用该架构的对象。

您可以通过以下方法更新架构对象：

同步架构。

在 Developer 工具之外更新架构文件时，请同步架构对象。同步架构对象时，Developer 工具会重新导入包含这些更改的所有架构 .xsd 文件。

编辑架构文件。

要从 Developer 工具中更新文件时，请编辑架构文件。编辑架构文件时，Developer 工具会在您为 .xsd 文件使用的编辑器中打开该文件。可以在其他编辑器中打开该文件，也可以在 Developer 工具中为 .xsd 文件设置默认编辑器。

可以使用架构在 Web 服务中定义元素类型。更新包含在 Web 服务的 WSDL 中的架构时，Developer 工具会更新该 Web 服务并在您打开该 Web 服务时将其标记为已更改。Developer 工具比较新架构与旧架构时，会通过名称特性标识架构组件。

如果名称特性未发生更改，Developer 工具会根据架构更改来更新 Web 服务。例如，您从 Developer 工具中编辑架构文件，并将元素“Item”的 maxOccurs 特性从 10 更改为“无限大”。保存该文件时，Developer 工具会在引用 Item 元素的任何 Web 服务中更新 maxOccurs 特性。

如果名称特性发生更改，Developer 工具会在您打开该 Web 服务时将其标记为已更改。例如，您在 Developer 工具之外编辑架构，并将复杂元素类型名称从“Order”更改为“CustOrder”。然后，您同步该架构。打开引用该元素的 Web 服务时，Developer 工具将在编辑器中用星号标记 Web 服务名称，以表明该 Web 服务包含更改。Developer 工具会将 CustOrder 元素类型添加到 Web 服务，但它不会删除 Order 元素类型。因为 Developer 工具不再能确定 Order 元素的类型，所以它会将该元素类型更改为 xs:string。

架构同步

架构组件发生更改时，您可以同步架构对象。同步架构对象时，Developer 工具会从架构文件重新导入对象元数据。

在 Developer 工具之外对架构对象进行复杂更改时，请使用架构同步。例如，您可以在执行以下操作后同步架构：

- 更改多个架构文件。
- 向架构添加架构文件或从架构中删除架构文件。
- 更改导入或包含元素。

Developer 工具会先验证架构文件，然后再更新架构对象。如果架构文件包含错误，则 Developer 工具不会导入这些文件。

要同步架构对象，请在**对象浏览器**视图中右键单击该架构对象，然后选择**同步**。

架构文件编辑

您可以从 Developer 工具中编辑架构文件以更新架构组件。

在 Developer 工具中编辑架构文件以对少量文件进行微小更新。例如，您可以对架构文件进行以下微小更新之一：

- 更改元素的 minOccurs 或 maxOccurs 特性。
- 向复杂类型添加特性。
- 更改简单对象类型。

编辑架构文件时，Developer 工具会在编辑器中打开架构文件的临时副本。可以通过您为 .xsd 文件使用的系统编辑器来编辑架构文件，也可以选择其他编辑器。还可以为 .xsd 文件设置 Developer 工具默认编辑器。编辑临时架构文件后进行保存。

Developer 工具会先验证该临时文件，然后再更新架构对象。如果架构文件包含错误或包含与架构对象中的其他架构文件冲突的组件，Developer 工具不会导入该文件。

注意：编辑并保存该临时架构文件后，Developer 工具不会更新**架构位置**列表中显示的架构文件。如果在 Developer 工具中编辑完架构文件后再同步架构对象，同步操作会覆盖您所做的编辑。

设置默认架构文件编辑器

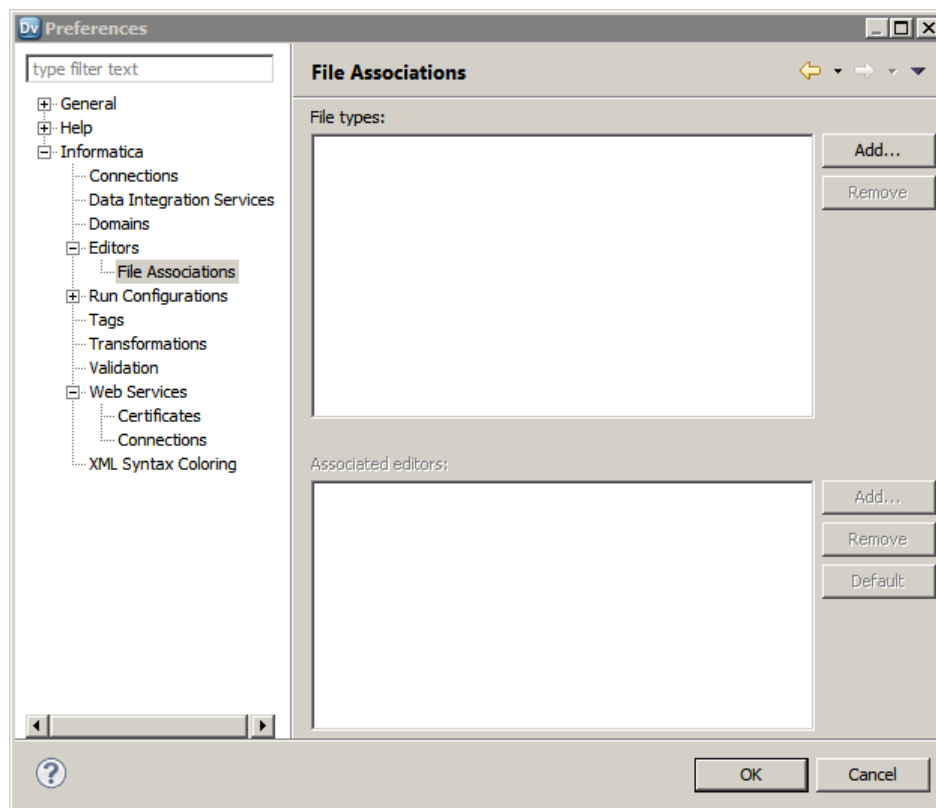
您可以设置 Developer 工具在您编辑架构文件时打开的默认编辑器。

1. 单击**窗口 > 首选项**。

此时将显示**首选项**对话框。

2. 单击**编辑器 > 文件关联**。

此时将显示**首选项**对话框的**文件关联**页。



3. 单击**文件类型**区域旁边的**添加**。

此时将显示**添加文件类型**对话框。

4. 输入 .xsd 作为文件类型，然后单击**确定**。

5. 单击**关联编辑器**区域旁边的**添加**。

此时将显示**选择编辑器**对话框。

6. 从编辑器列表中选择编辑器，或单击**浏览**选择其他编辑器，然后单击**确定**。

选择的编辑器会显示在**关联编辑器**列表中。

7. （可选）向**关联编辑器**列表添加其他编辑器。

8. 如果添加多个编辑器，可以更改默认编辑器。选择某个编辑器，然后单击**默认**。

9. 单击**确定**。

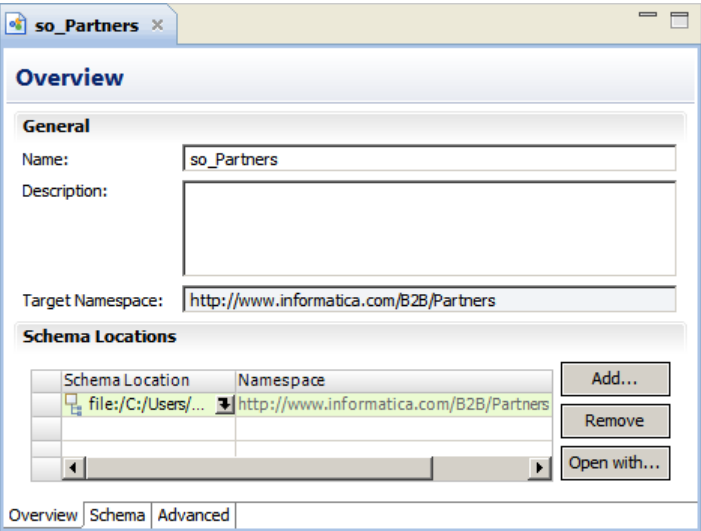
编辑架构文件

可以编辑架构对象中的任何架构文件。

1. 打开架构对象。

2. 选择**概览**视图。

此时将显示架构对象的**概览**视图。



3. 在**架构位置**列表中选择某个架构文件。

4. 单击**打开方式**，然后选择以下选项之一：

选项	描述
系统编辑器	该架构文件会在您的操作系统为 .xsd 文件使用的编辑器中打开。
默认编辑器	该架构文件会在您在 Developer 工具中设置为默认编辑器的编辑器中打开。此选项显示您是否已设置默认编辑器。
其他	选择要在其中打开架构文件的编辑器。

Developer 工具会打开架构文件的临时副本。

5. 更新临时架构文件，保存更改，然后关闭编辑器。

Developer 工具会提示您更新架构对象。

6. 要更新架构对象，请单击**更新架构对象**。

Developer 工具会根据您所做的更改更新架构文件。

证书管理

Developer 工具必须使用证书从需要客户端身份验证的 URL 导入 WSDL 数据对象和架构对象。

默认情况下，当托管 URL 的服务器使用受信任证书时，Developer 工具将从需要客户端身份验证的 URL 导入对象。如果托管 URL 的服务器使用非信任证书，请将此非信任证书添加到 Developer 工具中。如果不将非信任证书添加到 Developer 工具中，Developer 工具无法导入对象。向服务器管理员请求要从中导入对象的 URL 的证书文件和密码。

添加到 Developer 工具中的证书适用于您在 Developer 工具计算机上执行的导入。Developer 工具不在模型存储库中存储证书。

Informatica Developer 证书属性

如果要从需要使用非信任证书进行客户端身份验证的 URL 导入对象，请将证书添加到 Developer 工具中。

下表介绍了证书属性：

属性	说明
主机名	托管 URL 的服务器的名称。
端口号	URL 的端口号。
证书文件路径	客户端证书文件的位置。
密码	客户端证书文件的密码。

将证书添加到 Informatica Developer

添加证书时，可以配置当从需要使用非信任证书进行客户端身份验证的 URL 导入对象时 Developer 工具所使用的证书属性。

1. 单击**窗口 > 首选项**。
2. 选择 **Informatica > Web 服务 > 证书**。
3. 单击**添加**。
4. 配置证书属性。
5. 单击**确定**。

第 5 章

如何创建 SOAP Web 服务

本章包括以下主题：

- [创建 SOAP Web 服务概览, 34](#)
- [类型和元素, 34](#)
- [Web 服务概览视图, 35](#)
- [Web 服务 WSDL 视图, 37](#)
- [从 WSDL 数据对象创建 Web 服务, 37](#)
- [手动创建 SOAP Web 服务, 39](#)

创建 SOAP Web 服务概览

创建 SOAP Web 服务时，将在存储库中创建一个 Web 服务对象。可以从 WSDL 数据对象创建 SOAP Web 服务，也可以手动创建 SOAP Web 服务。一个 SOAP Web 服务可以具有一个或多个操作。

从 WSDL 数据对象创建 SOAP Web 服务时，请从 WSDL 数据对象中选择要包含在 Web 服务中的操作。每个操作都可以使用 SOAP 1.1 绑定或 SOAP 1.2 绑定，但不能同时使用这两种绑定。

手动创建 Web 服务时，请创建操作并定义每个操作的操作输入、操作输出和操作故障的元素。可以使用可重用对象来定义每个操作的操作输入和操作输出的元素。

Web 服务对象具有**概览**视图和 **WSDL** 视图。可以在**概览**视图中创建和配置操作。可以在 **WSDL** 视图中查看 WSDL 文件内容。

创建 Web 服务后，请为每个操作配置操作映射。可以有选择地向 Web 服务中添加操作。

类型和元素

从 WSDL 数据对象创建 Web 服务时，WSDL 数据对象将为每个操作定义元素和元素类型。手动创建 Web 服务时，可以定义元素或使用架构对象中的类型定义元素。

如果操作组件中包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，则配置操作映射时必须选择一个或多个类型、元素或属性。例如，如果操作输入包含 anyType 元素，则配置输入转换时应选择一个或多个类型。

使用架构对象定义元素类型时，Developer 工具会在 Web 服务的 WSDL 中包含该架构对象。在 Web 服务 WSDL 中删除该架构对象或删除该架构对象的链接时，Developer 工具会将该架构对象定义的元素元素类型改为 xs:string。可以将元素类型更新为其他类型。

手动创建 Web 服务时，Web 服务会为其用来定义操作组件的每个命名空间请求一个唯一的生成前缀值。例如，如果 Web 服务使用架构对象 schemaA 和架构对象 schemaB 定义类型，则 schemaA 和 schema B 不能对任何命名空间使用相同的生成前缀值。

Web 服务概览视图

Web 服务概览视图显示有关 Web 服务的常规信息以及有关 Web 服务操作和操作组件的详细信息。

常规属性

下表描述为 Web 服务配置的常规属性：

属性	说明
名称	Web 服务对象名称。
说明	Web 服务的说明。
命名空间	Web 服务的 targetNamespace。如果 Web 服务与 WSDL 数据对象关联，则该字段为只读。
前缀	targetNamespace 的前缀。如果 Web 服务与 WSDL 数据对象关联，则该字段为只读。
WSDL 数据对象	与 Web 服务关联的 WSDL 数据对象。如果 Web 服务是从 WSDL 数据对象创建，则显示该属性。
服务名称	服务名称。默认为 Web 服务的名称或在关联的 WSDL 数据对象中定义的服务名称。如果 Web 服务与 WSDL 数据对象关联，则该字段为只读。

操作区域

在“操作”区域的左侧选择某个操作时，详细信息将显示在右侧。“操作”区域的左侧显示包含相关输入、输出和故障的操作的树状视图。

下表描述了操作以及显示在“操作”区域右侧的输入和输出属性。

属性	说明
操作名称	操作的名称。
说明	操作的说明
绑定类型	操作的绑定类型。
输入	与操作输入相关联的 WSDL 消息名称。
输出	与操作输出相关联的 WSDL 消息名称。
元素名称	操作输入或输出所引用的 XSD 元素的名称。
名称	操作输入或输出所引用的元素的子元素。
类型	元素类型。

属性	说明
最少出现次数	元素在 XML 实例中的某一点可以出现的最少次数。
最多出现次数	元素在 XML 实例中的某一点可以出现的最多次数。
说明	元素的说明。

要查看其他子元素属性，请单击“说明”列中的双箭头以展开该窗口。

下表描述了展开“说明”列时将显示的其他子元素属性：

属性	说明
固定值	不会更改的特定元素值。
可空	元素可以包含空值。空元素包含元素标记，但不包含任何值和内容。
抽象	元素为抽象类型。XML 实例必须包含该类型派生的类型。如果没有派生的元素类型，则抽象类型不是有效类型。
最小值	XML 实例中的元素的最小值。
最大值	XML 实例中的元素的最大值。
最小长度	元素的最小长度。长度取决于元素类型，以字节、字符或项目为单位。
最大长度	元素的最大长度。长度取决于元素类型，以字节、字符或项目为单位。
枚举	元素的所有合法值的列表。
模式	定义有效元素值的表达式模式。

操作映射区域

下表描述了操作映射的列：

属性	说明
操作映射	操作映射名称。
操作	操作名称。
输入	与操作输入相关联的 WSDL 消息名称。
输出	与操作输出相关联的 WSDL 消息名称。
故障	与操作故障相关联的 WSDL 消息名称。

Web 服务 WSDL 视图

可以在 Web 服务的 **WSDL** 视图上预览 WSDL 文件的内容。

从 WSDL 对象创建 Web 服务时，可以查看 WSDL 文件的内容。

手动创建 Web 服务时，Developer tool 基于操作输入、操作输出、操作故障和表头的配置生成 WSDL 文件的内容。可将元素添加到 WSDL 的架构中，或者在 WSDL 的架构中编辑元素。还可以从 WSDL 删除导入的架构对象。

每个操作输入、操作输出、操作故障和表头均与 WSDL 中的一条消息对应。操作故障和表头可以共享 WSDL 中的消息。使用 WSDL 的架构中的元素创建故障或表头时，消息名称与元素名称相同。

可以查看 WSDL 设计或 WSDL 源。WSDL 设计显示 WSDL 内容的层次结构视图。WSDL 源以 XML 格式显示 WSDL 的内容。

要在 Developer tool 中预览 WSDL，可在 Web 服务的 **WSDL** 视图上选择**显示**字段旁边的**源**。

从 WSDL 数据对象创建 Web 服务

从 WSDL 数据对象创建 SOAP Web 服务时，WSDL 将定义一个或多个操作以及每个操作输入、操作输出和操作故障的元素。

如果 WSDL 有多个操作可供选择，您将能够添加操作。无法创建未在 WSDL 中定义的操作。

从 WSDL 数据对象创建 Web 服务之前，必须创建一个 WSDL 数据对象。WSDL 必须基于 SOAP 1.1 或 SOAP 1.2。从 WSDL 数据对象创建 Web 服务之前，可以使用**从 WSDL 数据对象创建 Web 服务**向导创建 WSDL 数据对象。

从 WSDL 数据对象创建 Web 服务时，Web 服务将与该 WSDL 数据对象相关。如果删除该 WSDL 数据对象，操作映射将失效，因此必须将一个 WSDL 数据对象与该 Web 服务相关联。从存储库中删除 WSDL 数据对象时，Developer tool 将删除 Web 服务与 WSDL 数据对象之间的关联。

步骤 1. 从 WSDL 数据对象创建 Web 服务

从 WSDL 数据对象创建 Web 服务时，Developer 工具将使用 WSDL 数据对象中定义的信息为 Web 服务中的每个操作创建操作映射。

1. 在**对象浏览器**视图选择一个项目或文件夹。
2. 单击**文件 > 新建 > 数据服务**。
此时将显示**新建数据服务**对话框。
3. 单击**来自 WSDL 数据对象的 Web 服务**，然后单击**下一步**。
4. 要创建 WSDL 数据对象，请单击**新建 WSDL 数据对象**。
 - a. 单击**位置**选项旁边的**浏览**输入 WSDL 所在的位置。单击**确定**。
 - b. 输入 WSDL 的名称。
 - c. 单击**完成**。
5. 输入 Web 服务的名称。
6. 单击**WSDL 数据对象**选项旁边的**浏览**选择 WSDL 数据对象。
7. 单击**操作**选项旁边的**浏览**选择一项或多项您要包含在 Web 服务中的操作。

WSDL 数据对象可以包含多个 SOAP 1.1 和 SOAP 1.2 格式绑定。您只能从一个这类绑定中选择特定操作。

8. 单击**下一步**。
9. 默认情况下，所选操作的 SOAP 版本将显示为**操作类型**。
10. 或者，为每个操作输入选择**映射输入**选项卡将数据从操作输入映射至输出端口。
您也可以在配置输入转换时映射数据。
11. 或者，为每个操作输出选择**映射输出**选项卡将数据从输入端口映射至操作输出。
您也可以在配置输出转换时映射数据。
12. 或者，为每个操作故障选择**映射故障**选项卡将数据从输入端口映射至操作故障。
您也可以在配置故障转换时映射数据。
13. 单击**完成**。

步骤 2. 将操作添加到 Web 服务

您可以选择将操作添加到 Web 服务。

1. 在**对象浏览器**视图中，打开一项与 WSDL 数据对象相关的 Web 服务。
2. 选择**概览**视图。
3. 在**操作**部分中，选择**操作映射**区域。
4. 从**操作映射**区域中，单击**选择**。
您将会看到**选择一个或多个操作**对话框。
5. 选择要添加的操作，然后单击**确定**。

将 WSDL 数据对象与 Web 服务关联

要将 WSDL 数据对象与 Web 服务关联，请选择一个 WSDL 数据对象，然后将每个操作映射与该 WSDL 数据对象中定义的操作关联起来。

从 WSDL 数据对象创建 Web 服务时，Web 服务将与该 WSDL 数据对象相关。如果删除该 WSDL 数据对象，操作映射将失效，因此必须将一个 WSDL 数据对象与该 Web 服务关联起来。

1. 在**对象浏览器**视图中，打开一项与 WSDL 数据对象相关的 Web 服务。
2. 选择**概览**视图。
3. 单击 **WSDL 数据对象** 选项旁边的**浏览**。
此时将显示**选择 WSDL 数据对象**对话框。
4. 选择 WSDL 数据对象，然后单击**确定**。
5. 在**操作映射**区域，将 WSDL 操作与每个 Web 服务操作映射关联起来。
 - a. 在操作映射行的**操作**列中单击鼠标右键，并单击**选择操作**。
此时将显示**选择操作**对话框。
 - b. 选择该操作，然后单击**确定**。

手动创建 SOAP Web 服务

在没有 WSDL 数据对象的情况下创建 SOAP Web 服务时，可以定义 Web 服务属性、操作和操作组件。操作组件包括操作输入、操作输出和操作故障。

手动创建 Web 服务时，可以定义一个或多个操作。手动创建操作时，可以选择 SOAP 绑定类型，并使用架构对象中的元素和类型来定义操作元素。也可以使用 Mapplet、可重用转换、逻辑数据对象、平面文件数据对象或关系数据对象来创建操作。从可重用对象创建操作时，可从该对象中选择要包括在该操作中的字段。

Developer tool 可使用您定义的 Web 服务属性和操作生成 WSDL 文件的内容。在创建 Web 服务后，可以预览 WSDL 文件的内容。

步骤 1.手动创建 Web 服务

使用**创建 Web 服务**向导创建 Web 服务。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件 > 新建 > 数据服务**。
此时将显示**新建数据服务**对话框。
3. 单击 **Web 服务**。
此时将显示**新建 Web 服务**对话框。
4. 输入 Web 服务的名称。
5. 或者，输入命名空间和命名空间前缀。
6. 单击**下一步**。
此时将显示**新建 Web 服务**对话框。

创建 Web 服务之后，必须创建操作。

步骤 2.创建操作

手动创建 Web 服务时，可以从“创建 Web 服务”向导或“新建操作”向导中创建操作。

下表列出了用于创建操作的选项：

用于创建操作的选项	路径
使用“创建 Web 服务”向导。	单击 新建 > 数据服务 > Web 服务 。 从可重用对象或空对象中。 注意: 可重用对象可以是物理数据对象、Mapplet 或转换。
使用“新建操作”向导。	在 对象浏览器 或 大纲 视图中右键单击某项 Web 服务，然后单击 新建 > 操作 。
使用“新建操作”向导。	创建 Web 服务，然后选择该 Web 服务的 概览 视图。在 操作 区域中，单击 新建 按钮旁的箭头，然后选择 新建操作 。
使用“新建操作”向导。	将可重用对象拖至 概览 视图的 操作 区域。

从可重用对象创建操作

使用**创建 Web 服务**向导从可重用对象创建 Web 服务和操作。

可以创建操作，以便在逻辑数据对象、平面文件数据对象和关系数据对象中查找数据。还可以从 Mapplet 或可重用转换创建操作。Mapplet 或可重用转换定义操作输入和输出的元素。无法从 Web 服务使用者转换创建操作。

完成创建 Web 服务的步骤之后，可以使用**创建 Web 服务**向导从数据对象、Mapplet 或可重用转换创建操作。

1. 在**新建 Web 服务**对话框中，单击**新建**按钮旁边的箭头。然后选择**操作 > 从可重用对象创建**。

此时将显示**选择可重用对象**对话框。

注意: 还可以通过以下方法从数据对象创建操作：在**对象浏览器**中打开某项 Web 服务并选择**概览 > 操作区域**。单击**新建**按钮旁边的箭头。

2. 选择该对象，然后单击**确定**。

3. 或者，输入操作名称和说明。

Developer tool 将使用您为操作提供的名称来定义操作输入和操作输出的名称。

4. 选择操作的绑定类型。

默认为 SOAP 1.1。在创建并保存操作后，无法更改操作的绑定类型。

5. 单击操作输入以查看和配置输入属性。

- a. 在**操作输入**选项卡上选择操作输入字段。

注意: 如果对象有多个输入组，请选择**操作输入**选项卡上的输入组和操作输入字段。

- b. 或者，为每个元素配置最少和最多出现次数。

- c. 或者，单击**映射输入**选项卡配置如何将数据从操作输入映射至输出端口。

您也可以在配置输入转换时映射数据。

6. 单击操作输出以显示和配置输出属性。

- a. 在**操作输出**选项卡上选择操作输出字段。

注意: 如果对象有多个输出组，请选择**操作输出**选项卡上的输出组和操作输出字段。

- b. 或者，为每个元素配置最少出现次数。

- c. 或者，单击**映射输出**选项卡配置如何将数据从输入端口映射至操作输出。

您也可以在配置输出转换时映射数据。

7. 如果操作发送用户定义的故障，请单击**新建**按钮旁边的箭头，然后单击**故障**。

您可以选择为故障创建元素，也可以选择可重用元素。

8. 单击每一个操作故障以显示和配置故障属性。

您可以单击**映射故障**选项卡配置如何将数据从输入端口映射至操作故障。您也可以在配置故障转换时映射数据。

9. 重复步骤 [1](#) 至 [8](#) 以创建和配置更多操作。

10. 单击**完成**。

Developer tool 可根据每个操作的配置方式为每个操作创建操作映射。

您可以选择创建元素，也可以选择创建预定义故障。

手动创建操作

使用**创建 Web 服务**向导定义操作，并为每个操作定义操作输入、操作输出以及操作故障。

完成创建 Web 服务的步骤之后，可以从**创建 Web 服务**向导中创建和配置操作。

1. 在**新建 Web 服务**对话框中，单击**新建**按钮旁边的箭头。然后选择**操作 > 创建为空**。

- a. 输入操作的名称。

Developer 工具使用操作名称来定义操作输入和操作输出的名称。

- b. 选择操作的 SOAP 版本作为**绑定类型**。

- c. 要定义故障消息，请单击**新建**按钮旁边的箭头，然后单击**故障**。
您可以选择为故障消息创建所有元素，也可以选择可重用元素。向导会将元素添加到故障消息的详细信息元素中。
2. 单击操作输入以显示和配置输入属性。
 - a. 单击**操作输入**选项卡。
 - b. 要添加元素，请单击**新建**按钮旁边的箭头，然后单击**元素**。要添加子元素，请选择一个元素，单击**新建**按钮旁边的箭头，然后单击**子元素**。
 - c. 要为每个元素指定类型，请单击**类型**字段中的选择按钮。选择一个 XSD 类型或架构对象类型，并单击**确定**。
提示: 单击**类型**字段，并输入要选择的类型的前几个字符。名称以所输入的字符开头的 XSD 类型将显示在列表中。
 - d. 为每个元素配置最少和最多出现次数。
 - e. 或者，输入每个元素的说明。
 - f. 或者，单击**映射输入**选项卡将数据从操作输入映射到输出端口。
您也可以在配置输入转换时映射数据。
3. 单击每一个操作故障以显示和配置故障属性。
您可以单击**映射故障**选项卡配置如何将数据从输入端口映射至操作故障。您也可以在配置故障转换时映射数据。
4. 单击操作输出以显示和配置输出属性。
 - a. 单击**操作输出**选项卡。
 - b. 要添加元素，请单击**新建**按钮旁边的箭头，然后单击**元素**。要添加子元素，请选择一个元素，单击**新建**按钮旁边的箭头，然后单击**子元素**。
 - c. 要为每个元素指定类型，请单击**类型**字段中的选择按钮。然后选择一个 XSD 类型或架构对象类型，并单击**确定**。
 - d. 为每个元素配置最少和最多出现次数。
 - e. 或者，输入每个元素的说明。
 - f. 或者，单击**映射输出**选项卡将数据从输入端口映射到操作输出。
您也可以在配置输出转换时映射数据。
5. 重复步骤 [1](#) 至 [4](#) 以添加和配置更多操作。
6. 单击**完成**。
Developer 工具可根据每个操作的配置方式为每个操作创建操作映射。
您可以选择创建元素，也可以选择创建预定义故障。

从“新建操作”向导中创建操作

在手动创建 Web 服务之后，使用“新建操作”向导创建操作。

1. 选择一种创建操作的方法。
 - 在“对象浏览器”中，右键单击某项 Web 服务，然后选择**新建 > 操作**。
 - 在“大纲”视图中，右键单击某项 Web 服务，然后选择**新建 > 操作**。
 - 在**对象浏览器**中打开某项 Web 服务，然后选择**概览 > 操作区域**。单击**新建**按钮旁的箭头。
此时将显示**新建操作**对话框。
2. 输入操作的名称，然后输入说明（可选）。

Developer tool 将使用操作名称来定义操作输入和操作输出的名称。

3. 单击**下一步**。

4. 选择操作的绑定类型。

默认为 SOAP 1.1。在创建并保存操作后，无法更改操作的绑定类型。

5. 或者，为每个操作输入选择**映射输入**选项卡将数据从操作输入映射至输出端口。

您也可以在配置输入转换时映射数据。

6. 或者，为每个操作输出选择**映射输出**选项卡将数据从输入端口映射至操作输出。

您也可以在配置输出转换时映射数据。

7. 或者，单击**新建**按钮旁的箭头，以便为操作创建故障。

您可以选择为故障消息创建所有元素，也可以选择可重用元素。向导会将元素添加到故障消息的详细信息元素中。您也可以在配置故障转换时映射数据。

8. 单击**完成**。

您可以选择创建元素，也可以选择创建预定义故障。

拖动可重用对象

将可重用对象拖至**概览**视图的**操作**区域，以从可重用对象创建操作。

1. 从**对象浏览器**视图中打开某项 Web 服务。

2. 从**概览**视图中，选择**操作**区域。

3. 将可重用对象从**对象浏览器**视图拖至**概览**视图中的**操作**区域。

此时将显示**新建操作**对话框。

4. 选择操作的绑定类型。

默认为 SOAP 1.1。在创建并保存操作后，无法更改操作的绑定类型。

5. 或者，为每个操作输入选择**映射输入**选项卡将数据从操作输入映射至输出端口。

您也可以在配置输入转换时映射数据。

6. 或者，为每个操作输出选择**映射输出**选项卡将数据从输入端口映射至操作输出。

您也可以在配置输出转换时映射数据。

7. 或者，单击**新建**按钮旁的箭头，以便为操作创建故障。

您也可以在配置故障转换时映射数据。

8. 单击**完成**。

您可以选择创建元素，也可以选择创建预定义故障。

步骤 3。创建元素

可以选择在 Web 服务的 **WSDL** 视图中创建元素。可以使用您在 Web 服务的 **WSDL** 视图中创建的元素创建操作故障或表头。一个元素可以包含多个元素和子元素。

创建元素之前，必须从**创建 Web 服务**向导创建 Web 服务并定义一个或多个操作。

1. 打开该 Web 服务。

2. 选择 **WSDL** 视图。

3. 此时将显示 WSDL 的**设计**窗口。然后，在“架构”部分选择一个元素或元素上面的命名空间条目。

4. 单击**新建元素**按钮。

5. 要在元素中添加元素，请单击**新建**按钮旁边的箭头，然后单击**元素**。

6. 要在元素中添加子元素，请选择一个元素，单击**新建**按钮旁边的箭头，然后单击**子元素**。
 7. 或者，为每个元素配置类型。
 - a. 单击**类型**字段中的**选择**按钮。
 - b. 选择一个 XSD 类型或架构对象类型，并单击**确定**。
 8. 或者，为每个元素配置最少和最多数量或出现次数。
 9. 或者，输入每个元素的说明。
- 可以选择创建预定义的故障。

更改元素的层次结构级别

创建元素后，可以更改层次结构级别。手动创建操作时，可以右键单击**操作输入**选项卡和**操作输出**选项卡，以便执行诸如删除、移动或复制元素或子元素等操作。

还可以通过拖动元素或子元素来更改元素的层次结构。

步骤 4.创建预定义故障

可以选择创建故障以在 SOAP 响应中发送用户定义的错误。创建预定义故障时，Developer tool 会将故障转换添加到操作映射。

创建 Web 服务，并定义一个或多个操作。可以在创建操作时创建故障，也可以将故障添加到操作。一个操作可以具有多个故障。

1. 打开一项 Web 服务。
2. 在**大纲**视图中，选择操作映射。

此时操作映射将显示在编辑器中。
3. 单击**属性**视图中的**操作**选项卡。
4. 单击**新建**按钮旁的箭头。
5. 单击**故障**。
6. 选择创建元素或重用元素。
 - 选择**创建新元素**，以便为故障创建元素。可以创建子元素和多次出现的元素。
 - 选择**重用现有元素**，以重用故障的元素。

Developer tool 会将元素添加到故障消息层次结构的详细信息元素中。

步骤 5.创建表头

可以选择创建表头以在 SOAP 消息的表头中接收或发送数据。可以为操作输入或操作输出创建表头。可以使用 HTTP POST 方法。

1. 打开一项 Web 服务。
2. 选择**大纲**视图中的操作映射。

此时操作映射将显示在编辑器中。
3. 单击**属性**视图中的**操作**选项卡。
4. 选择将表头添加到操作输入或操作输出中。
 - 选择输入以将表头添加到操作输入中。
 - 选择输出以将表头添加到操作输出中。

5. 单击**新建**按钮旁的箭头。
6. 单击**表头**，然后选择创建元素或重用元素。
 - 选择**创建新元素**，以便为表头创建元素。
 - 选择**重用现有元素**，以重用表头的元素。

第 6 章

操作映射

本章包括以下主题：

- [操作映射概览, 45](#)
- [操作映射常规选项卡, 46](#)
- [操作映射操作选项卡, 46](#)
- [操作映射高级选项卡, 46](#)
- [输入转换, 46](#)
- [输出转换, 48](#)
- [故障转换, 51](#)
- [故障处理, 54](#)
- [测试操作映射, 57](#)
- [自定义视图选项, 57](#)

操作映射概览

Informatica SOAP Web 服务包含操作映射。操作映射可为 Web 服务客户端执行 Web 服务操作。

操作映射可包含输入转换、输出转换和多个故障转换。输入、输出和故障转换可处理 SOAP 消息。映射还包含可基于客户端请求的 Web 服务操作检索、转换或更新数据的其他转换。

在 Developer tool 中创建 Web 服务后，为 Web 服务中的每个操作配置操作映射。操作映射代表操作的逻辑。

配置操作映射后，定义数据集成服务处理其在 SOAP 请求中接收的数据的方式。SOAP 请求可以是 SOAP 1.1 或 SOAP 1.2 格式，取决于与操作映射关联的绑定操作所使用的绑定类型。

输入转换从 Web 服务接收 SOAP 请求，然后将数据返回到映射中的转换下游。转换执行客户端请求的操作。

输出转换接收数据以返回到客户端。输出转换生成 SOAP 响应消息以发送到客户端。

如果出现错误，则数据集成服务会出现故障。数据集成服务从故障转换中返回用户定义的故障。用户定义的故障包含两种类型的故障：预定义故障和通用故障。

要配置操作映射，请完成以下步骤：

1. 配置输入、输出和故障转换。
2. 创建和配置其他转换以实施操作逻辑。
3. 链接端口。

4. 验证并保存该映射。
- 可以在操作映射**属性**视图的**常规**选项卡、**操作**选项卡和**高级**选项卡上查看和配置操作映射属性。

操作映射常规选项卡

在操作映射**属性**视图的**常规**选项卡上配置操作映射名称和说明。

常规选项卡还可显示与操作映射关联的操作的名称。

操作映射操作选项卡

在操作映射**属性**视图的**操作**选项卡上查看或配置操作属性。

手动创建 Web 服务时，可以使用**操作**选项卡定义故障或更新操作的输入或输出。您也可以为操作输入或操作输出添加表头。

操作映射高级选项卡

在**属性**视图的**高级**选项卡上配置操作映射高级属性。配置高级属性时，您可以指定是否要让数据集成服务验证 SOAP 请求的 XML。

下表介绍了操作映射的高级属性：

属性	说明
XML 架构验证	在运行时验证 SOAP 请求消息。选择 XML 无效错误或不验证 。当 XML 无效时，数据集成服务会在 SOAP 响应中返回故障，并在 Web 服务运行时日志中记录错误。

输入转换

输入转换代表 Web 服务 WSDL 中的输入元素和表头元素。输入转换从客户端接收 SOAP 请求。它将 XML 消息解析到关系数据组，并且将数据传递到操作映射中的其他转换。

在您为 Web 服务定义操作输入时，Developer tool 会创建输入转换。

使用输入转换**端口**选项卡查看操作输入层次结构、定义输出端口以及将数据从操作输入映射到输出端口。操作输入层次结构定义了 SOAP 请求消息层次结构。

可以将完整的 SOAP 请求映射为 XML，而不是在不同的输出端口中返回关系数据组。将 SOAP 请求映射为 XML 后，数据集成服务会在一个输出端口中返回完整的 SOAP 消息。

输入转换端口选项卡

在**端口**选项卡上定义输出组、定义输出端口以及将节点从操作输入映射到输出端口。

如果您不需要查看操作输入层次结构，可选择显示端口。显示端口时，可以定义组、定义端口，以及将节点从操作输入映射到输出端口。要将节点从操作输入映射到输出端口，请单击**位置**列中的字段，并在**选择位置**对话框中展开层次结构。然后，从层次结构中选择一个节点。

选择显示输入映射，以查看操作输入层次结构。该选项卡的左侧是**操作输入**区域，选项卡右侧是**端口**区域。**操作输入**区域显示 SOAP 请求消息层次结构。您可以在**端口**区域定义输出端口。将节点从操作输入映射到输出端口时，该节点的位置将显示在**操作输入**区域的**位置**列中。

显示输入映射时，可以选择显示层次结构中的输出端口。也可以选择查看将输入端口与操作输入中的节点连接在一起的行。

如果选择映射层次结构的第一级，则 Developer 工具会将第一级操作输入中的节点映射到输出端口。Developer 工具同时会创建映射数据所需的输出端口。如果层次结构的第一级包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer 工具不会创建端口或映射该层次结构的第一级。

将操作输入映射到端口的规则和指导

- 配置输入转换时，将操作输入层次结构中的节点映射到输出端口。
- 将操作输入层次结构中的节点映射到输出端口时，请注意以下规则和指导：
- 节点和输出端口的数据类型必须兼容。
 - 不能将一个节点映射到组中的多个输出端口。

配置输入转换

- 如果 Web 服务操作接收到 SOAP 请求消息，请配置输入转换以处理该请求。在转换中定义输出端口。将操作输入的节点映射到输出端口。
1. 在编辑器中选择输入转换。
 2. 单击**属性**视图的**端口**选项卡。
 3. 单击**输入映射**。
操作输入区域显示请求消息层次结构。在**端口**区域中定义输出端口。
 4. 或者，单击**显示行**以查看将输出端口连接到操作输入中节点的行。
可以选择查看所有行或查看选定端口的行。
 5. 或者，单击**显示为层次结构**以显示层次结构中的输出端口。
每个子组都显示在父组下方。
 6. 如果操作输入包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输入**区域中选择对象。在节点的**类型**列中，单击**选择**，然后从列表中选择一个或多个类型、元素或属性。
 7. 要添加输出组，请使用以下方法之一：

选项	说明
拖动节点	从 操作输入 区域的组节点或子节点将指针拖动到 端口 区域的空列中。如果该节点是组节点，则 Developer tool 将添加组，而不添加端口。如果存在其他输出组， 映射到新组 对话框将提示您将该组关联到其他组。Developer tool 将为关联组创建键。

选项	说明
手动添加组	单击 新建 > 组 以添加组。
选择 映射第一级层次结构	选择 映射第一级层次结构 。Developer tool 会将第一级操作输入中的节点映射到输出端口和组。Developer tool 同时会创建映射数据所需的输出端口和组。如果第一次层次结构包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer tool 不会创建端口或映射第一级层次结构。

8. 要添加输出端口并将节点映射到输出端口，请使用以下方法之一：

选项	说明
拖动节点	从操作输入的一个节点将指针拖动到 端口 区域中的组名称或端口。
单击 映射按钮	在 操作输入 区域中选择一个或多个节点。在 端口 区域中选择一个目标。单击 映射 。
复制端口	从其他转换中选择端口，并将其复制到 操作输入 区域。要复制端口，可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。
手动添加端口	单击 新建 > 字段 以添加端口。
选择 映射第一级层次结构	选择 映射第一级层次结构 。Developer tool 会将第一级操作输入中的节点映射到输出端口和组。Developer tool 同时会创建执行映射所需的输出端口和组。如果第一次层次结构包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer tool 不会创建端口或映射第一级层次结构。

9. 要清除端口的位置，请使用以下方法之一：

选项	说明
单击 清除按钮	在 端口 区域中选择一个或多个端口，然后单击 清除 。
删除将节点连接到端口的行	在操作输入中选择将节点连接到输出端口的一个或多个行，然后按 删除 。

10. 要将完整的 SOAP 请求映射为 XML，请在**操作输入**区域中右键单击**请求节点**，然后选择**映射为 XML**。

输出转换

输出转换代表 Web 服务 WSDL 中的输出元素和表头元素。输出转换会从操作映射中的关系数据组创建 SOAP 响应消息。WSDL 描述了大型 SOAP 消息层次结构，但 Web 服务操作可能会返回 SOAP 响应消息的部分数据。

定义 Web 服务操作输出后，Developer tool 会创建输出转换。

使用输出转换**端口**选项卡可查看操作输出层次结构，添加转换输入端口，以及将输入端口映射到操作输出。操作输出层次结构定义了 SOAP 响应消息层次结构。

可以将一个字符串或文本输入端口中的 XML 数据映射到整个 SOAP 响应。如果将 XML 数据映射到整个 SOAP 响应，则无法将端口映射到操作输出中的节点。

在输出转换**高级**选项卡中配置高级属性。

输出转换端口选项卡

可在**端口**选项卡上定义输入组，定义输入端口，以及将输入端口映射到操作输出节点。

显示端口时，可以手动添加组和端口，或将端口从其他转换复制到输出转换。可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。

显示输出映射时，可以定义输入组，定义输入端口，以及将输入端口映射到操作输出层次结构。该选项卡的左侧是**端口**区域，选项卡右侧是**操作输出**区域。**操作输出**区域显示 SOAP 响应消息层次结构。可在**端口**区域中定义输入组和输入端口。将**端口**区域中的输入端口映射到**操作输出**区域中的节点时，操作输出区域中的**位置**列中会显示输入端口位置。

如果选择映射第一级层次结构，则 Developer tool 会将输入端口映射到第一级操作输出中的节点。Developer tool 同时会创建映射数据所需的输入端口。如果第一次层次结构包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer tool 不会创建端口或映射第一级层次结构。

显示输出映射时，可以选择查看将输入端口连接到操作输入中的节点的行。

输出转换高级选项卡

在**高级**选项卡上配置输出转换高级属性。

下表介绍了可在输出转换**高级**选项卡上配置的属性：

属性	说明
已排序输入	允许数据集成服务在未处理所有输入数据的情况下生成输出。当输入数据按操作输入层次结构中的键排序时，请启用已排序输入。默认为已禁用。

将端口映射至操作输出的规则和准则

配置输出转换时，可以将端口映射至操作输出层次结构。

将输入端口映射至操作输出层次结构时，请遵守以下规则和准则：

- 输入端口和节点的数据类型必须兼容。
- 可以将输入端口映射到层次结构中的一个节点。
- 可以将一个输入组中的端口映射到操作输出中同一层次结构级别上的节点。
- 可以将一个输入组中的不同端口映射到操作输出中不同层次结构级别上的节点。
- 将输入端口映射到操作输出中的键。映射到键的任何端口都必须为字符串、整型或长整型数据类型。将数据映射到操作输出中位于 SOAP 消息所含层次结构级别之上的所有级别中的键。对于映射级别及之上的所有级别，需要包括外键。

注意：如果只计划映射操作输出层次结构的最低级别，则无需将输入端口映射到键。

- 可以将多个不同数据类型的输入端口映射到键。单击键的**位置**字段时，可以重新排列输入端口或删除其中一个端口。

配置输出转换

如果 Web 服务操作返回响应消息，请配置输出转换。定义每个转换中的输入端口，并将输入端口中的数据映射到操作输出层次结构中的节点。

1. 在编辑器中选择输出转换。
2. 单击**属性**视图的**端口**选项卡。

3. 单击**输出映射**。

操作输出区域显示操作输出层次结构。在**端口**区域中定义输入端口。

4. 或者，单击**显示行**以查看将输入端口连接到操作故障中节点的行。可以选择查看所有行或查看选定端口的行。

5. 如果操作输出包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输出**区域中选择对象。在节点的**类型**列中，单击**选择**，然后从列表中选择一个或多个类型、元素或属性。

6. 要添加输入组，请使用以下方法之一：

选项	说明
拖动节点	从 操作输出 区域的组节点或子节点将指针拖动到 端口 区域的空列中。如果该节点是组节点，则 Developer tool 将添加组，而不添加端口。
手动添加组	单击 新建 按钮旁边的箭头，然后单击 新建组 。
选择 映射第一级层次结构	选择 映射第一级层次结构 。Developer tool 会将第一级操作输入中的节点映射到输出端口和组。Developer tool 同时会创建映射数据所需的输入端口和组。

7. 要添加输入端口，请使用以下方法之一：

选项	说明
手动添加端口	单击 新建 按钮旁边的箭头，然后单击 新建端口 。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到输出转换。
复制端口	从其他转换中选择端口，并将其复制到 操作输出 区域。要复制端口，可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。
选择 映射第一级层次结构	选择 映射第一级层次结构 。Developer tool 会将第一级操作输出中的节点映射到输入端口和组。Developer tool 同时会创建执行映射所需的输入端口和组。

8. 要将输入端口中的数据映射到操作输出层次结构中的节点，请从每个输入端口或组将指针拖动到操作输出中的关联节点。输入字段位置显示在**操作输出**区域中的节点旁。

9. 要将输入端口中的 XML 数据映射到完整的 SOAP 响应，请右键单击该端口，然后选择**映射为 XML**。

10. 要将输入端口映射为复合键，请使用以下方法之一：

选项	说明
拖动输入端口	选择两个或更多个输入端口，然后将它们拖动到操作输出层次结构中的键。
从 选择位置 对话框中选择输入端口	在操作输出层次结构中单击键的 位置 列，然后选择输入端口。

11. 要清除节点的位置，请使用以下方法之一：

选项	说明
单击 清除 按钮	在 操作输出 区域中选择一个或多个节点，然后单击 清除 。

选项	说明
删除将端口连接到节点的行	在操作输出中选择将输入端口连接到节点的一个或多个行，然后按 删除 。

故障转换

故障转换表示 Web 服务 WSDL 中的故障元素。故障转换会在 Web 服务操作中生成用户定义的故障。

故障转换表示以下任一用户定义故障：

预定义故障

在预定义故障中，故障转换表示 Web 服务 WSDL 中的故障元素。在预定义故障中，故障转换通过 Web 服务操作映射中的关系数据来创建错误消息。

通用故障

在通用故障中，故障转换不表示 Web 服务 WSDL 中为 Web 服务操作定义的任何故障元素。当转换中出错时，通用故障的故障转换会返回通用错误消息。可以在 Developer tool 中为通用故障创建和配置故障转换。

一个操作映射可以包含多个故障转换。可以在映射中添加同一故障转换的多个实例，以便在不同映射部分中生成相同消息。

可以在映射中添加或删除故障转换，而无需更改操作签名。如果您知道在映射运行时永远不会出现故障错误，则可以从映射中删除故障转换。必须将故障转换连接到上游转换，否则映射无效。

在包含 SOAP 1.1 绑定的操作中创建故障时，向导会创建故障代码、故障字符串和故障触发者元素。在包含 SOAP 1.2 绑定的操作中创建故障时，向导会创建代码、原因、节点和角色元素。将元素添加到故障时，向导会将元素添加到故障中的详细信息组。使用故障转换**端口**选项卡查看操作故障层次结构，添加转换输入端口，并将输入端口映射到操作故障。操作故障层次结构为因用户定义的故障而产生的错误消息定义 SOAP 响应消息层次结构。

可以将一个字符串或文本输入端口中的 XML 数据映射到整个 SOAP 响应。如果将 XML 数据映射到整个 SOAP 响应，则无法将端口映射到操作故障中的节点。

在故障转换**高级**选项卡中配置高级属性。

故障转换端口选项卡

在**端口**选项卡上定义输入组，定义输入端口，并将输入端口映射到操作故障节点。

显示端口时，可以手动定义组和端口。或者，可以将端口从其他转换复制到故障转换。可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。

显示故障映射时，可以定义输入组，定义输入端口，以及将输入端口映射到操作故障层次结构。该选项卡的左侧是**端口**区域，选项卡右侧是**操作故障**区域。**操作故障**区域显示 SOAP 响应消息层次结构。将“端口”区域中的输入端口映射到**操作故障**区域中的节点时，输入端口位置将显示在**操作故障**区域的**位置**列中。

如果选择映射第一级层次结构，则 Developer tool 会将输入端口映射到第一级操作故障中的节点。Developer tool 同时会创建映射数据所需的端口。

故障转换高级选项卡

在高级选项卡上配置故障转换高级属性。

下表介绍了可在高级选项卡上配置的属性：

属性	说明
已排序输入	允许数据集成服务在不必处理所有输入数据的情况下生成输出。当输入数据按操作故障层次结构中的键排序时，请启用已排序输入。默认为已禁用。

将端口映射到操作故障的规则和指导

配置故障转换时，将输入端口映射到操作故障层次结构。

将输入端口映射到操作故障层次结构时，请注意以下规则和指导：

- 可以将输入端口映射到操作故障层次结构中的一个节点。输入端口和节点的数据类型必须兼容。
- 可以将一个输入组中的端口映射到操作故障中同一个层次结构级别中的节点。
- 可以将一个输入组中的不同端口映射到操作故障中不同层次结构级别中的节点。
- 必须将输入端口数据映射到操作故障层次结构中的键。映射到键的任何端口的数据类型都必须为字符串、整型或长整型。将数据映射到操作故障中位于 SOAP 消息所含层次结构级别之上的所有级别中的键。对于映射级别及之上的所有级别，需要包括外键。
- 可以将多个不同数据类型的输入端口映射到键。单击键的位置字段时，可以重新排列输入端口或删除其中一个端口。

创建故障转换

可以创建通用故障或预定义故障转换。对于通用故障，Web 服务 WSDL 未定义故障元素。对于预定义故障，Web 服务使用故障元素定义故障。

1. 打开一项 Web 服务。
2. 在大纲视图中，选择操作映射。
此时操作映射将显示在编辑器中。
3. 在编辑器中右键单击，然后选择**添加转换**。
此时将显示**添加转换**对话框。
4. 选择**故障**，然后单击**确定**。
此时将显示**添加故障**对话框。
5. 要创建故障转换，请完成以下步骤之一。
 - 选择**创建为通用故障**。
 - 选择**从故障元素创建为预定义故障**。
注意： Web 服务包含用于定义故障的元素。
6. 单击**确定**。
故障转换将显示为通用故障或预定义故障。

配置故障转换

如果 Web 服务操作返回故障，或者如果为通用故障创建故障转换，请配置每个故障转换。定义输入端口，并将输入端口中的数据映射到操作故障中的节点。

- 1. 在编辑器中选择故障转换。
- 2. 单击属性视图中的端口选项卡。
- 3. 单击故障映射。

操作故障区域显示响应或故障消息层次结构。在端口区域中定义输入端口。

- 4. 或者，单击显示行以查看将输入端口连接到操作故障中节点的行。可以选择查看所有行或查看选定端口的行。
- 5. 如果操作故障包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在操作故障区域中选择对象。在节点的类型列中，单击选择，然后从列表中选择一个或多个类型、元素或属性。
- 6. 要添加输入组，请使用以下方法之一：

选项	说明
拖动节点	从操作故障区域的组节点或子节点将指针拖动到端口区域的空列中。如果该节点是组节点，则 Developer tool 将添加组，而不添加端口。
手动添加组	单击新建按钮旁边的箭头，然后单击新建组。
选择映射第一级层次结构	选择映射第一级层次结构。Developer tool 会将第一级操作故障中的节点映射到输入端口和组。Developer tool 同时会创建映射数据所需的输入端口和组。

- 7. 要添加输入端口，请使用以下方法之一：

选项	说明
手动添加端口	单击新建按钮旁边的箭头，然后单击新建端口。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到故障转换。
复制端口	从其他转换中选择端口，并将其复制到操作故障区域。要复制端口，可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。
选择映射第一级层次结构	选择映射第一级层次结构。Developer tool 会将第一级操作故障中的节点映射到输入端口和组。Developer tool 同时会创建映射数据所需的输入端口和组。

- 8. 要将输入端口中的数据映射到操作故障层次结构中的节点，请从每个输入端口或组将指针拖动到操作故障中的关关节点。

输入字段位置显示在操作故障区域中的节点旁。

- 9. 要将输入端口中的 XML 数据映射到完整的 SOAP 响应，请右键单击该端口，然后选择映射为 XML。
- 10. 要将输入端口映射为复合键，请使用以下方法之一：

选项	说明
拖动输入端口	选择两个或更多个输入端口，然后将它们拖动到操作故障层次结构中的键。

选项	说明
从 选择位置 对话框中选择输入端口	在操作故障层次结构中单击键的 位置 列，然后选择输入端口。

11. 要清除节点的位置，请使用以下方法之一：

选项	说明
单击 清除 按钮	在 操作故障 区域中选择一个或多个节点，然后单击 清除 。
删除将端口连接到节点的行	在操作故障中选择将输入端口连接到节点的一个或多个行，然后按 删除 。

故障处理

当 Web 服务中出错时，数据集成服务会生成错误消息，并以故障的形式将此消息返回给 Web 服务客户端。

如果将 SOAP 1.1 请求发送给包含 SOAP 1.2 绑定的操作，Web 服务将使用 SOAP 1.1 生成故障。如果将 SOAP 1.2 请求发送给包含 SOAP 1.1 绑定的操作，Web 服务将使用 SOAP 1.2 生成故障。Web 服务可以生成系统定义的故障和用户定义的故障。

当出现用户定义的错误时，数据集成服务会向 Web 服务客户端返回一条错误消息。用户定义的故障可以是以下类型之一：

- 预定义
- 通用

当操作映射包含输出转换时，Web 服务将返回来自输出转换的数据，否则返回故障。如果在操作映射将数据提交至目标转换或外部应用程序之后发生故障，数据集成服务将无法回滚数据。映射将停止，且数据集成服务将放弃 Web 服务输出转换已接收到的数据。

SOAP 1.1 故障

对于 SOAP 1.1，故障是指具有以下结构的 SOAP 消息：

```
Fault (FaultName)
Key_Fault (FaultName)
faultcode      xs:QName
faultstring    xs:string
faultactor    xs:anyURI
detail
```

故障包含以下元素：

故障代码

故障标识代码，如错误消息编号。

故障字符串

错误说明。

故障触发者

导致故障发生的对象的相关可选信息。

详细信息

因故障而异的可选信息。

SOAP 1.2 故障

对于 SOAP 1.2，故障是指具有以下结构的 SOAP 消息：

```
Fault (FaultName)
Key_Fault (FaultName)
Code                tns:faultcodeEnum
Reason              tns:reasonText
Node                xs:anyURI
Role                xs:anyURI
detail
```

故障包含以下元素：

代码

故障标识。

代码的“值”元素必须是以下值之一：

- infasoapns:DataEncodingUnknown
- infasoapns:MustUnderstand
- infasoapns:Receiver
- infasoapns:Sender
- infasoapns:VersionMismatch

注意：可以展开“代码”故障元素，以便将“子代码”故障元素提升一个级别。“子代码”的架构类型为 xsd:QName。可以使用“子代码”故障元素来定义错误消息编号。

原因

错误说明。

节点

包含生成故障的 SOAP 节点的 URI。

角色

导致故障发生的对象的相关可选信息。

详细信息

因故障而异的可选信息。

系统定义的故障

数据集成服务在遇到系统错误时会生成一个系统定义的故障。出现系统错误时，数据集成服务在系统定义的故障中返回一条错误消息。

例如，当数字输入端口接收非数字数据时，数据集成服务可能返回以下故障：

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>WSCMN_10022</faultcode>
      <faultstring>[WSCMN_10022] Mapping execution failed:
[com.informatica.platform.ldtm.common.ExecutionException: [MPSVCCMN_10009] The Mapping Service Module
[MappingService] encountered an exception with the following details: [LDTM_0072] [ERROR] XML parsing
```

```

component [Input_S2R] message code: [66022], message body: A data conversion error occurred in field
[customerID] of the transformation group [Operation]. The field contains the following data:
[1100AA].].</faultstring>
<detail>                                </detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

注意: 该示例使用 SOAP 1.1。

可以在不使用故障转换的情况下将错误消息返回到 Web 服务客户端。可以通过在表达式转换中调用 ABORT(msg) 函数来引发故障。调用 ABORT 消息时，数据集成服务生成一个系统定义的故障消息。传递到 ABORT 函数的消息成为系统定义的故障中的故障字符串。

预定义故障

对于预定义故障，Web 服务使用故障元素定义故障。将故障转换配置为返回自定义错误消息。

在 Web 服务中配置故障转换时，必须定义返回错误条件的操作映射逻辑。需要定义一个转换，以生成传递到故障转换的错误消息。

定义故障转换时，需定义在故障代码、故障字符串和故障参与者中返回的数据。还可以在故障消息中添加更多元素，以便将更多信息返回到 Web 服务客户端。定义故障中的元素时，向导会将这些元素添加到详细信息组中。可以定义多次出现的元素，并创建详细信息组中元素组之间的层次结构关系。

如果每个错误的错误消息结构都相同，则可以在一个映射中多次使用同一个故障转换。否则，可以为要返回到 Web 服务客户端的每个错误消息配置不同的故障转换。

例如，可以配置一个故障转换，当在查找中未找到某位员工时，可以将消息返回到 Web 服务客户端。将错误编号传递给 faultcode 元素，将错误消息传递给 faultstring 元素。还需要将部门 ID 和员工 ID 返回到 Web 服务客户端。定义故障时，需要在详细信息组中添加 DeptID 和 EmployeeID 元素。

数据集成服务可返回以下故障：

```

<infasoapns:Envelope xmlns:infasoapns="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:infawsdlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.informatica.com/dis/ws/
Get_Employee_Info_Web_Service">
  <infasoapns:Body>
    <infasoapns:Fault>
      <faultcode>ERR_12345</faultcode>
      <faultstring>Web service failed to retrieve employee information.</faultstring>
      <detail>
        <tns:Employee_Not_Found>
          <tns:DeptID>100</tns:DeptID>
          <tns:EmployeeID>2428</tns:EmployeeID>
        </tns:Employee_Not_Found>
      </detail>
    </infasoapns:Fault>
  </infasoapns:Body>
</infasoapns:Envelope>

```

注意: 该示例使用 SOAP 1.1。

通用故障

可以定义通用故障，以便在未通过 WSDL 中的故障元素定义错误时，可向 Web 服务客户端返回错误消息。创建故障转换，以便在转换中出错时返回通用错误消息。

为 Web 服务中的通用故障创建故障转换时，必须定义返回错误条件的操作映射逻辑。

为通用故障定义故障转换时，如果操作绑定使用 SOAP 1.1 格式，则可以将数据定义为以故障代码、故障字符串和故障触发者元素形式返回。如果操作绑定使用 SOAP 1.2 格式，则可以将数据定义为以代码、原因、节点和角色元素形式返回。还可以选择将数据定义为以详细信息字符串形式返回。详细信息字符串是可选的，且具有

xsd:any 元素。如果要发送数据到详细信息字符串，请将详细信息元素映射到 Web 服务操作中定义的某个端口。还可以在故障消息中添加更多元素，以便将更多信息返回给 Web 服务客户端。

可以在详细信息组中的元素组之间创建层次结构关系。如果每个错误的错误消息结构都相同，则可以在一个映射中多次使用同一个故障转换。还可以为要返回给 Web 服务客户端的每个错误消息创建不同的故障转换。

注意：创建故障转换后，将无法更改通用故障或预定义故障的状态。

测试操作映射

预览操作映射的输出以验证其是否生成您希望的结果。还可以预览映射中的转换的输出。

创建 Web 服务配置以控制在您预览操作映射的输出或操作映射中的转换的输出时 Developer tool 应用的设置。使用**首选项**对话框配置默认的 Web 服务运行配置。还可以使用**运行**对话框创建可在**数据查看器**视图中指定的 Web 服务配置。

测试操作映射

测试操作映射，以预览操作的 SOAP 响应。您可以预览操作映射的输出，也可以预览来自映射中转换的输出。

预览数据前，需要选择默认数据集成服务。

1. 打开一项 Web 服务。
2. 选择**大纲**视图中的操作映射。
此时操作映射将显示在编辑器中。
3. 选择**数据查看器**视图。
4. 在**输入**窗口中输入一个请求。
5. 如果您想要查看特定转换的输出数据，请在编辑器中选择该转换。否则，操作映射的输出会在运行数据查看器时显示。
6. 单击**运行**。
请求的结果将显示在**输出**窗口中。
7. 要使用其他请求测试操作映射，请单击**重置**，然后重复步骤 4 至 6。

自定义视图选项

配置输入、输出以及故障转换时，可以更改 SOAP 消息层次结构，以便在**操作**区域中显示键。此外，还可以显示用于定义节点排序方式的分组构造。

要查看自定义视图选项，请单击**操作输入**区域、**操作输出**区域或**操作故障**区域中的**自定义视图**。

您可以启用以下选项：

全部、序列和选项

显示一行，指示元素定义是“全部”、“序列”还是“选项”。

“all”组中的节点都必须包含在 SOAP 消息中。

“sequence”组中的节点必须符合 WSDL 中指定的顺序。

“choice”组中必须至少有一个节点显示在 SOAP 消息中。

键

查看每个层次结构级别的键。

第 7 章

解析 Web 服务 SOAP 消息

本章包括以下主题：

- [解析 Web 服务 SOAP 消息概览, 59](#)
- [转换用户界面, 59](#)
- [多次出现输出配置, 60](#)
- [解析 anyType 元素, 62](#)
- [解析派生类型, 63](#)
- [解析 QName 元素, 64](#)
- [解析置换组, 65](#)
- [解析 SOAP 消息中的 XML 构造, 65](#)

解析 Web 服务 SOAP 消息概览

数据集成服务在解析 Web 服务转换中的 SOAP 消息时生成行数据。

Web 服务输入转换和 Web 服务使用者转换是解析 SOAP 消息的 Web 服务转换。

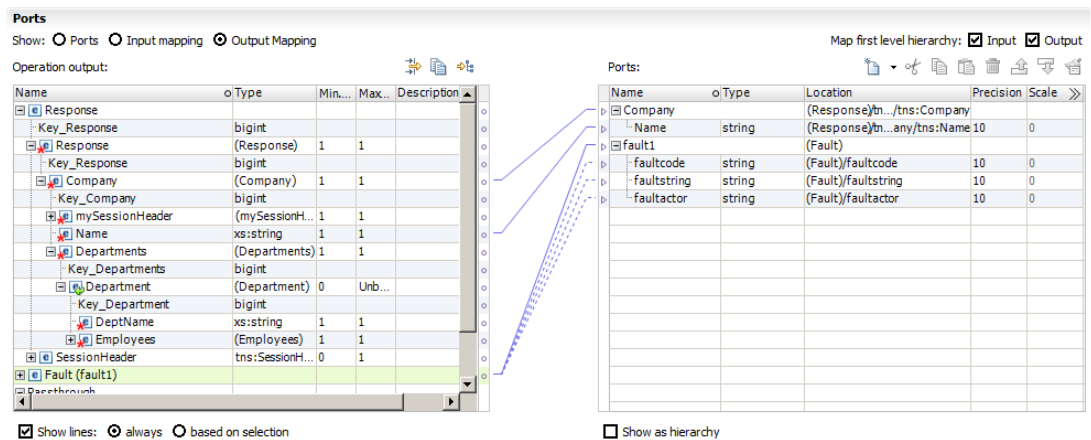
要将转换配置为解析 SOAP 消息，可以采用类似于 SOAP 消息层次结构的结构创建输出端口。将 SOAP 消息层次结构中的节点映射到端口。

可以配置规范化输出端口组、非规范化组和透视端口组。如果 SOAP 消息包含派生类型、anyType 元素或置换组，可以根据 SOAP 消息实例中可能出现的类型配置不同的输出组。

转换用户界面

Web 服务使用者转换和 Web 服务输入转换提供了一个用户界面，可用来将数据从 SOAP 消息映射到转换输出端口。

下图显示了 SOAP 1.1 消息节点与 Web 服务使用者转换中的输出端口之间的映射：



操作区域

操作区域包含 SOAP 消息层次结构。复杂节点或多次出现节点定义结构中的层次结构级别。Developer 在级别中添加键，用以定义这些级别之间的父子关系。

在上图中，SOAP 消息层次结构具有以下级别：

响应或请求

该级别表示响应或请求消息的根。

公司

请求数据的最高级别。

部门

公司内的多次出现部门。

员工

员工是部门内的一个复杂元素。

故障组

接收错误消息的故障消息组。

端口区域

可以将数据从 SOAP 消息级别映射到输出端口。每个输出端口组均可通过主外键关系与其他输出组关联。

在上图中，转换有多个与 SOAP 消息中的节点组相对应的输出端口组。

多次出现输出配置

当输入转换或 Web 服务使用者转换返回多次出现数据时，可以将输出端口配置为不同的配置。

可以配置规范化输出数据、透视输出数据或非规范化输出数据。

例如，SOAP 消息包含部门和员工复杂元素。每个部门包含多个员工。部门是员工的父级别。

SOAP 消息包含以下元素层次结构：

```
Departments
  Department_ID
  Department_Name
  Employees
```

```
Employee_ID
Employee_Name
```

规范化的关系输出

创建规范化输出数据时，数据值不会在输出组中重复。在 SOAP 消息中的层次结构级别和输出端口组之间创建一对一关系。

如果 SOAP 消息包含部门父级层次结构级别和员工子级层次结构级别，可创建以下端口组：

```
Departments
  Department_Key
  Department_ID
  Department_Name

Employees
  Department_Key
  Employee_ID
  Employee_Name
```

Department_Key 是一个将 Employees 输出组与 Department 组相关联的生成键。

生成的键

添加输出组时，Developer 工具通过生成的键将输出组与其他输出组关联。Developer 工具在父级组和子级组中添加长整型键。数据集成服务在运行时创建生成的键的键值。

示例

SOAP 层次结构含有以下节点：

```
Departments
  Dept_Key
  Dept_Num
  Dept_Name

Employees
  Dept_FK
  Employee_Num
  Employee_Name
```

为部门创建输出端口组时，将部门节点映射到端口区域中的一个空字段。Developer 工具将创建以下输出组：

```
Departments
  Dept_Num
  Dept_Name
```

如果将员工节点映射到端口区域中的一个空字段，Developer 工具会提示您将员工组与部门组关联。可以将员工组与多个组关联。Developer 工具在每个组中添加一个键。

Developer 工具创建以下组和生成的键：

```
Departments
  Key_Departments
  Dept_Num
  Dept_Name

Employees
  Key_Departments
  Employee_Num
  Employee_Name
```

注意：不必将节点映射到生成的键。数据集成服务在运行时创建键值。

Developer 工具可以在一个输出组中创建多个级别的生成的键。员工组可包含以下端口：

```
Employees
  Key_Employees
  Key_Departments
  Key_Managers
  Employee_Num
  Employee_Name
```

Key_Departments 和 Key_Managers 是指向父级组的生成的键。Key_Employees 是针对员工组的生成的键。将子级组关联到员工组时，将显示 Key_Employees。

非规范化的关系输出

可以对关系输出进行非规范化处理。对输出数据进行非规范化处理时，父级组中的元素值会对每个子元素重复。

要对输出数据进行非规范化，请将父级层次结构中的节点映射至子级输出端口组。

以下示例显示了 Employees 输出组中的 Department_ID 和 Department_Name：

```
Employees
  Department_ID
  Department_Name
  Employee_ID
  Employee_Name
```

对于部门中的每名员工，Department_ID 和 Department_Name 均重复：

Department_ID	Department_Name	Employee_ID	Employee_Name
100	Accounting	56500	Kathy Jones
100	Accounting	56501	Tom Lyons
100	Accounting	56509	Bob Smith

已转换的关系输出

您可以在一个输出组中包含特定数量的多次出现的元素。

要转换多次出现的元素，请将多次出现的子元素映射至父级输出端口组。Developer 工具会提示您定义要包含在父级输出端口组中的子元素数量。

下例显示了部门父级组中的两个 Employee_ID 实例：

```
Departments
  Department_ID
  Department_Name
  Employee_ID1
  Employee_ID2
```

解析 anyType 元素

anyType 元素代表 WSDL 或架构中所有全局类型的一个选项。将节点映射到 Developer 工具中的端口时，选择要在 SOAP 消息中为 anyType 元素显示的类型。必须将 SOAP 消息中的 anyType 元素替换为复杂类型或 xs:string。为所选的每个类型创建端口组。

必须选择类型以将数据映射到输出端口。如果 WSDL 或架构不包含全局类型，Developer 工具会将 anyType 元素替换为 xs:string。

要在操作区域中选择元素类型，请在 anyType 元素的**类型**列中单击**选择**。此时将显示一系列可用的复杂类型和 xs:string。

将 anyType 元素替换为派生类型时，数据集成服务一次为一个类型填充元素。SOAP 消息不同时包含基本类型和派生类型的数据。

派生类型示例

WSDL 包含 anyType 元素。将该元素替换为 AddressType 和称为 USAddressType 的派生类型。SOAP 消息层次结构包含以下组：

```
Address:AddressType (base type)
  Address: AddressType
    Street
    City

Address:USAddressType (derived type)
  Street
  City
  State
  ZipCode
```

SOAP 消息包含以下数据：

```
<address xsi: type ="AddressType">
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>
```

数据集成服务为 xsi:AddressType 返回一个行：

街道	城市
1002 Mission St.	San Jose

数据集成服务为派生类型 xsi:USAddressType 返回一个行：

街道	城市	省/自治区/直辖市	邮政编码
234 Fremont Blvd.	Sunnyvale	CA	94556

如果类型为 xsi:USAddressType，数据集成服务将不填充 AddressType。

解析派生类型

可以解析包含派生类型的 SOAP 消息。定义从 SOAP 消息接收数据的端口时，选择可在 SOAP 消息中出现的类型。所选类型的元素决定了需要创建的端口。

例如，WSDL 可包含 AddressType 和称为 USAddressType 的派生类型。可以在 Developer 工具操作区域创建以下组：

```
Address
  Address: AddressType
    Street
    City
```

Address:USAddressType
Street
City
State
ZipCode

SOAP 消息可包含以下数据：

```
<address>
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>

<address xsi:type="USAddressType">
<street>100 Cardinal Way</street>
<city>Redwood City</city>
<zip>94536</zip>
<state>CA</state>
</address>

<address>
<street>100 El Camino Real</street>
<city>Sunnyvale</city>
</address>
```

对于基本类型 Address，数据集成服务返回以下行：

街道	城市
1002 Mission St.	San Jose
234 Fremont Blvd	Sunnyvale
100 Cardinal Way	Redwood City
100 El Camino Real	Sunnyvale

对于派生类型 USAddress，数据集成服务返回以下行：

街道	城市	省/自治区/直辖市	邮政编码
234 Fremont Blvd.	Sunnyvale	CA	94556
100 Cardinal Way	Redwood City	CA	94536

数据集成服务返回所有基本类型的地址。数据集成服务返回派生类型的 US 地址。派生类型包括 USAddressType 从基本类型继承的街道和城市元素。

解析 QName 元素

数据集成服务解析 SOAP 消息中的 QName 元素时，其更新属于架构命名空间的 QName 值，以使用在架构中定义的命名空间前缀。其他情况下数据集成服务不更新元素值。

例如，架构为命名空间 "http://user/test" 定义了命名空间前缀 tns。SOAP 消息为同一个命名空间定义了命名空间前缀 mytns。当数据集成服务解析 QName 值 mytns:myElement 时，其将该值改为 tns:myElement。

数据集成服务在 SOAP 消息中生成 QName 元素时，其不更新元素的值。

解析置换组

置换组将一个元素替换为同一组中的其他元素。置换组类似于派生类型，但每个元素定义都包含一个置换组名称。

可以配置从置换组中的特定类型接收元素的输出端口组。可以创建从置换组中的其他类型接收元素的不同输出端口组。

解析 SOAP 消息中的 XML 构造

SOAP 消息可能包含选项、列表和联合元素等 XML 构造。

Web 服务转换可以解析包含这些构造的 SOAP 消息，但受到一些限制。

选项元素

选项元素将子元素限制为 <choice> 声明中的一个元素。

下面的文字显示了一个身份为员工或承包商的人员元素：

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="contractor" type="contractor"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

可以使用以下方法映射选项元素：

- 为输出组中的每个选项元素创建输出端口。部分元素在输出行中为空值。
- 为每个选项创建输出组。对于上例，创建员工组和承包商组。数据集成服务根据在 SOAP 消息中显示的元素生成行。

列表元素

列表是指可以包含多个简单类型值的 XML 元素，例如“Monday Tuesday Wednesday”。

数据集成服务可以返回列表作为字符串值。SOAP 消息中包含列表时，无法将项目从列表映射到独立的输出行。如果需要在映射中隔离，可以配置一个表达式转换以隔离列表中的元素。

联合元素

联合元素是一种由多种类型联合组成的简单类型。

以下文本显示了由两种简单类型（size_no 和 size_string）联合组成的元素 Size：

```
<xs:element name="Size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>
```

要将 size 映射到输出端口，请为 size 创建一个端口。将输出端口配置为字符串。可以在映射中配置另一个转换，以将数据转换为其他类型。

第 8 章

生成 Web 服务 SOAP 消息

本章包括以下主题：

- [生成 Web 服务 SOAP 消息概览, 66](#)
- [转换用户界面, 67](#)
- [端口和层次结构级别关系, 68](#)
- [键, 69](#)
- [映射端口, 70](#)
- [转换多次出现的端口, 72](#)
- [映射非规范化数据, 73](#)
- [派生类型和元素置换, 74](#)
- [生成 SOAP 消息中的 XML 构造, 75](#)

生成 Web 服务 SOAP 消息概览

数据集成服务在生成 SOAP 消息时从几组输入数据生成 XML 数据。创建 Web 服务使用者转换、Web 服务输出转换或故障转换时，您将配置要映射到 SOAP 消息层次结构的输入端口。

要配置转换以生成 SOAP 消息，请创建几组输入端口，然后将每个组映射到 SOAP 消息层次结构中的某个组。WSDL 或架构定义 SOAP 消息的结构。

您可以在非规范化输入数据的 SOAP 消息中配置几组数据。还可以将多次出现的输入数据转换为 SOAP 消息中多次出现的节点。

可以将数据映射到 SOAP 消息中的派生类型、anyType 元素或置换组。定义转换时，必须选择可以出现在 SOAP 消息中的类型。所选类型决定需要创建的输入端口。

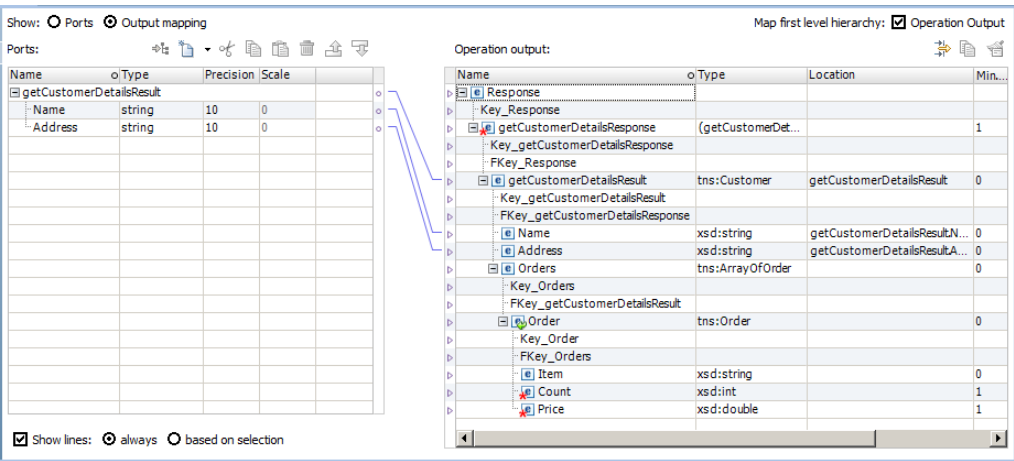
在 Developer 工具中查看 SOAP 消息层次结构时，该层次结构中包含键。这些键不会出现在 SOAP 消息中。数据集成服务使用键定义 SOAP 消息中各个组之间的父子关系。要配置键值，请将输入数据映射到 SOAP 消息中的键。

转换用户界面

Web 服务输出转换、故障转换和 Web 服务使用者转换中包含用于配置 SOAP 消息的用户界面。

要配置转换以生成 SOAP 消息，请在与 SOAP 消息层次结构相似的结构中创建输入端口。WSDL 或架构决定层次结构的结构。将每个输入端口映射到 SOAP 消息中的节点。

下图显示了输入端口与 Web 服务输出转换中的 SOAP 消息节点之间的映射：



输入端口区域

在输入端口区域中创建几组输入端口。将需要映射的 SOAP 消息层次结构中每个级别的输入端口包括在内。

必须创建用于接收数据的响应或请求输入组和子组。

创建输入端口组时，请在每个父组中定义一个主键。在每个子组中定义一个外键。外键将子组与父组关联。

除非要在 WSDL 根级别传递数据，否则不需要为响应级别或 WSDL 根级别定义键。例如，根级别可能包含 HTTP 表头。

您可能会创建与以下客户的组和顺序相似的端口组：

```
Response
  Response_Key

Customer_Details_Root
  Key_Cust_Det
  FK_Response_Key

Customer
  Customer_ID
  FK_Cust_Det
  Name
  Address

Orders
  Order_Num
  FK_Cust_ID

Order_Items
  Order_Num
  Item
  Count
  Price
```

操作区域

操作区域显示 SOAP 消息层次结构中由 WSDL 或架构定义的元素。 SOAP 消息不需要包含 WSDL 或架构中的所有元素。 该消息中包含您从输入端口映射的数据。

多次出现的节点和复杂节点定义 SOAP 消息结构中的层次结构级别。 Developer 工具将键添加到这些级别，以在各个级别之间创建父子关系。 层次结构中的所有级别（叶级别除外）都有一个主键。 每个子级别都有一个指向父级别的外键。 SOAP 消息层次结构中出现的键不会出现在 SOAP 消息实例中。 数据集成服务要求在生成 SOAP 消息时键中的值与数据的级别相关联。

位置列中包含组名称和输入端口，该端口中包含 SOAP 消息中的元素的数据。 将输入端口映射到节点之前，**位置**列始终为空。

在上图中，SOAP 消息包含客户详细信息和顺序的单个实例。“顺序”组包含多次出现的元素（名为 Order）。 SOAP 消息层次结构中包含具有关联键的以下级别：

```
Response
  GetCustomerDetailsResponse
    GetCustomerDetailsResult
      Orders
        Order
```

响应级别表示响应消息的根。 数据集成服务要求此级别将表头连接到 SOAP 消息。

GetCustomerDetailsResponse 级别是消息的根。

端口和层次结构级别关系

将输入端口映射到 SOAP 消息层次结构时，需要维护输入组与 SOAP 消息层次结构级别之间的关系。 例如，您可能有两个输入组，即“部门”和“员工”。

“部门”输入组接收以下各行：

Dept_num	名称	位置
101	HR	纽约
102	产品	加利福尼亚

“员工”输入组接收以下各行：

Dept_num	员工
101	Alice
101	Bob
102	Carol
102	Dave

映射“员工”组中的部门数量作为外键，用于在“部门”与“员工”组之间建立关系。 部门数量在部门层次结构级别出现，不在员工级别出现。

SOAP 消息中包含以下 XML 结构：

```
<department>
  <dept_num>101</dept_num>
  <name>HR</name>
  <location>New York</location>

  <employee>
    <name>Alice</name>
  </employee>

</employee>
```

```

        <name>Bob</name>
      </employee>
    </department>

    <department>
      <dept_num>102</dept_num>
      <name>Product</name>
      <location>California</location>

      <employee>
        <name>Carol</name>
      </employee>

      <employee>
        <name>Dave</name>
      </employee>
    </department>

```

键

SOAP 消息层次结构包括键。数据集成服务需要键值才能在 SOAP 消息中构建 XML 层次结构。

必须将输入端口数据映射到 SOAP 消息层次结构中的键。将数据映射到要提供数据的每个级别中的键。具有多次出现的节点时，您需要将该节点与父节点相关联。

键出现在 SOAP 消息中，不带类型。映射到键的任何端口都必须为 String、Integer 或 Bigint 数据类型。父组中的主键和每个子组中的外键必须具有相同的数据类型、精度和小数位数。可以将所生成的键映射到 SOAP 消息键。

可以将端口映射到同一层次结构级别中的节点和键。例如，您将 Employee_ID 映射到 SOAP 消息中的节点，然后将其映射到“员工”级别中的键。

如果层次结构中的两个组节点具有父子关系，请完成以下任务：

- 将端口映射到父节点组中的主键。
- 将端口映射到子节点组中的外键。

还必须将主键映射到输入端口，以删除带有空主键或带有重复主键的各个行。

可以通过将多个端口映射到同一个键，在 SOAP 消息中创建一个复合键。需要对数据进行非规范化和维护部分多次出现的值组合的唯一键时，请使用复合键。可以创建包含字符串值、长整型值或整数值值的复合键。

注意：可以在操作映射中包含表达式转换以生成键值。

复合键示例

从以下端口组配置唯一的 division-department 键：

```

Company
  Company_Num
  Company_Name

Division
  Company_Num
  Divison_Num
  Division_Name

Department
  Division_Num
  Dept_Num
  Dept_Name

```

Location

Dept_Num 在某个分支机构中是唯一的，但在公司下属的所有分支机构中不唯一。

您可能会配置一个包含分支机构和部门信息的“部门”组。配置分支机构编号和部门编号作为复合键的一部分：

```
Department
  Division_Num + Dept_Num (key)
  Dept_Name
  Location
```

映射端口的顺序决定键值。

映射端口

创建输入端口后，将每个输入端口映射到 SOAP 消息层次结构。端口的位置在**操作**区域中的节点旁边显示。

可以将端口映射到以下类型的节点：

基本节点

没有子元素或子属性且不可见的简单元素或属性。

多次出现的基本节点

在层次结构中的同一位置多次出现的简单元素或属性。

复杂节点

包含其他元素的元素。

如果父节点没有位置，则会接收输入组名称作为位置。父节点有位置时，层次结构级别中的每个节点都必须具有同一位置中的输出位置。

可以将输入组名称映射到层次结构级别中的父节点。Developer 工具将更新层次结构中父节点的位置字段。Developer 工具不更新属于层次结构中的组的子节点。将输入端口映射到子节点时，每个输入端口位置都必须与位置父节点的位置相同。

将输入组映射到层次结构级别后，可以稍后进行更改。可以单击**清除**，也可以删除“端口”与“操作”区域之间的行。要删除行，请拖动行的指针以选择行。单击**删除**。

映射端口

将端口映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于端口要映射到的节点类型。

下表介绍了将单个端口映射到**操作区域**中的不同目标节点时产生的结果：

目标节点	结果
基本节点	将单个端口映射到某个节点，但父节点没有位置时，该节点将接收端口的位置。父节点位置将接收单个端口的输入组的位置。将单个端口映射到某个节点，并且父节点已有位置时，您可以更改父节点的位置并清除同一级别中其他子节点的位置。层次结构级别位置将更改为端口的组名称。
多次出现的基本节点或多次出现的基本节点的主键	将单个端口映射到多次出现的基本节点时，Developer 工具会将基本节点的位置设置为所选端口的组。
复杂节点	将单个端口映射到复杂节点时，Developer 工具会将复杂节点的位置设置为包含该端口的组的位置。Developer 工具将提示您输入要向其分配端口的单次出现的基本节点。如果单次出现的基本节点有位置，您将无法映射复杂节点。

映射组

将输入组映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于组要映射到的节点类型。

下表介绍了将组映射到**操作区域**中的某个节点时产生的结果：

目标节点	结果
基本节点	不能将组映射到基本节点。
多次出现的基本节点	系统将提示您选择输入组中的端口，以更新节点和主键的位置。
多次出现的复杂节点	Developer 工具会将复杂节点的位置设置为组的位置。

映射多个端口

将多个端口映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于端口要映射到的节点类型。如果从相同组进行映射，则可以同时映射多个端口。

下表介绍了将多个端口映射到节点时产生的节点结果：

目标节点	结果
单个基本节点	将多个端口映射到单个节点时，请在 操作区域 中更新多个基本节点的位置。如果层次结构在级别中要更新的节点不足，Developer 工具将仅对可用节点映射端口。

目标节点	结果
多次出现的基本节点	将多个端口映射到多次出现的基本节点时，端口将转换为多次出现的节点。Developer 工具将根据映射的端口数量创建节点的实例。此时将显示一条消息，指出计划的端口数量。
多次出现的复杂节点	将多个端口映射到一个复杂节点时，必须选择要更新的单次出现的基本节点。将端口转换为多次出现的节点。Developer 工具将根据映射的端口数量创建节点的实例。

转换多次出现的端口

可以将多个输入端口映射到 SOAP 消息中一个多次出现的节点。Developer 工具会将输入数据转换为 SOAP 消息中的多个节点。

要更改要转换的元素数量，请选择**映射选项**对话框中的**替换现有转换**。

如果您从**端口**区域中删除了转换的其中一个端口实例，Developer 工具将从**操作**区域中删除所有实例。

转换示例

输入组可能具有以下行：

序号	名称	位置	emp_name1	emp_name2	emp_name3
101	HR	纽约	Alice	Tom	Bob
102	产品	加利福尼亚	Carol	Tim	Dave

每个行都包含一个部门编号和三个员工姓名。

“员工”是 SOAP 消息层次结构中多次出现的节点。可以将“员工”的所有实例从输入行映射到 SOAP 消息层次结构。选择出现的所有员工。单击**映射**。**映射选项**对话框将提示您从列表选择一个节点。

Developer 工具将更改“员工”节点，以包含 SOAP 消息层次结构中的多个姓名节点：

```
Department
  num
  name
  location
  Employee (unbounded)
    emp_name1
    emp_name2
    emp_name3
```

SOAP 消息返回以下层次结构：

```
<department>
  <num>101</num>
  <name>HR</name>
  <location>New York</location>
  <employee>
    <emp_name>Alice</name>
  </employee>
  <employee>
    <emp_name>Tom</name>
  </employee>
  <employee>
    <emp_name>Bob</name>
  </employee>
</department>

<department>
  <num>102</num>
  <name>Product</name>
```



```
<location>California</location>
<employee>
  <emp_name>Carol</name>
</employee>
<employee>
  <emp_name>Tim</name>
</employee>
<employee>
  <emp_name>Dave</name>
</employee>
</department>
```

映射非规范化数据

您可以映射非规范化数据并将其传递到 SOAP 消息中的规范化节点。

映射非规范化数据时，您将数据从一个输入组传递到 SOAP 消息层次结构中的多个节点。您可以在 SOAP 消息中创建类似于以下类型的关系的组关系：

线性节点关系

节点 A 是节点 B 的父节点。节点 B 是节点 C 的父节点。节点 C 是节点 D 的父节点。

层次结构节点关系

节点 A 是节点 B 的父节点。节点 A 同时是节点 C 的父节点。节点 B 与节点 C 无关。

下表显示了包含非规范化分支机构和部门数据的输入行：

分支机构	Dept_Num	Dept_Name	电话	Employee_Num	Employee_Name
01	100	Accounting	3580	2110	Amir
01	100	Accounting	3580	2113	Robert
01	101	工程部	3582	2114	Stan
01	101	工程部	3582	2115	Jim
02	102	设施部	3583	2116	Jose

输入数据包含唯一的员工编号和姓名。对于相同部门和分支机构中的每名员工，分支机构和部门数据都会重复。

线性组关系

配置端口时，可以为分支机构、部门和员工分别配置一个组。分支结构是部门的父节点，部门是员工的父节点。您可以按照以下线性结构配置组：

```
Division
  Division_Key
  Division_Num
  Division Name

Department
  Department_Key
  Division_FKey
  Dept_Num
  Dept_Name
  Phone

Employee
  Department_Fkey
  Employee_Num
  Employee_Name
```

尽管 Division_Num 和 Dept_Num 在输入数据中重复出现，但 SOAP 消息中包含唯一的分支机构和部门实例。在“分支机构”组中定义 Division_Num 作为主键。在“部门”组中定义 Dept_Num 作为主键。

层次结构组关系

您可以创建一个组层次机构，其中包含“分支机构”父组以及“部门”和“员工”子组。“部门”与“员工”之间不存在主键-外键关系。“部门”和“员工”是“分支机构”的子组。您可以按照以下结构配置组：

```
Division
  Division_Key
  Division_Num
  Division_Name

Department
  Division_FKey
  Dept_Num
  Dept_Name

Employee
  Division_FKey
  Employee_Num
  Employee_Name
```

派生类型和元素置换

可以将输入端口映射到 SOAP 消息中的派生复杂类型、anyType 元素和置换组。SOAP 消息可以包含基本类型和派生类型的元素。

在类型关系中，基本类型是指派生了其他类型的类型。派生类型继承基本类型中的元素。扩展复杂类型是指继承了基本类型中的元素并包含其他元素的派生类型。受限复杂类型是指限制基本类型中的部分元素的派生类型。

生成派生类型

如果 WSDL 或架构中包括派生类型，您必须选择要包含在 SOAP 消息中的类型。

例如，WSDL 定义了一个基本类型 AddressType。WSDL 中还包含属于派生 AddressTypes 的 USAddressType 和 UKAddressType。

每种类型都包含以下元素：

- AddressType 返回一个行：street, city
- USAddressType（扩展 AddressType）：state, zipCode
- UKAddressType（扩展 AddressType）：postalCode, country

在“操作”区域中选择 USAddressType 时，Developer 工具将在 SOAP 消息中为 USAddressType 元素创建一个组。该组包括 USAddress 的基本地址中的街道和城市以及州和 zipCode。扩展了基本类型的派生类型始终包括基本类型中的元素。

如果您为 SOAP 消息选择了所有可用的派生类型，Developer 工具将在 SOAP 层次结构中创建类似于以下组的组：

```
Address
  Address: Address
    Street
    City

Address:USAddressType
  Street
  City
  State
```

```
ZipCode
Address: UKAddressType
  Street
  City
  PostalCode
  Country
```

您需要为 Address、USAddress 和 UKAddress 定义输入端口组。

生成 anyType 元素和属性

部分架构元素和属性允许在 SOAP 消息中使用任何类型的数据。

anytype 元素表示选择所有全局已知类型。将端口映射到 SOAP 消息中的 anyType 元素之前，请选择可用的复杂类型或 xs:string。如果 WSDL 或架构中不包含复杂类型，Developer 工具会将 anyType 元素类型替换为 xs:string。

要在操作区域中选择元素类型，请在 anyType 元素的**类型**列中单击**选择**。此时将显示一系列可用的复杂类型和 xs:string。

以下元素和属性允许使用任意类型的数据：

anyType 元素

允许关联 XML 文件中的元素属于任意数据类型。

anySimpleType 元素

允许关联 XML 文件中的元素属于任意 simpleType。

ANY 内容元素

允许元素属于在架构中定义的任意全局元素。

anyAttribute 属性

允许元素属于已在架构中定义的任意属性。

生成置换组

使用置换组将一个元素替换为 SOAP 消息中的另一个元素。置换组的工作方式与派生类型相似，不同的是元素定义中包含一个置换组名称。

例如，您可能有一个基本类型 Address 以及派生类型 USAddress 和 UKAddress：

```
xs:element name="Address" type="xs:string"/>
<xs:element name="USAddress" substitutionGroup="Address"/>
<xs:element name="UKAddress" substitutionGroup="Address"/>
```

配置 SOAP 消息层次结构时，可以为 SOAP 消息中的 Address 选择要替换的元素。

生成 SOAP 消息中的 XML 构造

WSDL 或架构可以包含选项、列表或联合元素。Web 服务转换可以生成包含这些元素的 SOAP 消息。

选项元素

选项元素将子元素限制为 <choice> 声明中的一个元素。

要将端口映射到包含选项元素的 SOAP 消息，请创建一个包含选项构造中的所有元素的输入组。例如，项目说明可以是维度或权重：

```
item: description, choice {dimension, weight}
```

如果说明是维度，则说明属于包含长度、宽度和高度的复杂类型。

如果说明是权重，则说明属于简单字符串类型。

输入数据具有以下列和行：

说明	长度	宽度	高度	权重
框	20cm	18cm	15cm	NULL
coffee	NULL	NULL	NULL	500g

SOAP 消息包含的项目组中包含维度或权重说明：

```
Item
  Description
    Dimension
      Length
      Width
      Height
    Weight
```

输入数据中的 NULL 值成为 XML 输出中缺失的元素。

SOAP 消息包含以下数据：

```
<item>
  <desc>box</desc>
  <dimension>
    <length>20cm</length>
    <width>18cm</width>
    <height>15cm</height>
  </dimension>
</item>

<item>
  <desc>coffee</desc>
  <weight>500g</weight>
</item>
```

列表元素

列表是指在相同元素或属性中包含多个简单类型值的 XML 元素。如果输入数据中的列表的表示形式为合并的数字字符串，则数据集成服务可以处理该列表。

如果该列表中的每个项目都是一个独立的元素，例如 ClassDates1、ClassDates2 和 ClassDates3，则数据集成服务将无法以列表方式处理这些项目。如果需要在 SOAP 消息中返回一个列表，可以使用表达式转换将这些项目合并成一个字符串。

以下输入行中包含一个名为 ClassDates 的列表元素，其中包含周中的某一天：

CourseID	名称	ClassDates
Math 1	Beginning Algebra	Mon Wed Fri
History 1	World History	Tue Thu

数据集成服务可以返回具有以下 XML 结构的 SOAP 消息：

```
<class>
  <courseId>Math 1</courseId>
```

```

        <name>Beginning Algebra</name>
        <classDates>Mon Wed Fri</classDates>
    </class>
    <class>
        <courseId>History 1</courseId>
        <name>World History</name>
        <classDates>Tue Thu</classDates>
    </class>

```

联合元素

联合元素是一种由多种类型联合组成的简单类型。SOAP 消息中包含联合元素时，必须映射包含字符串中的数据
的单个输入端口。

例如，SOAP 消息中包含一个名为 size 的元素。大小由整数和字符串组成：

```

<xs:element name="size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>

```

输入行中包含具有说明和大小的项目。项目可以具有数值大小，例如 42。或者，项目可以具有字符串值类型的大
小，例如 large、medium 或 small。

下表显示了具有数值大小和字符串大小的输入行：

说明	大小
shoes	42
shirt	large

为项目大小创建一个端口。以字符串形式映射端口。SOAP 消息中包含以下元素：

```

<item>
  <desc>shoes</desc>
  <size>42</size>
</item>

<item>
  <desc>shirt</desc>
  <size>large</size>
</item>

```

第 9 章

Web 服务使用者转换

本章包括以下主题：

- [Web 服务使用者转换概览, 78](#)
- [WSDL 选择, 80](#)
- [Web 服务使用者转换端口, 81](#)
- [Web 服务使用者转换输入映射, 82](#)
- [Web 服务使用者转换输出映射, 85](#)
- [Web 服务使用者转换高级属性, 87](#)
- [筛选器优化, 91](#)
- [创建 Web 服务使用者转换, 93](#)
- [Web 服务使用者转换示例, 94](#)

Web 服务使用者转换概览

Web 服务使用者转换作为 Web 服务客户端连接到 Web 服务以访问或转换数据。Web 服务使用者转换属于多组转换。

Web 服务使用开放标准，例如 SOAP、WSDL 和 XML。SOAP 是 Web 服务所使用的通信协议。Web 服务客户端请求和 Web 服务响应都属于 SOAP 消息。WSDL 是一种 XML 架构，用于描述 Web 服务操作的协议、格式和签名。

Web 服务操作包括对信息的请求、更新数据的请求或执行任务的请求。例如，Web 服务使用者转换发送一个 SOAP 请求，要求运行名为 `getCustomerOrders` 的 Web 服务操作。该转换会在请求中传递客户 ID。Web 服务将检索客户信息和订单信息。Web 服务会在 SOAP 响应中将信息返回给该转换。

Web 服务使用者转换使用在 WSDL、Web 服务连接或端点 URL 输入端口中定义的端点 URL 连接到 Web 服务。可以在 Web 服务连接中为 Web 服务启用安全性。

SOAP 消息

Web 服务使用者转换使用简单对象访问协议 (SOAP) 与 Web 服务提供程序交换信息并请求 Web 服务。SOAP 可定义 Web 服务请求和响应消息的格式。

使用 Web 服务使用者转换转换数据时，该转换将生成 SOAP 请求并连接到 Web 服务。该转换使用在 WSDL 对象、Web 服务连接或端点 URL 输入端口中定义的端点 URL 连接到 Web 服务。SOAP 请求包含 Web 服务运行请求操作所需的信息。Web 服务操作会在 SOAP 响应中将数据返回给转换。该转换会映射 SOAP 响应中的数据，然后在输出端口中返回数据。

Web 服务使用者转换对 ISO-8859-1 中的 SOAP 消息头进行编码。

该转换可以使用文档/文字编码处理 SOAP 消息。文档/文字样式需要一个 XML 架构来描述 SOAP 消息。SOAP 消息由 XML 构成。如果 SOAP 消息包含多次出现的元素，则元素组将形成 XML 层次结构中的级别。如果一个级别嵌套在另一个级别内，则这些组都相互关联。

SOAP 请求消息可以包含层次结构数据。例如，如果 Web 服务使用者转换向销售数据库发送一个添加客户订单的请求，则该转换会在 SOAP 请求消息中传递两组数据。一组包含客户 ID 和名称，另一组包含订单信息。订单信息将出现多次。

SOAP 响应消息可以包含层次结构数据。例如，如果 Web 服务使用者转换生成一条关于客户订单的 SOAP 请求，则 Web 服务将在 SOAP 响应中返回订单表头和多次出现的订单详细信息元素。

WSDL 文件

WSDL 文件包含对要传递给 Web 服务的数据的说明，以便发件人和收件人了解要交换的数据。必须将 WSDL 文件导入存储库，才能创建 Web 服务使用者转换。

WSDL 描述了对数据执行的操作以及对协议或传输的绑定，以便 Web 服务使用者可以按照正确的格式发送请求消息。WSDL 描述了连接到 Web 服务的网络地址。

WSDL 包含有关如何对 SOAP 请求消息和响应消息进行编码的信息。SOAP 编码方式决定 SOAP 消息正文的格式。它描述了 Web 服务用于和 Web 服务使用者进行通信的请求消息和响应消息的格式。Web 服务开发人员可以使用各种工具包创建 Web 服务。工具包支持不同方式的 SOAP 编码消息。

Web 服务使用者转换支持文档/文字 SOAP 编码样式。可以对 Web 服务使用者转换使用 WSDL 1.1。不能使用 WSDL 附件，例如 MIME、DIME 和 MTOM 消息。

操作

对于自身支持的每个行为，Web 服务都包含一项操作。

例如，Web 服务可以包含名为 getcustomerid 的操作，用于接收客户名称并使用客户详细信息进行响应。操作输入包含客户名称的元素。操作输出包含基于客户名称的客户详细信息的元素。

配置 Web 服务使用者转换时，请定义转换如何将数据映射到操作输入，以及转换如何从操作输出映射数据。在转换中配置以下信息：

输入映射

定义如何将转换输入端口映射到 Web 服务操作输入节点。操作输入为操作定义了 SOAP 请求中的元素。

输出映射

定义如何将 Web 服务操作输出节点映射到转换输出端口。操作输出为操作定义了 SOAP 响应中的元素。

Web 服务安全

可以在 Web 服务连接中为 Web 服务启用安全性。可以配置以下安全类型：

Web 服务安全

数据集成服务向 Web 服务提供程序发送 SOAP 请求时，可以包含 Web 服务安全表头。Web 服务安全表头包含身份验证信息，因此 Web 服务提供程序可以对数据集成服务进行身份验证。

Web 服务使用者转换提供用户名标志。数据集成服务可在 SOAP 请求中创建单独的安全 SOAP 表头，然后将该请求传递给 Web 服务提供程序。

可以在 Web 服务连接中使用以下类型的 Web 服务安全：

- PasswordText。数据集成服务不更改 WS-Security SOAP 表头中的密码。

- PasswordDigest。数据集成服务将密码与临时值和时间戳结合。数据集成服务对密码应用 SHA 哈希，采用 base64 编码方式进行编码，并在 SOAP 表头中使用编码后的密码。

传输层安全

使用安全套接字层 (SSL) 实现 TCP/IP 的传输层 (TCP 层) 上的安全性。Web 服务将 SSL 上的超文本传输协议 (HTTPS) 用作安全消息传输的 Web 地址。Web 服务使用者转换可以使用 TLS 1.2、TLS 1.1 或 TLS 1.0。可以将以下身份验证与传输层安全结合使用：HTTP 身份验证、代理服务器身份验证和 SSL 证书。

SSL 身份验证

通过 HTTPS 协议进行连接时，可以使用 SSL 身份验证。

可以使用以下类型的 SSL 身份验证：

- 单向 SSL 身份验证
- 双向 SSL 身份验证

HTTP 身份验证

通过 HTTP 协议进行连接时，可以使用 HTTP 身份验证。

可以使用以下 HTTP 身份验证方法：

- 基本身份验证
- 摘要身份验证
- NT LAN Manager (NTLM) 身份验证

WSDL 选择

创建 Web 服务使用者转换之前，必须将 WSDL 文件导入模型存储库中。WSDL 定义要运行的 Web 服务的操作签名。导入 WSDL 时，Developer tool 将创建一个物理数据对象，您可以将其重用于其他转换。

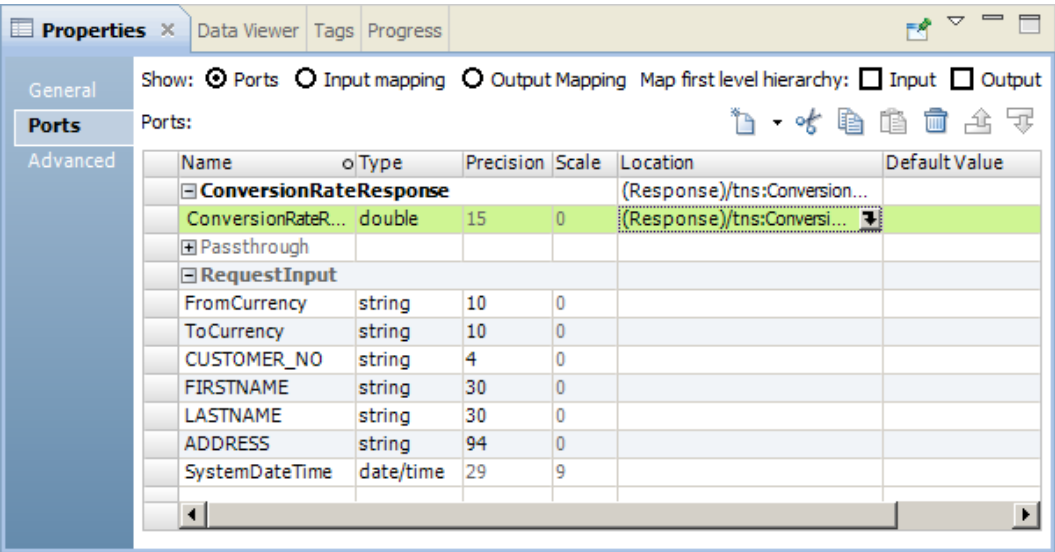
一个 WSDL 可以定义多个操作。创建 Web 服务使用者转换时，需要选择要运行的操作。可以查看 Web 服务使用者转换中操作输入和操作输出的层次结构。这些层次结构将定义 SOAP 请求消息和 SOAP 响应消息的结构。

还可以通过单向输入操作导入 WSDL。在通过单向输入操作导入 WSDL 时，必须创建虚拟输出端口。

Web 服务使用者转换端口

查看转换端口时，如果不需要查看操作层次结构，则可以显示端口。显示端口时，可以定义组，定义端口，以及将操作输出中的节点映射到输出端口。

下图显示了一个不可重用的 Web 服务使用者转换的端口：



一个 Web 服务使用者转换可以包含多个输入组和多个输出组。创建端口后，可以创建组，然后将这些端口添加到组中。请基于操作输入层次结构或操作输出层次结构的结构在组层次结构中定义端口。通过添加键，可以将子组与父组关联。在层次结构中，除了最下面的组外，其他所有组都必须具有主键。在层次结构中，除了根组以外，其他所有组都必须具有外键。

该转换包含一个名为 RequestInput 的根输入组。必须在这个根输入组中添加主键。主键必须为 String、Bigint 或 Integer 类型。

可以向该根输入组添加其他传递端口。这些传递端口会将数据不加修改地传递给转换。传递端口可以在输入数据中出现一次。可以向任何输出组添加传递端口。将输出端口关联到输入端口。传递给 SOAP 请求的输入值会在 SOAP 响应的输出行中重复出现。

还可以向根输入组添加 HTTP 表头、Cookie 端口、动态 URL 端口和用于 Web 服务安全身份验证的端口。根组中的数据只出现一次。

要将操作输出节点映射到输出端口，请单击位置列中的字段并展开选择位置对话框中的层次结构。然后，从层次结构选择一个节点。

HTTP 表头输入端口

Web 服务可能需要其他 HTTP 表头。可以在根输入组中创建输入端口，以将其他表头信息传递到 Web 服务提供程序。

要添加 HTTP 表头和 HTTP 端口，请选择根输入组，然后单击新建按钮旁边的箭头。然后，单击 HTTP 表头。输入表头名称和端口名称。

可以创建多个 HTTP 表头。

其他输入端口

可以将预定义的输入端口添加到 Web 服务使用者转换。

可以添加以下的预定义输入端口：

Cookie 端口

可以配置 Web 服务使用者转换以使用 Cookie 身份验证。远程 Web 服务服务器会根据 cookie 跟踪 Web 服务使用者用户。某个映射多次调用一个 Web 服务时，可以提高性能。

将 cookie 端口映射到 Web 服务请求消息时，Web 服务提供程序将在响应消息中返回 cookie 值。您可以将该 cookie 值传递到映射中的其他转换下游，或者也可以将该 cookie 值保存在文件中。将 cookie 值保存在文件中时，可以将该 cookie 配置为 Web 服务使用者转换的输入。

您可以将 cookie 输出端口传递到任意 Web 服务使用者转换输出组。

端点 URL 端口

Web 服务使用者转换使用端点 URL 连接到 Web 服务。可以在 WSDL 文件、Web 服务连接或端点 URL 输入端口中定义端点 URL。当转换动态接收端口中的 URL 后，数据集成服务将替代在 WSDL 文件或 Web 服务连接中定义的 URL。

对于每个 Web 服务请求，Web 服务使用者转换都可以有一个 URL 端口值。将端点 URL 端口添加到根输入组。

WS-Security 端口

在 Web 服务连接中启用 Web 服务安全。启用 Web 服务安全时，必须在 Web 服务连接或 WS-Security 输入端口中定义用户名和密码。

添加 WS-Security 端口时，可以将用户名和密码传递到转换中的输入端口。当转换动态接收端口中的用户名和密码后，数据集成服务将替代在 Web 服务连接中定义的值。

注意：Web 服务连接有一个用于 HTTP 和 WS-Security 身份验证的用户名和密码。

要添加预定义的输入端口，请在**端口**区域中单击根输入组。单击**新建按钮**旁边的箭头，然后单击**其他端口**。选择要添加的端口。

Web 服务使用者转换输入映射

查看转换端口时，系统将显示输入映射，从中可以查看操作输入层次结构。显示输入映射时，可以定义输入组，定义输入端口，以及将输入端口映射到操作输入节点。

输入映射包含以下区域：

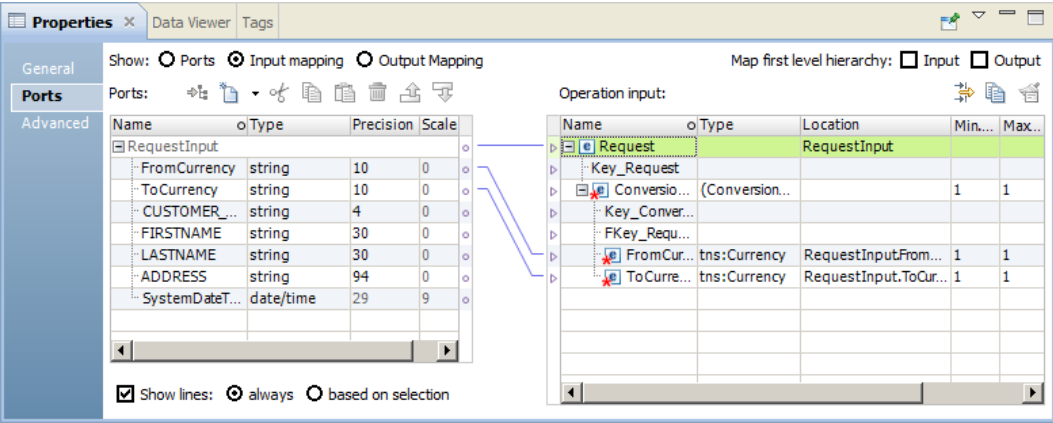
端口

可在**端口**区域中创建转换输入组和输入端口。

操作输入

操作输入区域显示了 Web 服务使用者转换发送到 Web 服务的 SOAP 请求消息中的节点。用于创建转换的 WSDL 数据对象定义了操作输入层次结构。

下图显示了一个不可重用的 Web 服务使用者转换的输入映射：



创建输入端口后，请将**端口**区域中的输入端口映射到**操作输入**区域中的节点。将输入端口映射到操作输入中的节点后，该端口的位置将显示在**操作输入**区域的**位置**列中。

如果选择映射操作输入层次结构的第一级，则 Developer 工具会将操作输入第一级中的节点映射到输入端口。开发程序工具还会创建端口以执行映射。如果层次结构的第一级包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer 工具不会映射该层次结构的第一级。

可以将一个字符串或文本输入端口中的 XML 数据映射到整个 SOAP 请求消息。如果将 XML 数据映射到整个 SOAP 请求，则无法将端口映射到操作输入中的节点。

可以选择查看将输入端口与操作输入中的节点连接在一起的线。

将输入端口映射到节点的规则和准则

将输入端口映射到操作输入层次结构中的节点时，请遵循以下规则：


- 可以将一个输入端口映射到层次结构中的一个节点。可以将同一端口映射到层次结构中任意数量的键。
- 输入端口和节点的数据类型必须兼容。
- 可以将一个输入组中的端口映射到操作输入中的多个层次结构级别。
- 必须将输入端口映射到操作输入中的键。映射到键的任何端口都必须为 String、Integer 或 Bigint 数据类型。将数据映射到操作输入中位于 SOAP 消息所含层次结构级别之上的所有级别中的键。对于映射级别及之上的所有级别，需要包括外键。

注意：如果只打算映射操作输入层次结构的最低级别，则无需将输入端口映射到键。

- 可以将多个 String、Bigint 或 Integer 输入端口映射到**操作输入**区域中的键以创建复合键。为复合键单击**位置**字段时，可以重新对输入端口进行排序，或者删除其中一个端口。

自定义视图选项

您可以更改操作输入层次结构以便在**操作输入**区域中显示键。此外，还可以显示分组构造，用于定义节点的排序方式。

单击**自定义视图**按钮 ()，该按钮位于**操作输入**区域中。启用以下任意选项：

“序列”、“选项”和“全部”

显示一行，指示元素定义是“序列”、“选项”还是“全部”。

“全部”组中的节点都必须包含在 SOAP 消息中。

- “序列”组中的节点必须符合 WSDL 中指定的顺序。
- “选项”组中必须至少有一个节点显示在 SOAP 消息中。

键

在**操作输出**区域中查看键。 **操作输入**区域包含每个组的键。可以在**端口**区域中将键添加到输入端口。

将输入端口映射到操作输入

显示转换输入映射时，您可以定义输入组、定义输入端口并将输入端口映射到操作输入节点。

1. 打开 Web 服务使用者转换。
2. 要显示转换输入映射，请使用以下方法之一：
 - 对于可重用转换，请单击**概览**视图。选择显示输入映射。
 - 对于不可重用转换，请单击**属性**视图中的**端口**选项卡。选择显示输入映射。
3. 为根输入组定义主键。
4. 要将输入组或端口添加到**端口**区域，请使用以下方法之一：

选项	描述
拖动节点	将 操作输入 区域中的组节点或子节点拖动到 端口 区域的空列中。如果该节点是组节点，则 Developer 工具将添加组，而不添加端口。
手动添加组或端口	要添加组，请单击 新建 按钮旁边的箭头，然后单击 组 。要添加端口，请单击 新建 按钮旁边的箭头，然后单击 字段 。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到 Web 服务使用者转换。
复制端口	从其他转换中选择端口，并将其复制到 端口 区域。要复制端口，可以使用键盘快捷键或 Developer 工具中的复制和粘贴按钮。
选择映射第一级层次结构	选择 映射第一级层次结构 。Developer 工具会将第一级操作输入中的节点映射到输入端口和组。Developer 工具还会创建输入端口和组以执行映射。

5. 如果手动创建端口或从其他转换中复制端口，请单击**操作输入**区域中的**位置**列，然后从列表中选择端口。
6. 要将输入端口映射为复合键，请使用以下方法之一：

选项	描述
拖动输入端口	选择两个或多个输入端口，然后将它们拖动到操作输入层次结构的键中。
从 选择位置 对话框中选择输入端口	在操作输入层次结构中单击键的 位置 列，然后选择输入端口。

7. 要清除节点的位置，请使用以下方法之一：

选项	描述
单击 清除	在 操作输入 区域中选择一个或多个节点，然后单击 清除 。
删除将端口连接到节点的行	在操作输入中选择将输入端口连接到节点的一个或多个行，然后按 删除 。

- 8. 如果关联的 WSDL 数据对象包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输入**区域中选择对象。在节点的类型列中，单击**选择**，然后从列表选择一个或多个类型、元素或属性。
- 9. 要将字符串或文本输入端口中的 XML 数据映射到完整的 SOAP 请求，请在端口中右键单击，然后选择**映射为 XML**。

Web 服务使用者转换输出映射

查看转换端口时，系统将显示输出映射，从中可以查看操作输出层次结构。显示输出映射时，可以定义输出组，定义输出端口，以及将操作输出节点映射到输出端口。

输出映射包含以下区域：

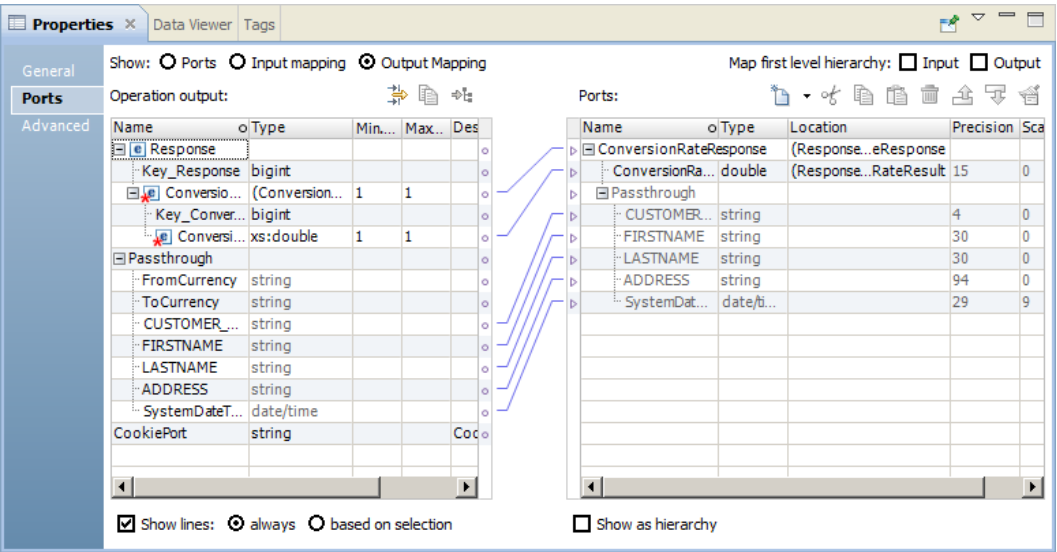
操作输出

操作输出区域显示了 Web 服务返回给 Web 服务使用者转换的 SOAP 响应消息中的节点。用于创建转换的 WSDL 数据对象定义了操作输出层次结构。

端口

可在**端口**区域中创建转换输出组和端口。

下图显示了一个不可重用的 Web 服务使用者转换的输出映射：



创建输出端口后，将**操作输出**区域中的节点映射到**端口**区域中的端口。将操作输出中的节点映射到输出端口时，节点的位置将显示在**端口**区域的**位置**列中。

选择映射操作输出层次结构的第一级时，Developer 工具会将其中的节点映射到输出端口。开发程序工具还会创建端口以执行映射。如果层次结构的第一级包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer 工具不会映射该层次结构的第一级。

可以选择在层次结构中显示输出端口。每个子组都显示在父组下方。还可以选择查看将操作输出中的节点连接到输出端口的行。

如果从存储库中删除了关联的 WSDL 数据对象，则 Developer 工具会将操作节点的位置保留在输出映射中。显示输出映射时，**端口**区域仍将在输出端口的**位置**列中显示操作节点的位置。如果对转换关联了其他 WSDL，则

Developer 工具会检查每个位置是否有效。如果操作节点位置不再有效，则 Developer 工具将在输出映射的**端口**区域中清除该位置。

将节点映射到输出端口的规则和准则

将操作输出层次结构中的节点映射到输出端口时，请查看以下规则：

- 操作输出节点和输出端口必须使用兼容的数据类型。
- 不能将一个节点映射到组中的多个输出端口。
- 除了传递端口，每个输出端口都必须具有有效位置。
- 如果要将多次出现的子节点拖动到空的输出端口，则必须将该组关联到其他输出组。选择组后，Developer 工具将创建用于关联组的键。
- 将多次出现的元素拖动到包含父元素的组时，可以配置要包含的子元素的出现次数。也可以将父组替换为转换输出中多次出现的子组。

将 SOAP 消息映射为 XML

可以将完整的 SOAP 消息映射为 XML，而不是将数据返回到不同的输出端口。

将 SOAP 消息映射为 XML 后，数据集成服务会在一个端口中返回完整的 SOAP 消息。不要创建输出端口。

要映射完整的消息，请右键单击**操作输出**区域中的根组。选择**映射为 XML**。

Developer 工具将创建一个字符串输出端口。精度为 65535 个字节。

自定义视图选项

可以通过更改操作输出层次结构，在**操作输出**区域中显示 Cookie 端口、传递端口和键。此外，还可以显示分组构造，用于定义节点的排序方式。

单击**自定义视图**按钮，该按钮位于**操作输出**区域中。启用以下任意选项：

“Sequence”、“Choice”和“all”

显示一行，指示元素定义是“sequence”、“choice”还是“all”。

“all”组中的节点都必须包含在 SOAP 消息中。

“sequence”组中的节点必须符合 WSDL 中指定的顺序。

“choice”组中必须至少有一个节点显示在 SOAP 消息中。

键

在**操作输出**区域中查看键。**操作输出**区域包含每个组的键。可以在**端口**区域中将键添加到输出端口。

传递端口

操作输出区域显示传递端口。传递端口是指在转换中传递数据而不更改数据的端口。可以将操作输出中的传递端口映射到任何 Web 服务使用者转换输出组。传递端口会一次性接收数据，因此，该端口处于 SOAP 消息中的根级别。

Cookie 端口

显示 Cookie 端口。如果配置 Cookie 身份验证，则远程 Web 服务服务器将根据 Cookie 跟踪 Web 服务使用者用户。如果映射请求消息中的 Web 服务 Cookie，则 Web 服务将在响应消息中返回 Cookie。可以将操作输出中的 Cookie 映射到任何 Web 服务使用者转换输出组。

将操作输出映射到输出端口

显示转换输出映射时，您可以定义输出组、定义输出端口并将操作输出节点映射到输出端口。

- 1. 打开 Web 服务使用者转换。
- 2. 要显示转换输出映射，请使用以下方法之一：
 - 对于可重用转换，请单击**概览**视图。选择显示输出映射。
 - 对于不可重用转换，请单击**属性**视图中的**端口**选项卡。选择显示输出映射。
- 3. 要将输出组或端口添加到**端口**区域，请使用以下方法之一：

选项	描述
拖动节点	将 操作输出 区域中的组节点或子节点拖动到 端口 区域的空列中。如果该节点是组节点，则 Developer 工具将添加组，而不添加端口。
手动添加组或端口	要添加组，请单击 新建 按钮旁边的箭头，然后单击 组 。要添加端口，请单击 新建 按钮旁边的箭头，然后单击 字段 。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到 Web 服务使用者转换。
复制端口	从其他转换中选择端口，并将其复制到 端口 区域。要复制端口，可以使用键盘快捷键或 Developer 工具中的复制和粘贴按钮。
选择映射第一级层次结构	选择 映射第一级层次结构 。Developer 工具会将第一级操作输出中的节点映射到输出端口和组。Developer 工具还会创建输出端口和组以执行映射。

- 4. 如果手动创建端口或从其他转换中复制端口，请单击**端口**区域中的**位置**列，然后从列表中选择节点。
- 5. 要清除端口的**位置**，请使用以下方法之一：

选项	描述
单击 清除	在 端口 区域中选择一个或多个端口，然后单击 清除 。
删除将节点连接到端口的行	在操作输出中选择将节点连接到输出端口的一个或多个行，然后按 删除 。

- 6. 如果关联的 WSDL 数据对象包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输出**区域中选择对象。在节点的**类型**列中，单击**选择**，然后从列表选择一个或多个类型、元素或属性。
- 7. 要将完整的 SOAP 响应消息映射为 XML，请在**操作输出**区域中右键单击根组，然后选择**映射为 XML**。

Web 服务使用者转换高级属性

Web 服务使用者转换高级属性包括跟踪级别、通用故障端口、Web 服务连接和并发 Web 服务请求消息。

可以在“高级”选项卡中为 Web 服务使用者转换定义以下高级属性：

跟踪级别

此转换的日志中显示的详细信息量。可以选择精简、普通、详细初始化或详细数据。默认值为“普通”。

SOAP 操作

对于 Web 服务使用者转换，需使用常量值替代 WSDL 中定义的 SOAP 操作值。

启用通用 SOAP 故障处理

返回 WSDL 中未定义的故障消息。在 GenericFault 输出组中创建输出端口以处理故障代码和消息。

下表介绍了 SOAP 1.1 和 SOAP 1.2 的故障输出端口：

SOAP 1.1 的故障输出端口	SOAP 1.2 的故障输出端口	说明
Fault Code	Code*	返回故障标识代码。
Fault String	Reason*	在故障消息中返回错误说明。
Fault Detail	详细端口	在通用故障消息中返回 Web 服务提供程序传递给 Web 服务使用者转换的自定义信息。
Fault Actor	Role	返回导致故障发生的对象的相关信息。
-	Node	返回生成故障的 SOAP 节点的 URI。
* Code 和 Reason 输出端口是分层次的。		

注意：可以展开 Code 故障输出端口，以便将 SubCode 故障输出端口提升一个级别。

启用 HTTP 错误处理

返回来自 Web 服务的任何 HTTP 错误。在 GenericFault 输出组中创建 HTTP 错误输出端口。

将故障视为错误

将故障消息添加到映射日志。出现故障时，数据集成服务将增加映射的错误计数。禁用此属性才能允许早期选择优化和推入优化。默认情况下启用该属性。

连接

标识连接到 Web 服务的 Web 服务连接对象。在 Developer 工具中创建 Web 服务连接。在 Developer 工具或 Administrator 工具中编辑 Web 服务连接。配置 Web 服务连接时，请配置端点 URL、Web 服务所需的安全类型和连接超时时限。

Web 服务使用者转换使用端点 URL 连接到 Web 服务。可以在 WSDL 文件、Web 服务连接或端点 URL 输入端口中定义端点 URL。

使用以下准则确定何时配置 Web 服务连接：

- 如果使用的端点 URL 与 WSDL 文件中的 URL 不同，且打算使用端点 URL 输入端口，请配置一个该连接。
- 如果连接到的 Web 服务需要 Web 服务安全、HTTP 身份验证或 SSL 证书，请配置一个该连接。
- 如果要更改默认的连接超时时限，请配置一个该连接。

注意：可以将存储库中的 WSDL 数据对象与 Web 服务连接关联。关联的连接会成为从该 WSDL 创建的每个 Web 服务使用者转换的默认连接。

启用压缩

使用 GZIP 压缩方法启用 SOAP 请求编码，然后使用 GZIP 或 deflate 启用 SOAP 响应解码。

XML 架构验证

在运行时验证 SOAP 响应消息。选择 **XML 无效错误或不验证**。

已排序输入

允许数据集成服务在未处理所有输入数据的情况下生成输出。当输入数据按操作输入层次结构中的键排序时，请启用已排序输入。

推入优化

启用推入优化。单击**推入优化**属性中的**打开**按钮以选择用于接收筛选值的筛选器端口。对于每个筛选器端口，请选择在 Web 服务响应中包含已筛选列的输出端口。

有副作用

该复选框指示 Web 服务除返回行之外还执行其他功能。如果 Web 服务除返回行之外还修改对象或与其他对象或功能进行交互，则 Web 服务使用者转换会产生副作用。该 Web 服务可能会修改数据库、计入总数、引发异常、编写电子邮件或调用其他具有副作用的 Web 服务。要允许推入优化或早期选择优化，请禁用**有副作用**属性。默认情况下启用该属性。

启用并发

允许 Web 服务使用者转换创建多个到同一 Web 服务的并发连接，从而并行发送多个 Web 服务请求。允许 Web 服务使用者转换创建与 Web 服务的多个并发连接时，可以设置总内存消耗限制和并发连接限制数。

下表介绍了这些选项：

选项	说明
启用并发	创建到同一 Web 服务的多个并发连接。
并发连接限制	并发 Web 服务连接数。默认值为 20。
总并发内存限制(以 MB 为单位)	所有并发连接的总内存分配限制。默认值为 100 MB。

Web 服务错误处理

可以将 Web 服务使用者转换配置为将 SOAP 故障和 HTTP 错误传递给映射中的下游。出现故障时，可以增加错误计数。可以在转换高级属性中配置 Web 服务错误处理方式。

Web 服务返回响应消息或故障。故障属于错误。Web 服务可能会基于出现的错误生成不同的故障。

Web 服务使用者转换可以返回以下类型的故障：

SOAP 故障

WSDL 所定义的 SOAP 故障。配置可在 Web 服务响应消息中返回故障的输出错误端口。对于 SOAP 1.1 绑定，数据集成服务将返回该故障的故障消息、故障代码、故障字符串以及故障触发者等元素。对于 SOAP 1.2 绑定，数据集成服务将返回该故障的故障消息、代码、原因、节点和角色等元素。

通用 SOAP 故障

Web 服务将在运行时生成通用 SOAP 故障。SOAP 1.1 绑定和 SOAP 1.2 绑定的故障元素不同。WSDL 未定义通用 SOAP 故障。通用 SOAP 故障包括身份验证失败和 SOAP 请求错误。

HTTP 错误

如果在转换中启用了 HTTP 错误处理，则 Developer 工具将添加 HTTP 故障输出端口。数据集成服务将通过一个字符串端口返回来自 Web 服务的 HTTP 错误。每个 HTTP 错误都包含一个错误代码和一条消息。

如果来自 Web 服务的 SOAP 响应包含无效的 XML 数据，则 Web 服务使用者转换将返回错误。

可以配置是否将 SOAP 故障视为错误。如果启用了“将故障视为错误”，并且发生 SOAP 故障，则数据集成服务会使映射中的错误计数递增。该故障将显示在消息日志中。

消息压缩

如果启用了 SOAP 消息压缩，则 Web 服务使用者转换会压缩 Web 服务请求消息，并接收经过压缩的 Web 服务响应消息。

Web 服务使用者转换将使用 GZip 压缩对 SOAP 请求进行编码。该转换将接受使用 GZip 或 deflate 压缩进行编码的响应消息。

数据集成服务接收来自 Web 服务的响应时，将检查 SOAP 消息中的内容编码 HTTP 表头并对该消息进行解码。

默认设置为无压缩编码。Web 服务不会压缩 SOAP 响应。

下表显示了在打开或关闭压缩时请求消息和响应消息的表头：

压缩	表头
打开	Content-Encoding 表头：GZip Accept-Encoding 表头：GZip、deflate
关闭	空的 Content-Encoding 表头 空的 Accept-Encoding 表头

有时，Web 服务会使用默认压缩对响应消息进行编码。如果该消息是通过 GZip 或 deflate 编码的，则 Web 服务使用者转换可对该消息进行解码。如果 Web 服务是以意外的方式对响应消息进行编码的，则 Web 服务使用者转换会将该消息记录到映射日志中。

可在转换高级属性中启用压缩。

并发

您可以启用 Web 服务使用者转换来创建指向 Web 服务的多个并发连接，以便同时发送多个 Web 服务请求。

例如，在查询银行信息时，可将 Web 服务使用者转换配置为并发，以便同时发送多行。如果存在 20 个输入行，则可以同时发送 20 个请求，从而加快处理速度。

在 Web 服务使用者转换中启用并发后，可配置总内存消耗限制。

在 Web 服务使用者转换中启用并发后，可配置并发的 Web 服务连接数。

针对并发的规则和准则

使用并发时，请使用以下规则和准则：

- 并发支持将已排序的输入行作为与 Web 服务的多个并发连接：不支持已排序的输出行。
- 如果数据集超过 100 行，请使用并发。
- 建议不要增加并发 Web 服务连接的数量。并发 Web 服务连接的数量将链接到操作系统使用的套接字数量。增加套接字数量的成本较高。
- 使用并发功能时，请使用具有多内核处理器并具有至少 100 MB RAM 的系统，以实现最佳性能。
- 并发内存限制表示调用 Web 服务时并发工作流使用的内存。
- 启用 Web 服务使用者转换中的并发时，可以配置内存使用限制。请确保内存使用不超过服务器上的物理 RAM。

并发的最佳实践

要在使用并发时获得最佳性能，请遵循以下最佳实践：

- 避免更改总并发内存限制的默认值和并发连接限制的默认值。
- 避免对少于 100 行的数据集使用并发。
- 使用并发时，避免在映射中使用传递端口。

筛选器优化

筛选器优化通过减少传递映射的行数来提高性能。数据集成服务可以应用早期选择优化或推入优化。

数据集成服务应用筛选器优化方法时，会将筛选器移至尽可能靠近映射中的源的位置。如果数据集成服务无法将筛选器移至映射中的转换之前，则可以将筛选器逻辑推入转换中。

启用通过 Web 服务使用者转换执行早期选择优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用早期选择优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。

通过 Web 服务使用者转换执行推入优化

在 SQL 数据服务中，当 Web 服务使用者转换位于虚拟表中时，您可以通过该转换配置推入优化。

映射调用 Web 服务以根据最终用户 SQL 查询中的语句检索数据集或数据子集。最终用户 SQL 查询中包含一个可选筛选条件。

通过推入优化，Web 服务使用者转换在筛选器端口中接收筛选器值。筛选器端口是指未连接的输入端口，您在配置推入优化时将该端口标识为筛选器端口。筛选器端口具有默认值，该值可确保最终用户查询中不包含筛选器时 Web 服务返回所有行。筛选器端口不是传递端口。

注意：筛选器字段必须属于 Web 服务请求中根组的一部分。

配置筛选器端口时，应将 Web 服务使用者转换中负责从 Web 服务响应中接收列数据的输出端口标识为输出端口。例如，如果筛选器端口是指名为 EmployeeID 的输入端口，响应中的输出端口可能是名为 EmployeeNum 的端口。Developer 工具需要与输入筛选器端口和输出端口关联，以便将筛选器逻辑从读取的虚拟表推送到 Web 服务使用者请求。Web 服务请求的输入端口通常与 Web 服务响应中的输出端口不同。

筛选器字段不能是传递端口。配置筛选器端口时，端口的默认值更改为筛选条件的值，因而传递输出端口值会发生变化。基于输出传递端口的筛选器返回意外结果。

您可以将多个筛选器表达式推送到 Web 服务使用者转换。每个筛选条件的格式都必须如下所示：

```
<Field> = <Constant>
```

筛选条件必须通过 AND 联接。不能通过 OR 联接这些条件。

通过 Web 服务使用者转换执行推入优化示例

SQL 数据服务返回所有客户的订单，或者根据收到的来自用户的 SQL 查询返回特定客户的订单。

数据服务中包含具有以下组件的逻辑数据对象：

客户表

包含客户信息的 Oracle 数据库表。

Web 服务使用者转换

调用 Web 服务的转换可检索客户的最新订单。Web 服务使用者转换包含 customerID 和 orderNum 对应的输入端口。该转换具有包含接收的来自客户表的客户数据的传递端口。orderNum 端口为筛选器端口，处于未连接状态。orderNum 的默认值为 “*”。Web 服务在 Web 服务请求中收到此值后，将返回所有订单。

订单虚拟表

从 Web 服务接收客户和订单数据的虚拟表。最终用户会查询此虚拟表。订单中包含客户列、订单 ID 列以及客户和订单数据。

最终用户将以下 SQL 查询传递到 SQL 数据服务：

```
SELECT * from OrdersID where customer = 23 and orderID = 56
```

数据集成服务将拆分该查询以优化映射。数据集成服务使用早期选择优化并将筛选器逻辑 customer = 23 移至读取的客户表。数据集成服务使用推入优化并将筛选器逻辑 orderID = 56 推送到 Web 服务使用者转换筛选器端口。Web 服务使用者转换检索客户 23 的订单 ID 56。

启用通过 Web 服务使用者转换执行推入优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用推入优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。
4. 单击**推入优化**属性中的**打开**按钮。
5. 在“已优化输入”对话框中选择筛选器端口名称。
您可以选择多个筛选器端口。
6. 单击**输出列**。
7. 对于每个筛选器端口，请选择在 Web 服务响应中包含已筛选列的输出端口。
8. 输入每个筛选器端口的默认值。

注意：不能为 Web 服务使用者端口配置默认值，除非该端口为筛选器端口。

创建 Web 服务使用者转换

可以创建可重用或不可重用的 Web 服务使用者转换。可重用转换可存在于多个映射中。不可重用转换存在于单个映射内。

可以从单个 WSDL 对象创建 Web 服务使用者转换以用于 SOAP 1.1 绑定和 SOAP 1.2 绑定。

1. 要创建转换，请使用以下方法之一：

选项	说明
可重用	在对象浏览器视图中选择一个项目或文件夹。单击文件 > 新建 > 转换。选择 Web 服务使用者转换，然后单击下一步。
不可重用	在映射或 Mapplet 中，将“转换”选项板中的 Web 服务使用者转换拖动到编辑器中。

此时将显示新建 Web 服务使用者转换对话框。

2. 浏览并选择 WSDL 数据对象以定义 Web 服务请求和响应消息。

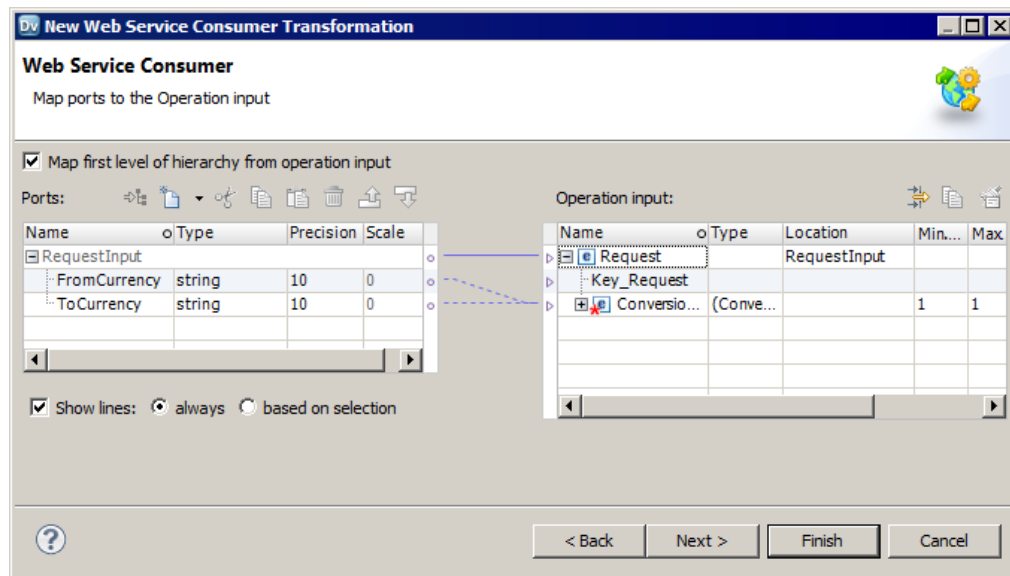
如果 WSDL 不在存储库中，可以从“新建 Web 服务使用者转换”对话框中导入 WSDL。

3. 浏览并从 WSDL 中选择一项操作。

可以选择包含 SOAP 1.1 绑定或 SOAP 1.2 绑定的操作。

4. 单击下一步。

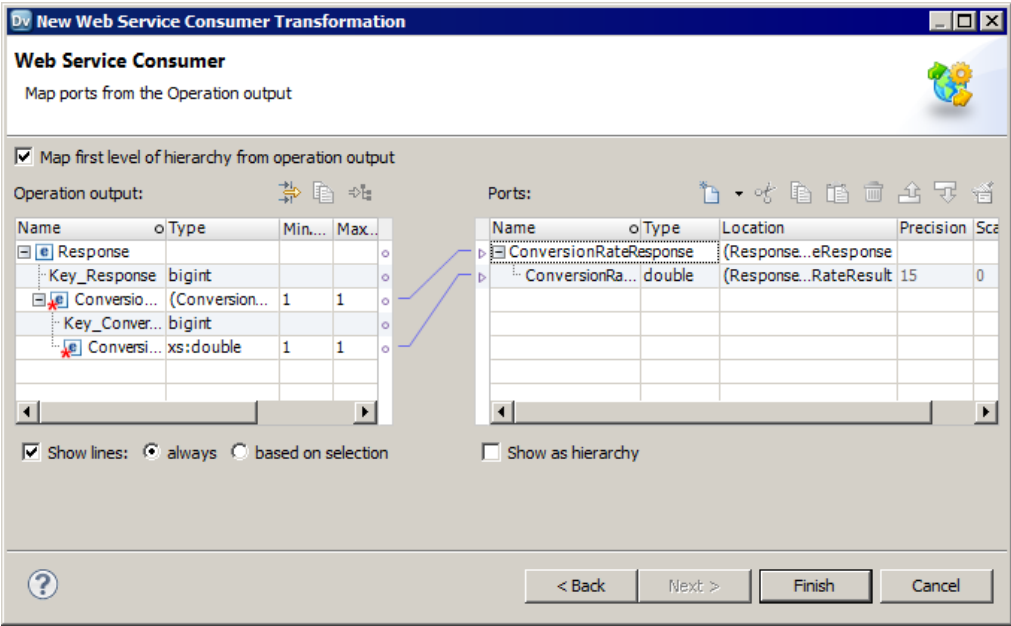
此时将显示将端口映射到操作输入屏幕。端口区域显示转换输入组和输入端口。操作输入区域显示请求消息层次结构。



5. 定义输入组和输入端口，并将输入端口映射到操作输入节点。

6. 单击下一步。

此时将显示从操作输出中映射端口屏幕。操作输出区域显示响应消息层次结构。端口区域显示转换输出组和输出端口。



7. 定义输出组和输出端口，并将操作输出节点映射到后者。
8. 单击**完成**。
9. 单击**高级**视图以配置转换属性和 Web 服务连接。

Web 服务使用者转换示例

您的组织需要向销售组织公开 RT100 产品线的订单信息。销售团队需要每日查询订单摘要和订单详细信息。

创建一个逻辑数据对象，该对象可公开虚拟表中的每日订单信息。读取映射包含一个 Web 服务使用者转换，该转换可返回最新的 RT100 订单。该 Web 服务使用者转换将使用一个 Web 服务返回 RT100 产品线的每日订单摘要和订单详细信息。

输入文件

输入文件是包含产品线编号的平面文件。

请创建物理数据对象以定义输入文件。该文件包含一个字段，即 Product_Line。字段值为 RT100。在**运行时属性**视图中定义物理数据对象的位置。

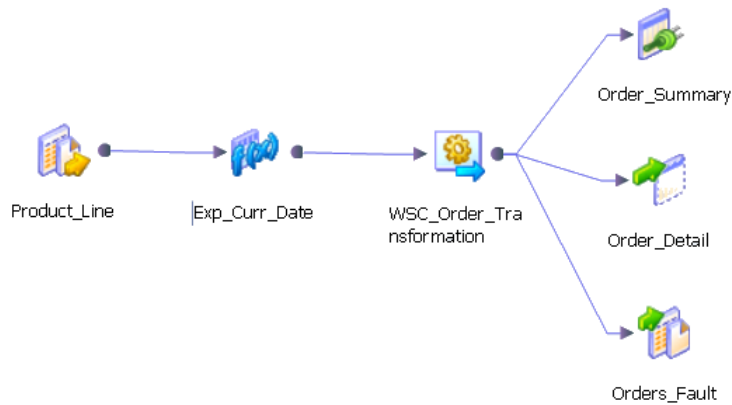
逻辑数据对象模型

组织中的业务分析将创建用于描述顺序摘要和顺序详细表结构的逻辑数据模型。逻辑数据模型包含 Order_Summary 和 Order_Detail 逻辑数据对象。

该分析通过定义逻辑数据模型的建模工具创建架构。您可以从架构中导入逻辑数据模型，并创建 Order_Summary 和 Order_Detail 逻辑数据对象。

逻辑数据对象映射

逻辑数据对象映射介绍如何通过逻辑数据对象访问数据。



读取映射包含以下对象：

Product_Line

包含产品线编号的输入平面文件。

Exp_Curr_Date 转换

用于返回当前日期和 Web 服务使用者转换根级别输入组主键的表达式转换。

WSC_Order 转换

使用 Web 服务检索顺序信息的 Web 服务使用者转换。该转换会将产品线和当前日期传递给请求消息中的 Web 服务。该转换将通过响应消息中的 Web 服务接收顺序信息。

Order_Summary 表

包含顺序信息的逻辑数据对象，例如 Order_No、Customer_Id、Qty 和 Order_Date。

Order_Detail 表

包含详细顺序信息的逻辑数据对象，例如 Order_No、Customer_Id、Qty 和 Status。

Orders_Fault

用于接收通用故障消息的输出平面文件。

Web 服务使用者转换

Web 服务使用者转换会将产品线、日期和序列号作为输入来接收。转换将使用 Get_Order_Info Web 服务操作以检索订单信息。

创建 Web 服务使用者转换时，请选择描述请求并响应 Web 服务消息的 WSDL 数据对象。Web 服务消息包含 XML 元素的层次结构组。一个元素可能包含其他元素。某些元素可能出现多次。从存储库中的 Order_Info WSDL 对象创建转换。

配置转换输入端口并将这些端口映射到操作输入层次结构。将操作输出层次结构的节点映射到输出端口。定义 Web 服务连接和运行时属性。

转换输入映射

在端口视图中显示输入映射时，您可以定义输入端口并将其映射到操作输入中的节点。

转换端口区域具有一个根组和一个订单组。根组为请求输入组。将一个端口添加到请求输入组来表示主键。

订单组具有 **Select_Date** 和 **Select_Product_Line** 输入端口。

将输入端口映射到**操作输入**区域的 **Order_Date** 和 **Product_Line** 节点。

操作输入区域定义 Web 服务使用者转换传递到 Web 服务的请求消息。默认情况下，这些节点将出现在**操作输入**区域。

转换输出映射

在**端口**视图上显示输出映射时，您可以通过将操作输出的节点映射到转换输出组来定义输出端口。

Web 服务将在 Web 服务响应消息中返回以下层次结构：

```
Response
  Orders
    Order
      Key_Order
      Order_ID
      Order_Date
      Customer_ID
      Total_Qty
      Order_Details
        Order_Detail
          Product_ID
          Description
          Qty
          Status
```

Web 服务返回多个订单。Order 为 Orders 级别中多次出现的节点。对于每个订单，Web 服务可以返回多个订单明细。Order_Detail 为 Order_Details 级别中多次出现的节点。

注意：Developer 工具将 Key_Order 节点添加至用户界面中。您可以将键映射到输出组以定义组之间的关系。例如，Order_ID 是 Order 中的主键，同时也是 Order_Details 中的外键。

在**端口**区域创建以下输出组：

```
Order
  Order_ID
  Order_Date
  Customer_ID
  Total_Qty

Order_Detail
  Order_ID
  Product_ID
  Description
  Qty
  Status
```

每当 Order_ID 的值更改时，数据集成服务就会从 Order 组写入一行。

每当 Order_ID 和 Product_ID 的值更改时，数据集成服务就会从 Order_Detail 组写入一行。

转换高级属性

请为 Web 服务使用者转换配置以下高级属性：

启用通用 SOAP 故障处理

添加用于接收 SOAP 故障消息的输出端口。

连接

选择一个 Web 服务连接以访问 Web 服务。

启用压缩

Web 服务使用者转换将使用 GZIP 压缩 Web 消息。

第 10 章

REST Web 服务

本章包括以下主题：

- [REST Web 服务概览, 97](#)
- [REST Web 服务过程, 98](#)
- [Web 服务使用者转换流程, 98](#)
- [REST Web 服务资源, 98](#)
- [资源映射, 101](#)
- [REST Web 服务输出转换, 103](#)
- [请求消息, 104](#)
- [响应消息格式, 106](#)
- [响应数据预览, 107](#)

REST Web 服务概览

Informatica REST Web 服务可处理 HTTP 数据请求，并在 JSON 文件或 XML 文件中返回响应。

外部应用程序、Web 浏览器或 REST Web 服务使用者转换可以连接到 REST Web 服务并发送请求。REST Web 服务会处理请求并将响应发送回到客户端。

例如，Web 服务客户端发送了运行 Web 服务操作的请求。该 Web 服务客户端在请求中传递了一个客户 ID。Web 服务将从 Orders 表检索客户信息和订单信息。Web 服务会在 JSON 文件中将信息返回到客户端。

可以通过在 Developer tool 中定义数据服务来创建 Informatica REST Web 服务。还可以将数据对象部署为 REST Web 服务。

Informatica REST Web 服务包含以下组件：

资源

资源包含 REST Web 服务将运行的映射和 Web 服务将返回的响应消息的定义。资源还包含资源 ID，这是输出数据中的键字段。可以从数据对象创建资源，也可以手动定义资源。当您将对对象部署为 REST Web 服务时，Developer tool 将创建资源。一个 Web 服务可以包含多个资源。

请求消息

Web 服务客户端向 Web 服务发送的要求执行某个任务的请求。Informatica Web 服务可以执行 HTTP GET 方法。请求消息是一个字符串，其中包含 Web 服务的名称、用于执行任务的资源的名称和网络位置以及用于筛选输出的参数。

资源映射

将数据返回到 Web 服务客户端的映射。可以创建默认资源映射或自定义资源映射。默认资源映射包含具有相同端口的一个读取转换和一个输出转换。除了读取转换和输出转换之外，自定义资源映射可能还包含其他转换。自定义资源映射所包含的读取转换可能具有与输出转换不同的端口。

响应消息

一个 JSON 或 XML 文件，其中包含要返回到 Web 服务客户端的数据。响应消息可以包含元素的层次结构和多次出现的数据。

REST Web 服务过程

REST Web 服务会处理来自 Web 服务客户端（例如外部应用程序、Web 浏览器或 REST Web 服务使用者转换）的请求。

以下过程描述了数据集成服务如何处理来自 Web 服务客户端的 Web 服务请求：

1. 数据集成服务接收来自 Web 服务客户端的请求。
2. 数据集成服务的 REST Web 服务模块通过运行资源映射以从数据对象检索行来处理该请求。
3. 如果请求包含筛选器参数，数据集成服务将筛选输出行。
4. REST Web 服务模块向 Web 服务客户端发送响应消息。

Web 服务使用者转换流程

外部应用程序或 Web 服务使用者转换可作为 Web 服务客户端连接到 Web 服务。

以下过程描述 Web 服务使用者转换如何发送请求以及接收来自 Web 服务的响应：

1. Web 服务使用者转换生成请求，并通过连接对象连接到 Web 服务。
2. Web 服务使用者转换收到来自 Web 服务的响应。
3. Web 服务使用者转换从该响应提取数据，并在转换输出端口中返回数据。

REST Web 服务资源

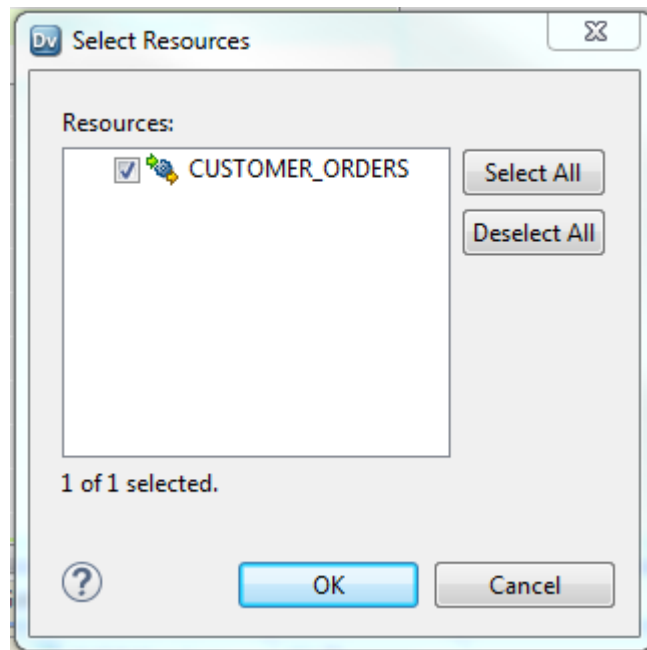
资源包含 REST 响应消息的定义和用于返回响应的数据访问方法。创建 Informatica REST Web 服务时，可以手动定义资源，也可以从数据对象创建资源。

可以从关系数据对象或平面文件数据对象创建资源。从数据对象创建资源时，Developer tool 将创建包含一个读取转换和一个输出转换的默认资源映射。输出转换包含与读取转换相同的端口。

可以手动定义输出响应消息的结构。手动创建资源时，您需要定义响应消息元素。然后定义将数据返回到响应消息的映射。

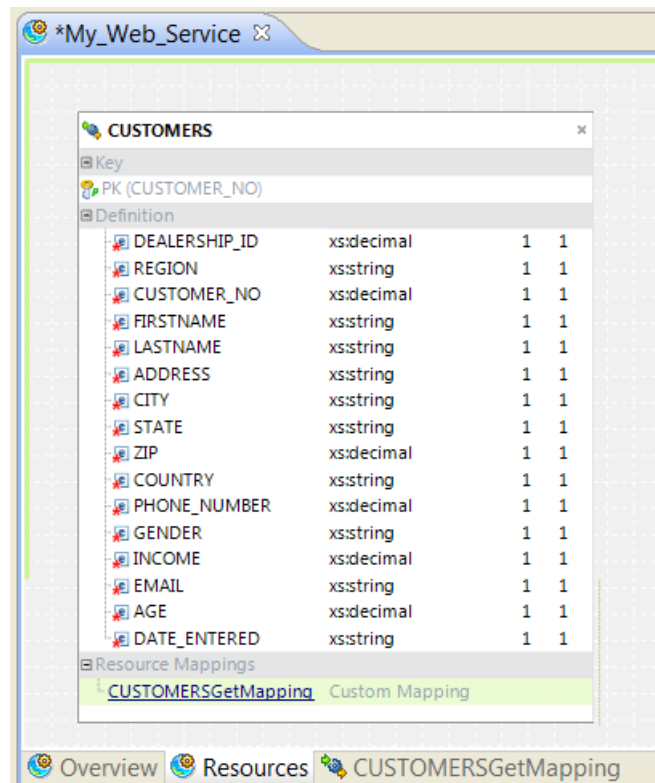
一个 REST Web 服务可以包含多个资源。每个资源都包含用于检索数据的资源映射和输出响应的定义。可以选择要在“资源”选项卡中查看的资源。在编辑器中右键单击，然后选择**显示资源**。

下图显示了“选择资源”对话框：



可以在资源选项卡中查看资源的组件。

下图显示了 REST Web 服务的资源选项卡中某个资源的组件：



该资源包含以下组件：

键

响应消息中的数据的索引。Web 服务客户端可以请求特定键的数据。可以将任何简单类型元素设置为输出中的键。

定义

输出响应消息中的元素。可以通过展开资源定义或导航到资源属性的架构视图来查看元素。

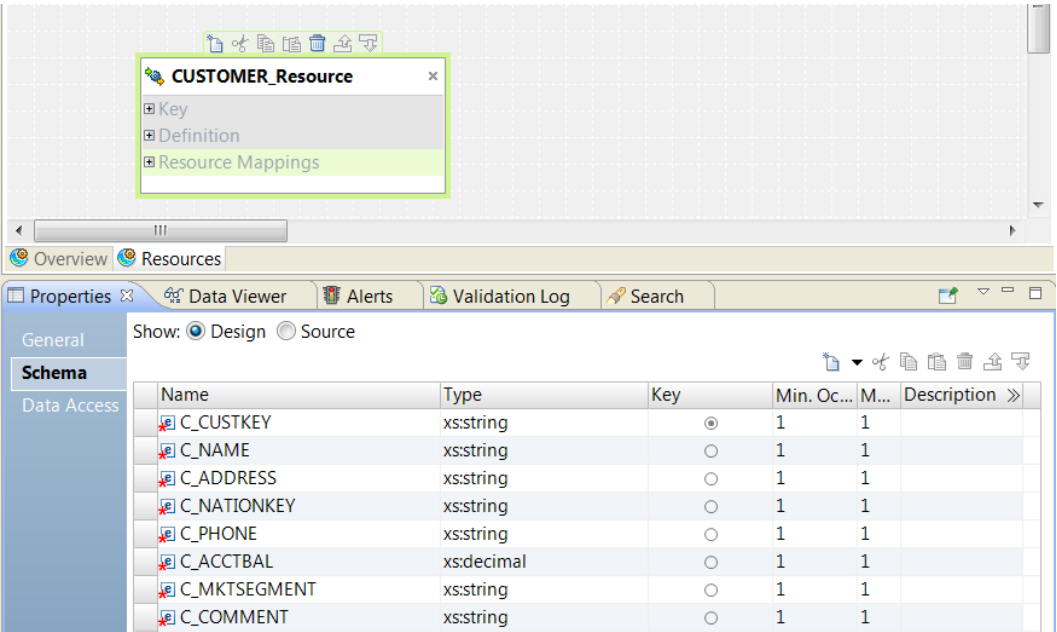
资源映射

执行以下操作的映射：检索数据，转换数据以及在响应消息中返回数据。默认情况下，资源映射包含一个读取转换和一个输出转换。可以向映射添加任何转换。可以通过单击资源中的链接来查看资源映射。

REST Web 服务架构视图

在资源属性的架构视图中查看或修改 REST 资源定义的结构。

下图显示了“架构”视图：



可以选择架构的设计或源视图。选择源视图可查看 XML 格式的架构。

使用设计视图可更改架构中的元素或元素类型。可以更改元素的顺序。可以在设计视图中更改键。还可以定义多次出现的字段。请输入字段必须出现的最小次数和最大次数。可以选择无限大以创建可出现无限次的多次出现的字段。

更改架构的格式时，所做更改会显示在输出转换中。您可能需要更新输出转换中的端口。

数据对象同步

当资源包含默认映射时，Developer tool 可以将资源定义与数据对象同步。

当资源键与数据对象中的主键不同时，Developer tool 无法将资源定义与数据对象同步。

资源键

可以在资源定义中定义键。定义键时，数据集成服务将为输出数据中的列编制索引。Web 服务客户端可按键请求特定输出行。

从数据对象创建资源时，Developer tool 在默认情况下使用该数据对象中的主键作为资源键。如果主键包含多个列，Developer tool 会用“+”分隔列数据来创建主键。

可以对 Developer tool 进行配置，使其使用与源的主键不同的输出列作为主键。

手动创建资源时，资源键必须为简单类型的元素。

资源映射

资源映射是读取数据对象并将数据返回到 REST 响应消息的映射。Informatica 资源映射执行 GET 方法。映射不会解析 Web 服务请求消息。

资源映射不包含用于筛选客户端请求的数据的筛选器转换或查找转换。如果 Web 服务请求消息包含筛选器查询，数据集成服务会在映射检索数据之后对数据进行筛选。在 Administrator 工具中为数据集成服务启用了“结果集缓存”属性时，数据集成服务会在响应消息中缓存数据。如果 Web 服务请求包含筛选器参数，数据集成服务会筛选缓存中的行。

定义资源映射时，将创建以下类型之一的资源映射：

默认资源映射

默认资源映射包含一个读取转换和一个输出转换。该映射不包含其他转换。映射将按原样返回数据对象中的所有行。要创建默认资源映射，请从数据对象创建 Web 服务，或者将数据对象部署为 Web 服务。

自定义资源映射

自定义资源映射是默认映射以外的任何其他映射。自定义资源映射所包含的读取转换可能具有与输出转换不同的列。在读取转换和输出转换之间，自定义资源映射可能还包含其他转换。转换可以添加列或更改管道中的列。

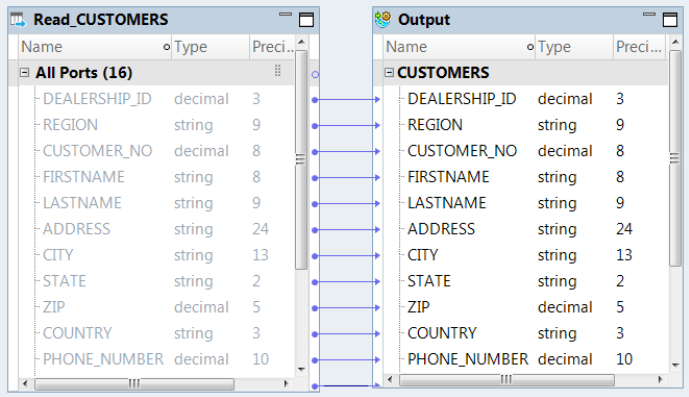
例如，资源映射可能包含将合并两个列并返回第三个列的表达式转换。或者，资源映射可能包含将返回多次出现的行的转换。手动创建资源时，可以创建自定义资源映射。还可以通过更改读取转换或向映射添加更多转换来更改默认映射。

默认资源映射

默认资源映射包含一个读取转换和一个输出转换。从数据对象创建 Web 服务并且未进行更改时，将创建默认资源映射。

从数据对象创建资源时，向导将创建包含一个读取转换和一个输出转换的默认资源映射。读取转换将读取已从中创建了资源的数据对象。在默认资源映射中，输出转换包含与读取转换相同的列。

下图显示了将从 Customers 数据对象返回所有列的默认资源映射：



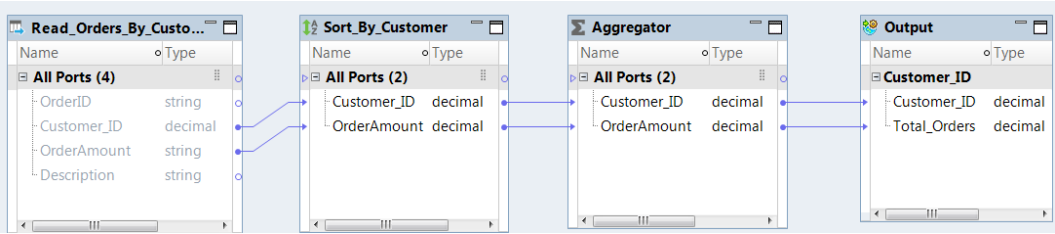
该资源映射不包含用于筛选数据的筛选器转换或查找转换。但是，REST Web 服务客户端可以向具有默认资源映射的 Web 服务发送带有筛选器参数的请求。数据集成服务将筛选资源映射生成的输出数据中的数据。

自定义资源映射

自定义资源映射所包含的读取转换可以具有与输出转换不同的端口。除了读取转换和输出转换之外，自定义资源映射还可以包含其他转换。

可以通过修改默认资源映射或手动创建 REST Web 服务来创建自定义资源映射。

下图显示了一个自定义资源映射：



该映射包含以下转换：

读取转换

读取订单文件。每个订单都包含客户 ID。客户 ID 可以多次出现。

排序器转换

按客户 ID 对订单进行排序。

汇总器转换

计算每个客户的订单总额。

输出转换

按客户返回订单总额。

要创建此自定义资源映射，请手动定义 REST Web 服务资源定义。手动定义资源定义时，您需要定义响应消息元素。在此示例中，响应消息仅包含客户 ID 和订单总额。

定义资源定义之后，Developer tool 将创建包含一个输出转换的资源映射。随后便可以向该映射添加读取转换和其他转换。

上图中的客户 ID 是键。Web 服务客户端可能会请求特定客户的订单数量。数据集成服务将按该键筛选输出数据。该映射不包含筛选器转换。

REST Web 服务输出转换

输出转换将从操作映射中的关系数据组创建 REST Web 服务响应消息。当您定义 REST Web 服务时，Developer tool 将创建输出转换。

当您创建 REST Web 服务时，Developer tool 将基于您已定义的资源定义创建输出转换。Developer tool 将基于响应消息结构为转换创建输入端口。该输出转换包含将输入端口映射到响应消息中的节点的映射。

可以通过更改架构来更改输出消息层次结构中的元素。可以更改转换中的输入端口以与架构更改保持一致。

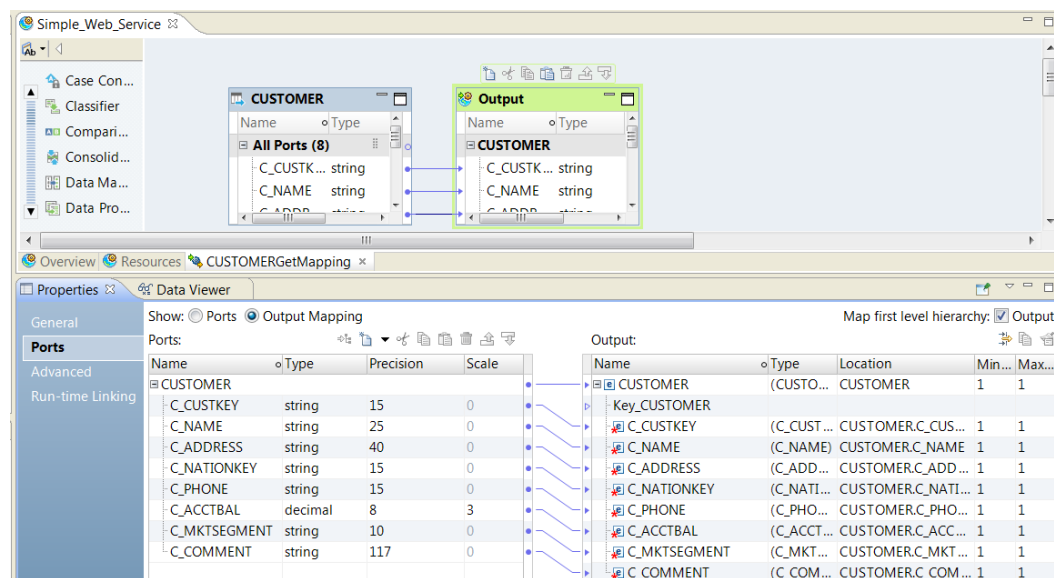
可以在输出转换的端口选项卡上查看 REST 响应消息层次结构。可以选择查看转换端口或查看转换输入端口和响应消息层次结构之间的映射。

显示端口时，可以手动添加组和端口，或将端口从其他转换复制到输出转换。可以使用键盘快捷键或 Developer tool 中的复制和粘贴按钮。

显示输出映射时，可以定义输入组，定义输入端口，以及将输入端口映射到响应消息元素。该选项卡的左侧是端口区域，右侧是输出区域。输出区域显示响应消息层次结构。可以在端口区域中定义输入组和输入端口。将端口区域中的输入端口映射到操作输出区域中的节点时，输入端口位置会显示在输出区域中的位置列中。

显示输出映射时，可以选择显示输入端口与操作输入中的节点的连接线条。

下图显示了输出转换中的简单输出映射：



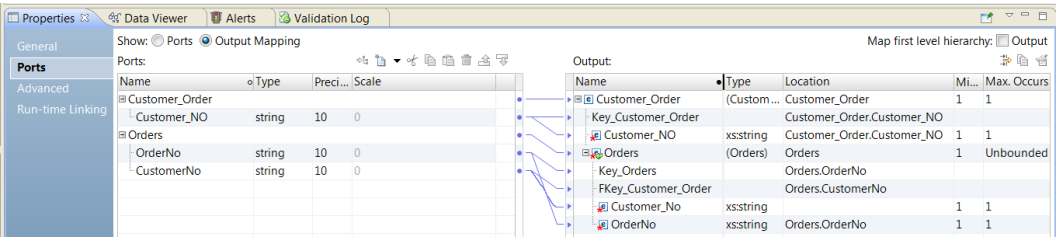
在上图中，转换将接收一组代表客户的端口。它将返回仅出现一次的元素。

如果选择映射第一级层次结构，则 Developer tool 会将输入端口映射到第一级操作输出中的节点。默认情况下，**映射第一级层次结构**选项处于启用状态。Developer tool 同时会创建映射数据所需的输入端口。如果第一次层次结构包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer tool 不会创建端口或映射第一级层次结构。

REST 输出转换中多次出现的数据

REST 输出转换可以接收多个数据组，并在响应消息中返回输出数据的多个级别。响应消息中的每个数据组由键进行链接。

下图显示了返回客户的多个订单的输出映射：



输出映射的端口区域包含两个组：Customer_Order 和 Orders。每个组都包含一个客户编号。客户编号将成为用于在响应消息中将客户数据和订单数据链接起来的键。

资源定义中的主键是 Customer_No。Customer_No 是 Customer_Order 中的一个键。它是 Orders 中的外键。主键必须包含唯一值。在处理多次出现的订单时，数据集成服务会基于该键返回一个客户的所有订单。主键和外键不会显示在响应消息中。

输出映射的输出区域包含响应消息。必须将输入端口链接到输出中的每个键。

注意：当 REST 输出转换收到重复的主键值时，预览将失败并显示错误。

请求消息

Web 服务客户端使用 URI 字符串向 Web 服务发送请求消息。该 URI 标识了主机端口、Web 服务名称和要访问的 Web 服务资源。请求消息可以包含查询参数，以筛选 Web 服务输出中的特定行。

如果 URI 不包含标识符或参数，REST Web 服务将在 REST Web 服务响应中返回数据对象中所有行的列表。Web 服务将返回包含所有客户行的 JSON 文件或 XML 文件。

以下示例显示了将返回所有客户的列表的请求：

http://myhost:8095/DataIntegrationService/RESTSERVICE/Rsrc_CUSTOMER/CUSTOMER

以下文本显示了 JSON 响应消息：

```
{
  "CUSTOMERS": {
    "CUSTOMER": [
      {
        "C_ACCTBAL": 9331.13,
        "C_ADDRESS": "38 Summit Drive",
        "C_COMMENT": "Call immediately if delay",
        "C_CUSTKEY": {
          "@url": "http://myhost:8095/DataIntegrationService/RESTSERVICE/Rsrc_CUSTOMER/CUSTOMER/CUSTOMER/63",
          "$": 63
        },
        "C_MKTSEGMENT": "AUTOMOBILE",
        "C_NAME": "Customer#000000063",
        "C_NATIONKEY": 21,
        "C_PHONE": "31-952-552-9584"
      },
      {
        "C_ACCTBAL": -646.64,
        "C_ADDRESS": "44 Ocean Avenue",
        "C_COMMENT": "Has dangerous animal in the house",
        "C_CUSTKEY": {
          "@url": "http://myhost:8095/DataIntegrationService/RESTSERVICE/Rsrc_CUSTOMER/CUSTOMER/CUSTOMER/64",
          "$": 64
        }
      }
    ]
  }
}
```



```

\0002FRsrc_CUSTOMER\0002FCUSTOMER\0002F64",
    "$": 64
  },
  "C_MKTSEGMENT": "BUILDING",
  "C_NAME": "Customer#000000064",
  "C_NATIONKEY": 3,
  "C_PHONE": "13-558-731-7204"
}
}}

```

筛选资源映射中的数据

当请求消息包含查询参数时，数据集成服务会筛选具有查询参数值的输出数据。

数据集成服务可以筛选默认资源映射和自定义资源映射中的输出行。

可以在请求消息查询中包含以下条件：

```
<> = != >= <=
```

要在 URI 中配置参数，请包含以下子句：

```
?filtercondition=<列名称> <操作数><值>
```

例如，以下请求消息将搜索具有“BUILDING”市场细分值的所有客户行：

```

http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/?
filterCondition=C_MKTSEGMENT='BUILDING'

```

如果列数据类型为字符串，请使用单引号 (') 将搜索值括起来。如果列数据类型为数值，请勿使用引号将搜索值括起来。

例如，以下请求包含数据类型为数值的搜索参数：

```

http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/?
filterCondition=C_ACCTBAL=9331.13

```

REST Web 服务将返回帐户余额为 9331.13 的客户。

以下文本以 XML 格式显示了 REST 响应消息：

```

<CUSTOMERs>
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">
<tns:C_CUSTKEY url="http://myServer:8095/DataIntegrationService/RestService/REST_Web_Service/CUSTOMER/
63">63</tns:C_CUSTKEY>
<tns:C_NAME>Customer#000000063</tns:C_NAME>
<tns:C_ADDRESS>IXRSpVwWZraKII</tns:C_ADDRESS>
<tns:C_NATIONKEY>21</tns:C_NATIONKEY>
<tns:C_PHONE>31-952-552-9584</tns:C_PHONE>
<tns:C_ACCTBAL>9331.13</tns:C_ACCTBAL>
<tns:C_MKTSEGMENT>AUTOMOBILE</tns:C_MKTSEGMENT>
<tns:C_COMMENT>Apply discount</tns:C_COMMENT>
</tns:CUSTOMER>
</CUSTOMERs>

```

如果要在查询中包含多个参数，请使用“AND”联接这些参数。

```

http://usw1mj02ee4j:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service0/CUSTOMER/?
filterCondition=C_ACCTBAL=9331.13 AND C_NATIONKEY='21'

```

如果列数据类型为“日期/时间”，您可以将 URI 中的参数字符串转换为“日期/时间”格式并指定参数字符串的格式。

例如，以下语句可将参数字符串转换为“日期/时间”格式：

```
?filterCondition= O_ORDERDATE=TO_DATE('1994-11-17 00:00:00.000000000','YYYY-MM-DD HH24:MI:SS.NS')
```

按键搜索

可以基于资源键配置 REST Web 服务查询。当请求包含键值时，您不必在查询中引用列名称。

使用以下格式可在名为 CUSTOMER 的资源的资源键中搜索编号 64：

```
http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/64
```

该 URI 包含资源名称和资源键值：

```
/CUSTOMER/64
```

查询不需要引用作为键的列的名称。

按资源键进行查询时，Web 服务可以使用与该键关联的 URI 来检索客户。可以在 URI 的响应消息键值中查看特定行。在以下示例中，键为 C_CUSTKEY，键值为 64：

```
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">
  <tns:C_CUSTKEY url="http://my Server:8095/DataIntegrationService/RestService/REST_Web_Service/CUSTOMER/64">64</tns:C_CUSTKEY>
  <tns:C_NAME>Customer#000000064</tns:C_NAME>
  <tns:C_ADDRESS>MbCeGY20kaKK3oalJD,OT</tns:C_ADDRESS>
  <tns:C_NATIONKEY>3</tns:C_NATIONKEY>
  <tns:C_PHONE>13-558-731-7204</tns:C_PHONE>
  <tns:C_ACCTBAL>-646.64</tns:C_ACCTBAL>
  <tns:C_MKTSEGMENT>BUILDING</tns:C_MKTSEGMENT>
  <tns:C_COMMENT>
    Customer has an angry dog in the yard
  </tns:C_COMMENT>
</tns:CUSTOMER>
```

响应消息格式

可以对 REST Web 服务进行配置，使其以 JSON 或 XML 格式返回响应消息。

在响应消息中，C_CUSTKEY 字段是资源键。C_CUSTKEY 字段包含键值和 URL，应用程序可以使用它们来按特定键访问输出数据中的客户。

在 JSON 格式的 URL 中，反斜杠使用 unicode 编码为 \u002f。

URL 中的资源键值可能还包含编码值，原因如下：

- 键值包含 URL 地址，例如 http://www.informatica.com。
- 键值包含百分号 (%)，例如 20%。
- 键值包含空格。

以下文本是 JSON 格式的响应消息：

```
{
  "CUSTOMERS": {
    "CUSTOMER": [
      {
        "C_ACCTBAL": 9331.13,
        "C_ADDRESS": "IXRSpVWwZraKII",
        "C_COMMENT": "Good customer",
        "C_CUSTKEY": {
          "@url": "http:\u002F\u002FHostName:8095\u002FDataIntegrationService\u002FSimple_Web_Service\u002FRsrc_CUSTOMER\u002FCUSTOMER\u002F63",
          "$": 63
        },
        "C_MKTSEGMENT": "AUTOMOBILE",
        "C_NAME": "Customer#000000063",
        "C_NATIONKEY": 21,
        "C_PHONE": "31-952-552-9584"
      }
    ]
  }
}
```

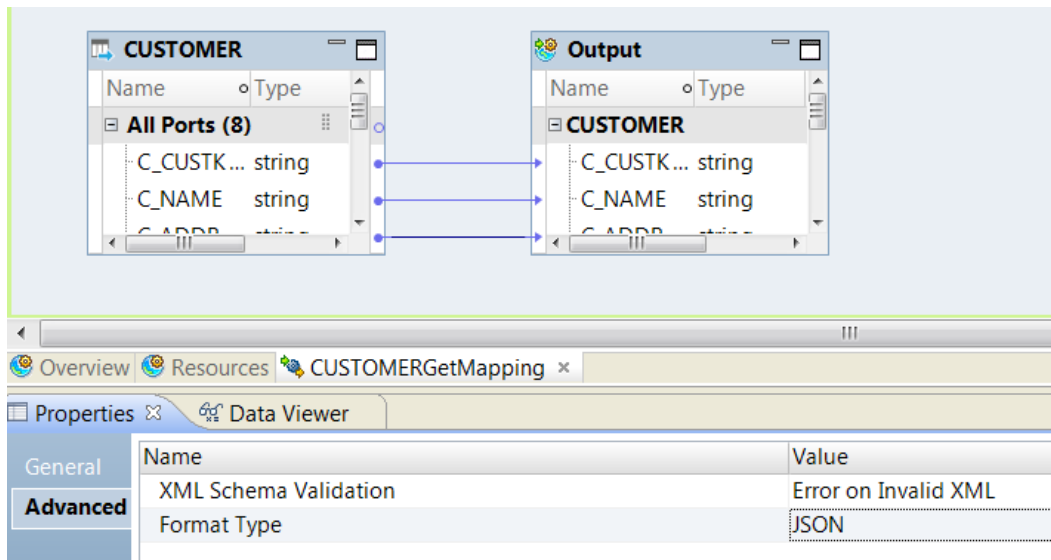
```
    }
  }
}
```

以下文本是 XML 格式的响应消息：

```
<?xml version="1.0" encoding="UTF-16LE" standalone="no"?>
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">
  <tns:C_CUSTKEY url="Simple_Web_Service/CUSTOMER/63">63</tns:C_CUSTKEY>
  <tns:C_NAME>Customer#000000063</tns:C_NAME>
  <tns:C_ADDRESS>IXRSpVWwZraKII</tns:C_ADDRESS>
  <tns:C_NATIONKEY>21</tns:C_NATIONKEY>
  <tns:C_PHONE>31-952-552-9584</tns:C_PHONE>
  <tns:C_ACCTBAL>9331.13</tns:C_ACCTBAL>
  <tns:C_MKTSEGMENT>AUTOMOBILE</tns:C_MKTSEGMENT>
  <tns:C_COMMENT>Good
customer</tns:C_COMMENT>
</tns:CUSTOMER>
```

可以设置响应消息的格式。可以在资源映射的**属性视图的高级选项卡**上更改响应消息格式。为 **Format Type** 选择 JSON 或 XML。

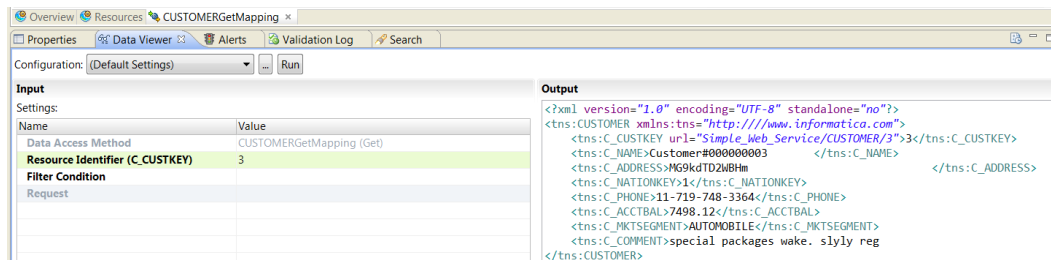
下图显示了资源映射的**属性视图的高级选项卡**：



响应数据预览

可以在 Developer tool 的**数据查看器**视图中测试 Web 服务。可以按资源键筛选数据，也可以配置表达式来筛选数据。

下图显示了“数据查看器”视图：



要在**数据查看器**视图中筛选响应消息，请配置以下输入设置：

数据访问方法

要运行的资源映射。一个 Web 服务可以包含多个资源映射。

资源标识符

要用来进行搜索的键值。Developer tool 会标识资源键的名称。

筛选条件

可在表达式编辑器中配置的筛选器表达式。该表达式可以引用多个输出列。如果输出包含层次结构数据，则筛选条件必须引用父组中的元素。可以同时输入筛选条件和资源标识符。

第 11 章

如何创建 REST Web 服务

本章包括以下主题：

- [创建 REST Web 服务, 109](#)
- [如何手动创建 REST Web 服务, 110](#)
- [步骤 1. 创建 REST Web 服务资源, 110](#)
- [步骤 2. 定义资源映射, 113](#)
- [步骤 3. 配置输出映射, 116](#)
- [步骤 4. 在“数据查看器”视图中测试映射, 118](#)
- [步骤 5. 部署应用程序, 120](#)
- [步骤 6. 从浏览器查询 Web 服务, 121](#)
- [如何从数据对象创建 REST Web 服务, 123](#)
- [如何将数据对象部署为 REST Web 服务, 126](#)

创建 REST Web 服务

可以将数据对象部署为 REST Web 服务，也可以使用向导定义 Web 服务。

创建 Web 服务之前，请确定要包含在响应消息中的元素。可以使用以下方法之一基于响应消息结构创建 Web 服务：

手动创建 Web 服务。

如果响应消息包含必须通过资源映射进行计算或设置格式的数据，请手动创建 Web 服务。如果映射具有多个源，也请手动创建 Web 服务。如果响应消息包含层次结构数据或多次出现的数据，则必须手动定义结构。

从数据对象创建 Web 服务。

如果响应消息包含来自一个数据对象的数据，您可以基于该数据对象创建资源定义。这时可以从数据对象中选择特定列，并且可以在创建 Web 服务之后更改资源映射。

将数据对象部署为 Web 服务。

如果响应消息可以包含数据对象中的所有列，请将数据对象部署为 Web 服务。

如何手动创建 REST Web 服务

可以在 REST Web 服务资源中手动定义列。手动定义资源时，Developer tool 将创建包含一个输出转换（定义了响应消息）的资源映射。Developer tool 不会在资源映射中创建读取转换。定义资源之后，必须向该映射添加读取转换。当响应消息结构与源数据不同时，请手动创建资源。

要创建 REST Web 服务，请执行以下步骤：

1. 创建 REST Web 服务资源。
2. 定义用于检索数据的资源映射。
3. 配置 REST 输出转换映射，以将来自输入端口的数据映射到输出层次结构中的元素。
4. 测试该映射。
5. 将应用程序部署到数据集成服务。
6. 从浏览器访问 Web 服务。

示例 REST Web 服务

Hypostores 有一个 REST Web 服务，该服务将返回客户的所有订单。销售员可以从 Web 浏览器向该 Web 服务发送请求以获取订单。该请求包含一个或多个客户编号。该 Web 服务将在 JSON 文件中返回每个客户编号和该客户所有订单的列表。

该 REST Web 服务包含一个映射，此映射将读取 Customers 表以检索客户名称。此映射包含一个查找转换。该查找转换将从 Orders 表检索每个客户的所有订单。此映射还包含一个将返回分层 JSON 文件的 REST 输出转换。JSON 文件结构中的父组包含客户编号和客户名称。该 JSON 文件还包含一个代表客户订单的子组。此组包含多次出现的订单。每个订单都具有订单编号、价格和订单日期。

步骤 1. 创建 REST Web 服务资源

REST Web 服务资源包含 REST Web 服务响应消息的定义和用于返回响应的数据访问方法。创建 Informatica REST Web 服务时，可以从数据对象定义资源，或者也可以手动定义资源。

在此示例中，您需要手动创建资源，因为响应消息为层次结构格式并且包含多次出现的数据。无法从存储库中的数据对象创建资源。

创建 REST Web 服务资源

创建 Web 服务资源时，您需要定义响应消息的结构。

1. 在 Developer tool 中，单击**文件 > 新建 > 数据服务**。
2. 选择 **REST Web 服务**，然后单击**下一步**。
3. 输入 Orders_Web_Service 作为 Web 服务名称。单击**下一步**。
4. 在 **REST 资源**对话框中，单击**从空对象创建**。

此时将显示默认资源。

New REST Web Service

REST Resource
Create REST resources and define mappings on REST resources.

Name: type filter text

Name: Resource

Description:

Definition:

Name	Type	Key	Min. Occurs	Max. Occurs	Description

HTTP method(s):
☒ Get

< Back Next > **Finish** Cancel

- 5. 在**名称**字段中，将资源名称从 Resource 更改为 Orders_Resource。
- 6. 在**定义**面板上，单击**新建 > 元素**。
- 7. 输入以下元素：

名称	类型	键	最小出现次数	最大出现次数
Customer_Key	string	是	1	1
Customer_Name	string		1	1
Orders	string		1	无限大

- 8. 选择 Orders 元素，然后单击**新建 > 子元素**。
- 9. 在 Orders 下面输入以下子元素：

名称	类型	键	最小出现次数	最大出现次数
Order_Key	integer		1	1
Order_Price	decimal		1	1
Order_Date	integer		1	1

- 10. 验证是否已启用 Get HTTP 方法。

11. 单击完成。

下图显示了 Orders_Resource 中的元素：

The screenshot shows a 'New REST Web Service' dialog box. The 'Name' field is 'Orders_Resource' and the 'Description' is 'Returns orders by customer'. The 'Definition' section contains a table with the following data:

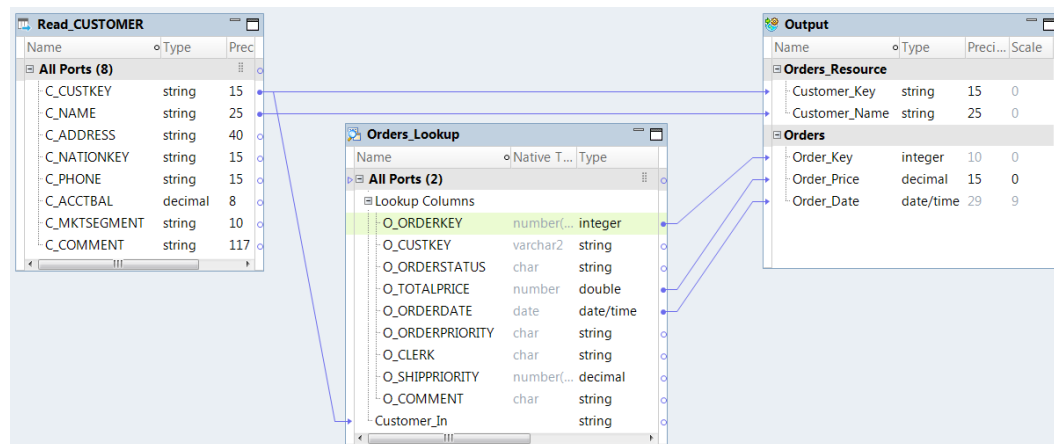
Name	Type	Key	Min...	Max. Occurs	Description
Customer_Key	xs:string	<input checked="" type="radio"/>	1	1	
Customer_Name	xs:string	<input type="radio"/>	1	1	
Orders	(Orders)	<input type="radio"/>	1	Unbounded	
Order_Key	xs:int...	<input type="radio"/>	1	1	
Order_Price	xs:dec...	<input type="radio"/>	1	1	
Order_Date	xs:date	<input type="radio"/>	1	1	

The 'HTTP method(s)' section has a checked box for 'Get'. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

步骤 2. 定义资源映射

定义资源之后，Developer tool 将创建包含一个 REST 输出转换的资源映射。REST 输出转换结构定义了 Web 服务将返回到客户端的响应消息。请向该映射添加读取转换和查找转换。

下图显示了要创建的资源映射：



该映射包含以下对象：

Read_Customer

Read_Customer 转换将读取客户表。此转换将返回表中的所有客户行。

Orders_Lookup

此查找转换将检索每个客户的所有订单。

Output

此 REST 输出转换将接收来自 Read_Customer 转换的客户信息。此转换将接收来自 Orders_Lookup 转换的订单信息。此 REST 输出转换将生成一个分层 JSON 文件，其中列出了每个客户的订单。

注意：该资源映射不包含用于限制客户从客户表检索的筛选器转换。

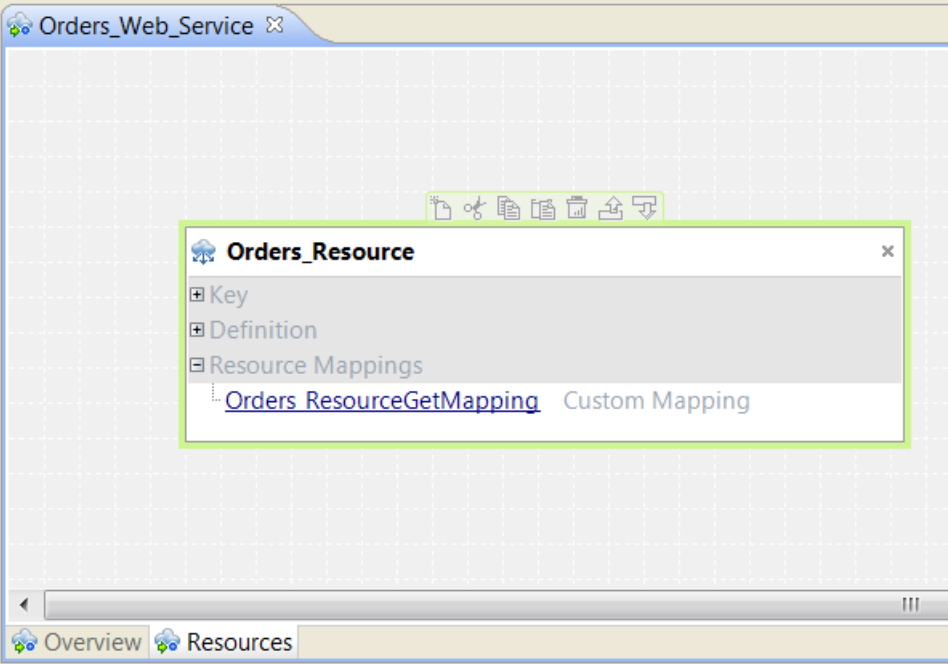
数据集成服务将根据客户端请求中的任何筛选器来筛选输出数据。

定义资源映射

Developer tool 将创建一个资源映射，其中包含已创建的 REST 输出转换。可以在该资源映射中定义一个读取转换和一个查找转换。

1. 在资源选项卡中，展开 Orders_Resource 中的资源映射链接。

下图显示了 Orders_Resource 中的 Orders_Resource 链接：



2. 单击 Orders_Resource 中的 **Orders_ResourceGetMapping** 链接。
此时映射将显示在编辑器中。
3. 将 Customer 数据对象作为读取转换添加到映射中。
4. 单击 REST 输出转换。
5. 在 REST 输出转换的属性选项卡上，单击端口视图。
6. 将 Customer_Key 的精度更改为 15，并将 Customer_Name 的精度更改为 25。

A screenshot of the 'Properties' window in a software application. The 'Ports' tab is selected. It shows a table of ports for the 'Orders_Resource' mapping. The table has columns: Name, Type, Precision, Scale, and Description. There are two ports listed for 'Orders_Resource': 'Customer_Key' (string, precision 15, scale 0) and 'Customer_Name' (string, precision 25, scale 0). Below this, there are four ports listed for 'Orders': 'Order_Key' (integer, precision 10, scale 0), 'Order_Price' (decimal, precision 15, scale 0), 'Order_Date' (date/time, precision 29, scale 9), and 'Customer_FKey' (string, precision 15, scale 0).

	Name	Type	Preci...	Scale	Description
Orders_Resource					
1	Customer_Key	string	15	0	
2	Customer_Name	string	25	0	
Orders					
1	Order_Key	integer	10	0	
2	Order_Price	decimal	15	0	
3	Order_Date	date/time	29	9	
4	Customer_FKey	string	15	0	

7. 将 Read_Customer 中的 Customer_Key 端口和 Customer_Name 端口连接到 REST 输出转换。

下图显示了 Read_Customer 转换和 REST 输出转换之间的链接：

Name	Type	Prec
All Ports (8)		
C_CUSTKEY	string	15
C_NAME	string	25
C_ADDRESS	string	40
C_NATIONKEY	string	15
C_PHONE	string	15
C_ACCTBAL	decimal	8
C_MKTSEGMENT	string	10
C_COMMENT	string	117

Name	Type	Preci...	Scale
Orders_Resource			
Customer_Key	string	15	0
Customer_Name	string	25	0
Orders			
Order_Key	integer	10	0
Order_Price	decimal	15	0
Order_Date	date/time	29	9

- 8. 要添加查找转换，请在映射中右键单击，然后单击**添加转换**。
- 9. 选择查找转换。
- 10. 选择“关系数据对象查找”，然后单击**下一步**。
- 11. 在**新建查找转换**对话框中，浏览到并选择 Orders 物理数据对象。选择“返回所有行”。

New Lookup Transformation

Lookup

Create a relational data object Lookup transformation.

Relational Data Object:

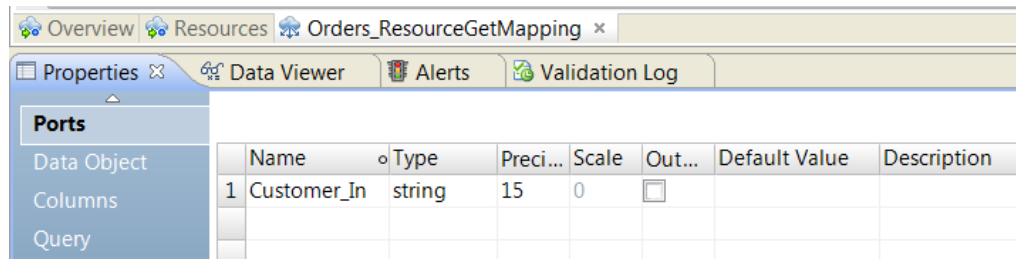
Name:

Location:

On multiple matches:

- 12. 单击**完成**。
- 13. 单击映射中的查找转换以将其选定。
- 14. 在查找转换的**属性**视图中，选择**端口**选项卡。

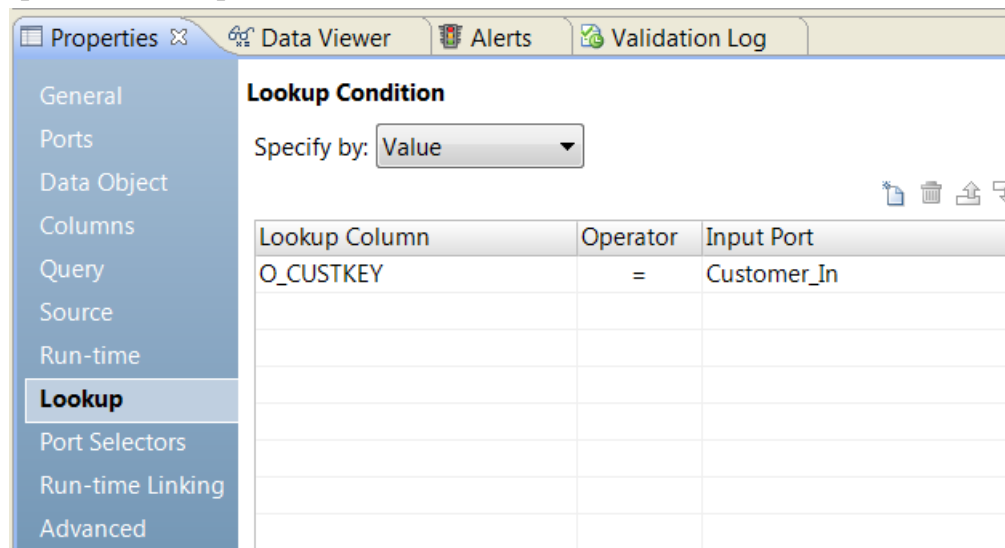
15. 添加名为 Customer_In 的端口。此端口是一个字符串，其精度为 15。不必为输出启用此端口。



16. 在查找转换的属性视图中，单击查找选项卡。

17. 输入以下查找条件：

O_CUSTKEY = Customer_In



18. 在映射画布中，将 Read_Customer 中的 C_CUSTKEY 端口链接到查找转换中的 Customer_In 端口。

19. 将查找转换中的以下输出端口链接到 REST 输出转换中的端口：

查找转换	输出转换
O_ORDERKEY	Order_Key
O_TOTALPRICE	Order_Price
O_ORDERDATE	Order_Date

步骤 3. 配置输出映射

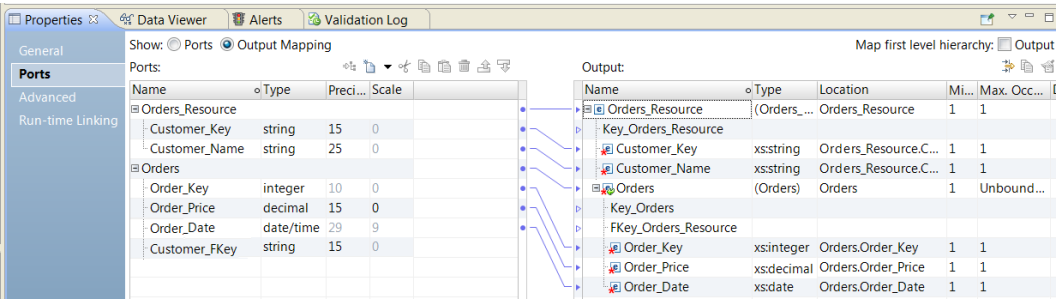
REST 输出转换将接收来自 Customer 转换和 Orders 转换的信息，并返回一个 JSON 文件。

在 Orders_Resource 映射中，REST 输出转换包含两个输出数据组。父组包含客户信息，并且对客户仅出现一次。子组包含订单。Orders 组多次出现。

如果 REST 输出转换包含多个端口组，Developer tool 将创建一些键来将这些组链接起来。Developer tool 会在每个组中创建一个主键。Developer tool 会为每个子组创建一个外键。Orders 组中的外键包含每个订单的客户编号。

查看 REST 输出转换中的端口时，不会显示键。键不具有端口。但是，必须将数据链接到键才能定义键值。要查看键，请在 REST 输出转换的**属性**选项卡的**端口**视图中显示输出映射。**端口**视图将显示输入端口或输出映射，具体取决于您选择要显示的选项。

下图显示了**输出映射**视图：



Developer tool 将在输出中创建以下键：

键名称	说明
Key_Orders_Resource	Orders_Resource 组的主键。
Key_Orders	Orders 组的主键。
FKey_Orders_Resource	用于将 Orders 组链接到 Orders_Resource 组的外键。

必须将输入端口链接到输出键，否则映射无效。

注意: 可以将一个端口链接到**输出**面板上的多个元素。但是，不能将**端口**面板上的一个组中的字段链接到**输出**面板上的多个组中的元素。同一个**端口**组中的所有字段必须链接到同一个**输出**组中的元素。

配置输出映射

要在客户和订单之间创建主键外键关系，每个组都必须包含客户编号。

- 在 REST 输出转换的 Orders 组中，在 Order_Date 下面添加 Customer_Fkey 端口。
Customer_Fkey 端口将接收客户编号。您需要添加此端口，以接收 Orders 组中的客户编号以及填充外键。客户编号并不与每个订单一起显示在输出中。
- 将**端口**面板中的 Customer_Key 链接到**输出**面板中的 Key_Orders_Resource。要创建链接，请单击 Customer_Key 将其选定，然后选择 Key_Orders_Resource。单击**端口**面板上的**映射**图标以创建链接。
注意: 不能通过拖动端口来创建链接。
- 选择 Order_Key 端口和 Key_Orders 元素。单击**映射**以创建链接。
Order_Key 必须具有与 Key_Orders 元素和 Order_Key 元素的链接。

4. 选择 Customer_Fkey 端口和 Fkey_Orders_Resource 元素。单击**映射**以创建链接。

下图显示了具有与多个键的链接的输出映射：

Name	Type	Preci...	Scale
Orders_Resource	string	15	0
Customer_Key	string	25	0
Orders	integer	10	0
Order_Price	decimal	15	0
Order_Date	date/time	29	9
Customer_FKey	string	15	0

Name	Type	Location	Min. O...	Max. Occurs
Orders_Resource	(Orders_...	Orders_Resource	1	1
Key_Orders_Resource	Orders_Resource.C...	Orders_Resource.C...	1	1
Customer_Key	xsstring	Orders_Resource.C...	1	1
Customer_Name	xsstring	Orders_Resource.C...	1	1
Orders	(Orders)	Orders	1	Unbounded
Key_Orders	Orders.Order_Key	Orders.Order_Key	1	1
FKey_Orders_Resource	Orders.Customer_F...	Orders.Customer_F...	1	1
Order_Key	xsinteger	Orders.Order_Key	1	1
Order_Price	xsdecimal	Orders.Order_Price	1	1
Order_Date	xsdate	Orders.Order_Date	1	1

步骤 4. 在“数据查看器”视图中测试映射

可以在**数据查看器**视图中运行 REST 资源映射。可以在**数据查看器**视图中按资源 ID 或筛选条件筛选输出数据。

可以通过输入 Orders_Resource 组中的元素的值来筛选数据。元素是 Customer_Key 或 Customer_Name。不能使用 Orders 组中的元素来筛选输出数据。Orders 组是一个子组。

可以按资源标识符筛选输出数据，也可以输入筛选条件。资源标识符是键，即 Customer_Key 元素。筛选条件是可在表达式编辑器中输入的表达式。如果不按键进行搜索，可以使用筛选条件来筛选输出数据。

如果未输入资源标识符或筛选条件，映射将在**数据查看器**视图中返回所有行。

按资源 ID 筛选输出

可以输入资源 ID，以便按键值选择输出数据。资源 ID 是在创建资源时选择的键。在此示例中，资源 ID 是 Customer_Key。

1. 在**数据查看器**视图中，在资源标识符的**值**列中输入有效的客户编号。
2. 单击**运行**。

输出将包含所输入的客户的数据。

下图显示了客户编号 10 的输出：

Name	Value
Data Access Method	Orders_ResourceGetMapping (Get)
Resource Identifier (Customer_Key)	10
Filter Condition	
Request	

```
{
  "Orders_Resource": {
    "Customer_Key": {
      "@url": "Orders_Web_Service\\u002FOrders_Resource\\u002F10",
      "$": 10,
    },
    "Customer_Name": "Cust128",
    "Orders": [
      {
        "Order_Date": "1996-01-05",
        "Order_Key": 18151,
        "Order_Price": 295014.94
      },
      {
        "Order_Date": "1996-10-31",
        "Order_Key": 20357,
        "Order_Price": 59049.28
      }
    ]
  }
}
```

按筛选条件筛选输出

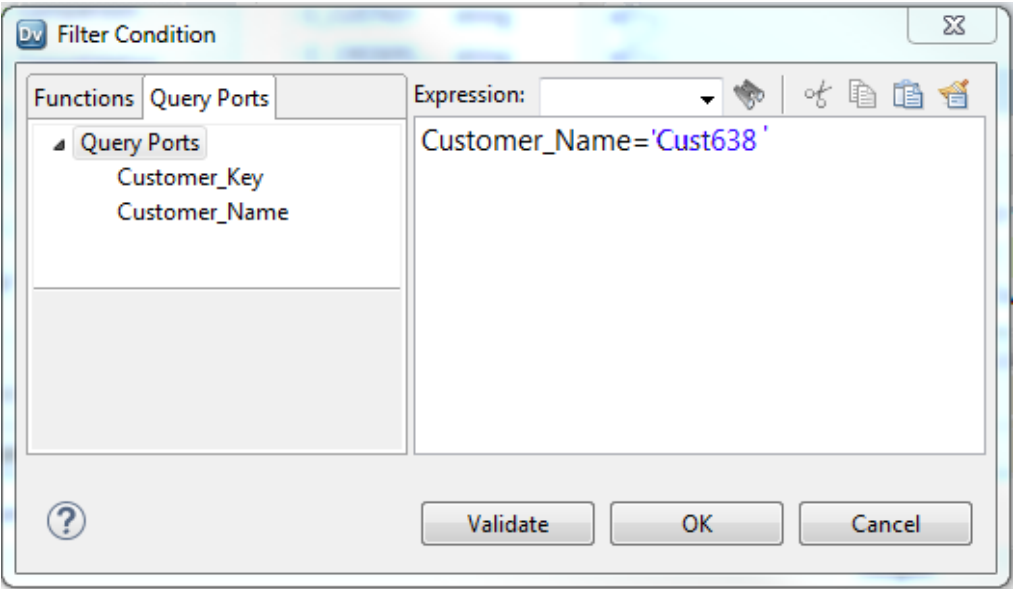
可以输入筛选条件，以便按特定值选择输出数据。

1. 在**数据查看器**视图中，在筛选条件的值列中单击选择箭头。

此时将显示表达式编辑器。

2. 单击**查询端口**选项卡。

此时将显示端口列表：



3. 选择查询端口并创建筛选器表达式。

必须用单引号将搜索值括起来。

4. 单击**验证**以验证表达式。

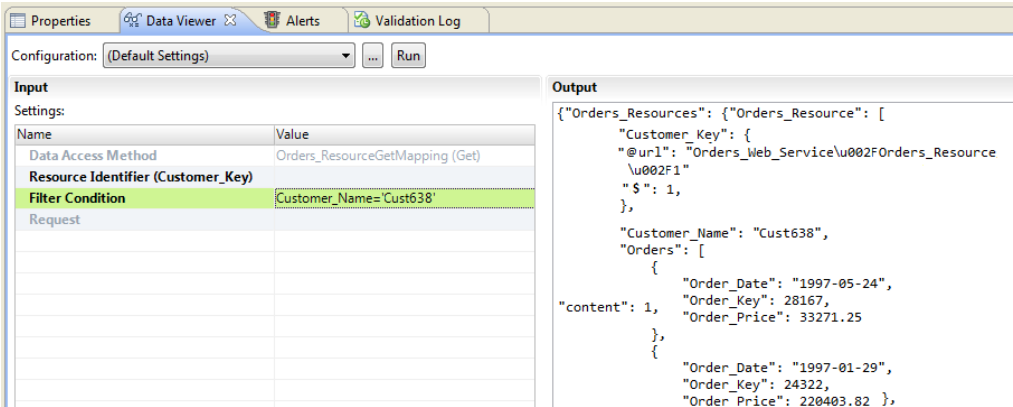
5. 如果表达式有效，则单击**确定**。

该表达式将作为筛选条件显示在**数据查看器**视图中。

6. 在**数据查看器**视图中，单击**运行**。

客户名称为 Cust638 的订单会显示在“输出”面板上。

下图显示了“输出”面板上的结果：



步骤 5.部署应用程序

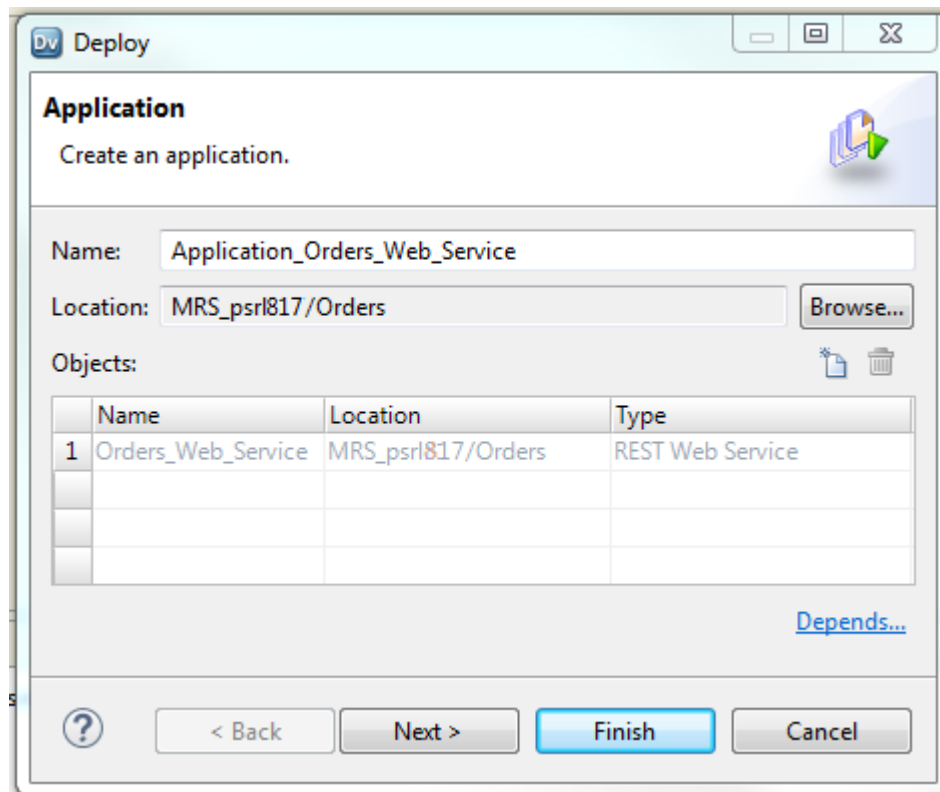
将 Web 服务作为应用程序部署到数据集成服务。必须部署应用程序，以使 Web 服务客户端能够连接到 Web 服务。

部署 Web 服务之后，可以在 Administrator 工具中查看应用程序。

部署应用程序

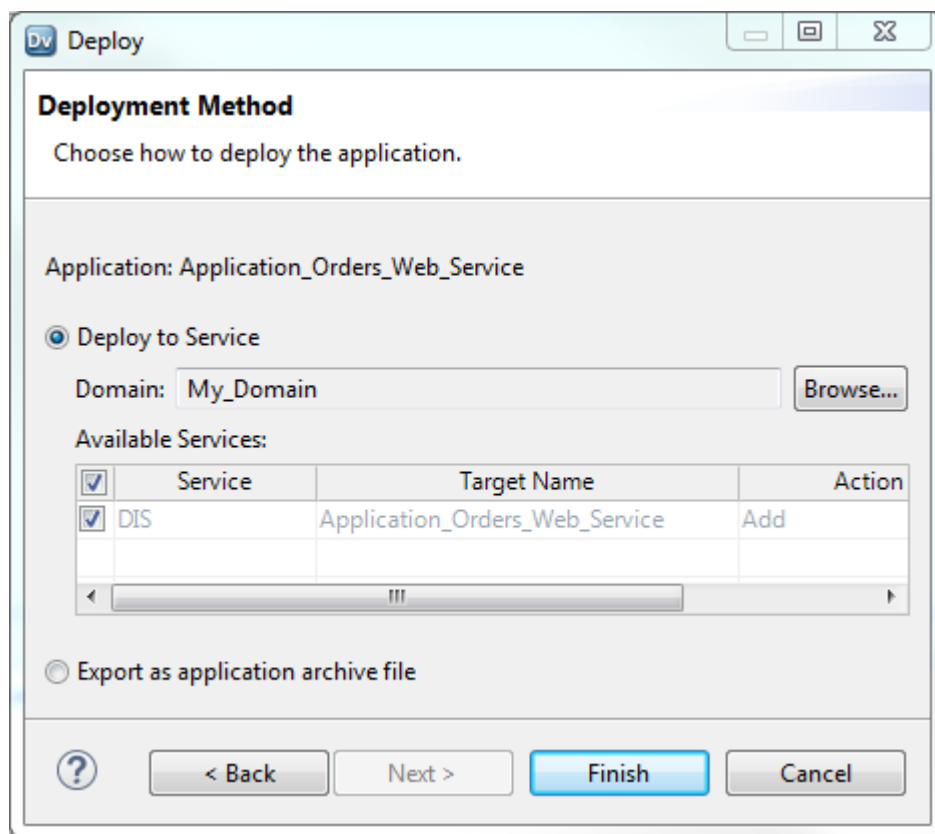
将应用程序部署到数据集成服务。

1. 在对象浏览器视图中，右键单击 Orders_Web_Service，然后单击部署：



2. 接受默认名称和位置。单击下一步。

3. 选择要将应用程序部署到的域和数据集成服务。



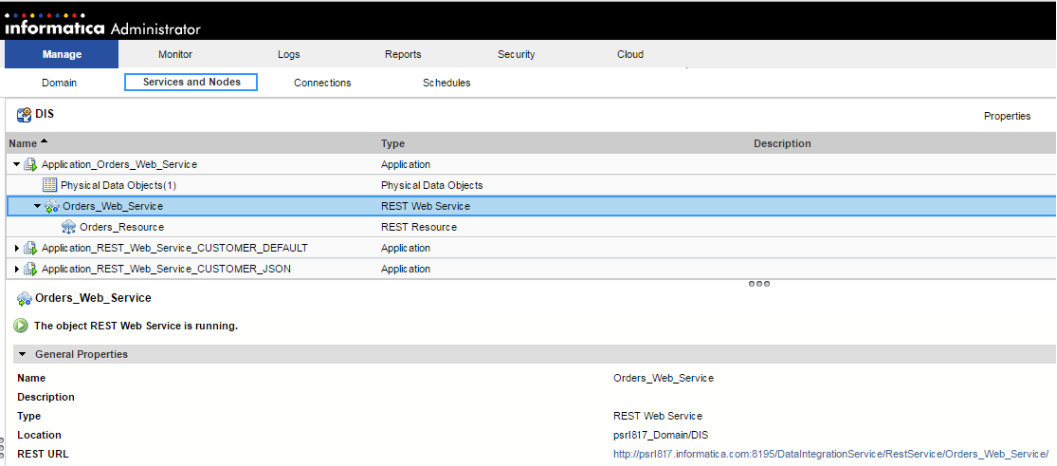
4. 单击**完成**以部署应用程序。

步骤 6. 从浏览器查询 Web 服务

从浏览器查询 REST Web 服务。

将 Web 服务部署到数据集成服务之后，该 Web 服务会作为应用程序显示在 Administrator 工具中。可以在应用程序的**常规属性**面板上查看用于访问该 Web 服务的 URL。

下图显示了 Web 服务的常规属性面板上的 REST URL：



查询 Web 服务

可以通过在 URL 中包含资源 ID 或筛选器表达式来查询 Web 服务。

1. 在 Administrator 工具中，在 Web 服务应用程序的常规属性面板上单击 REST URL。

数据集成服务将列出该 Web 服务中的资源。

下图显示了 Orders_Web_Service 中的资源：



Resources:

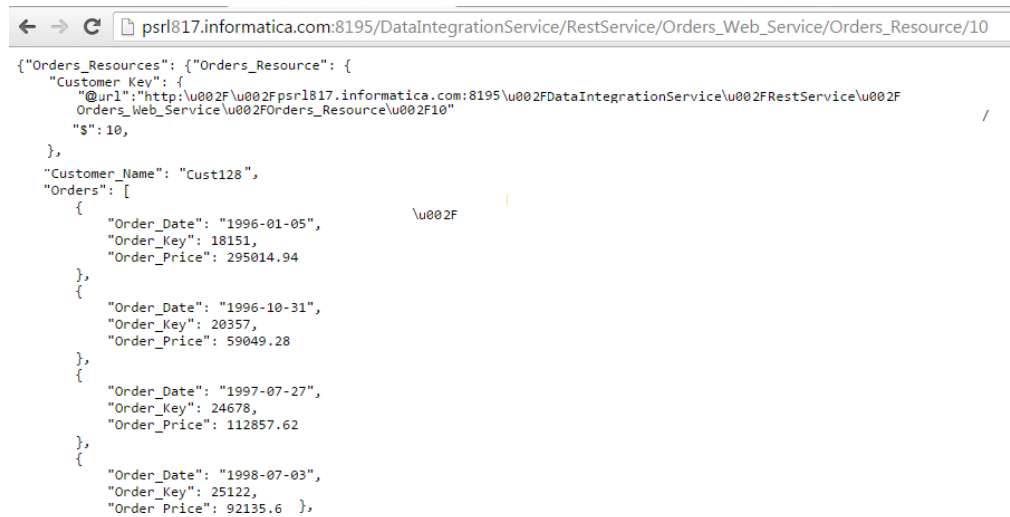
- [Orders_Resource](#)

2. 单击 **Orders_Resource** 链接。

Web 服务将从资源映射返回所有输出数据。

3. 要搜索特定客户，请在浏览器中将资源名称和资源 ID 添加到 URL 的结尾。该链接为 http://psr1817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/10。

下图显示了在 Orders_Resource 中搜索客户编号 10 的查询结果：



The screenshot shows a REST client interface with a browser-like address bar containing the URL: `psrl817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/10`. The main area displays a JSON response for a search query. The response is a JSON object with the following structure:

```
{
  "Orders_Resources": {
    "Orders_Resource": {
      "Customer_Key": {
        "@url": "http://psrl817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/10",
        "$": 10,
      },
      "Customer_Name": "Cust128",
      "Orders": [
        {
          "Order_Date": "1996-01-05",
          "Order_Key": 18151,
          "Order_Price": 295014.94,
        },
        {
          "Order_Date": "1996-10-31",
          "Order_Key": 20357,
          "Order_Price": 59049.28,
        },
        {
          "Order_Date": "1997-07-27",
          "Order_Key": 24678,
          "Order_Price": 112857.62,
        },
        {
          "Order_Date": "1998-07-03",
          "Order_Key": 25122,
          "Order_Price": 92135.6,
        },
      ]
    }
  }
}
```

4. 要按客户名称而非键进行搜索，请在 URL 中包含筛选条件而非资源 ID：

`http://psrl817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/?filterCondition=Customer_Name='Cust628'`

如何从数据对象创建 REST Web 服务

可以从模型存储库中的数据对象创建 REST Web 服务。Developer tool 将创建包含一个读取转换和一个输出转换的默认映射。可以选择要包含在 REST Web 服务资源中的数据对象列。

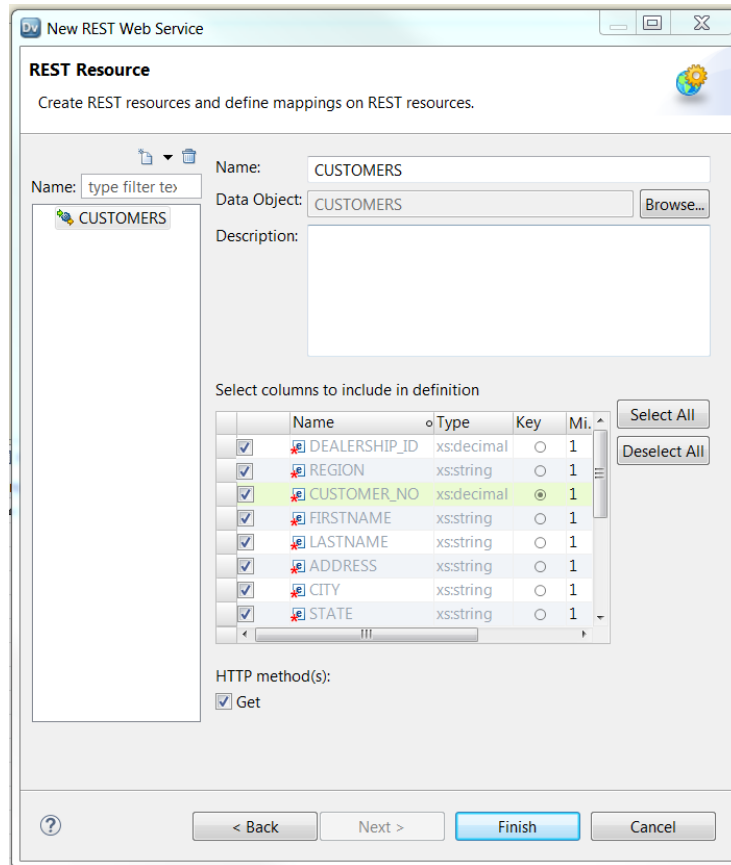
1. 单击**文件 > 新建 > 数据服务**。
2. 在**选择向导**对话框中，选择**REST Web 服务**。单击**下一步**。
3. 在**REST Web 服务**对话框中，输入 Web 服务的名称。默认名称为 `REST_Web_Service`。单击**下一步**。
4. 在“REST 资源”对话框中，单击**新建 > 从数据对象创建**。
5. 从模型存储库的数据对象列表中选择数据对象，然后单击**确定**。

“REST 资源”对话框显示了从所选数据对象创建的 REST 资源。

6. 选择要包含在资源中的列。选择资源的主键。

默认情况下，所有列均处于选中状态。可以取消选中不想包含在资源中的列。默认情况下选择 GET 方法。请勿清除此选项。

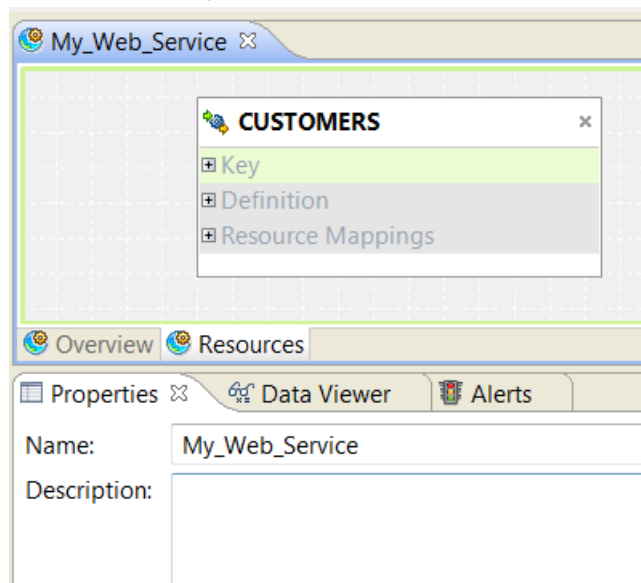
下图显示了“REST 资源”对话框：



7. 单击**完成**以创建 Web 服务。

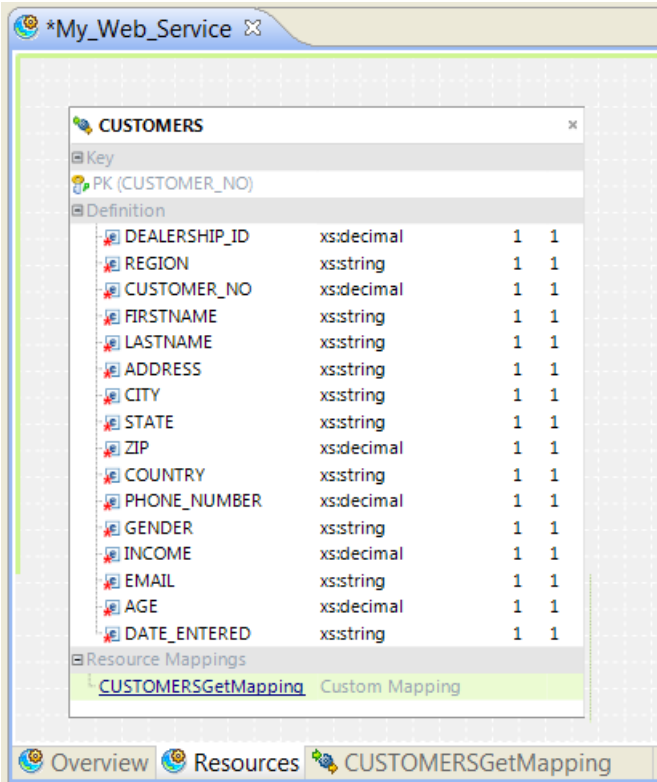
Developer tool 中会显示该 REST Web 服务的**资源视图**。

下图显示了 Developer tool 中的**资源视图**：



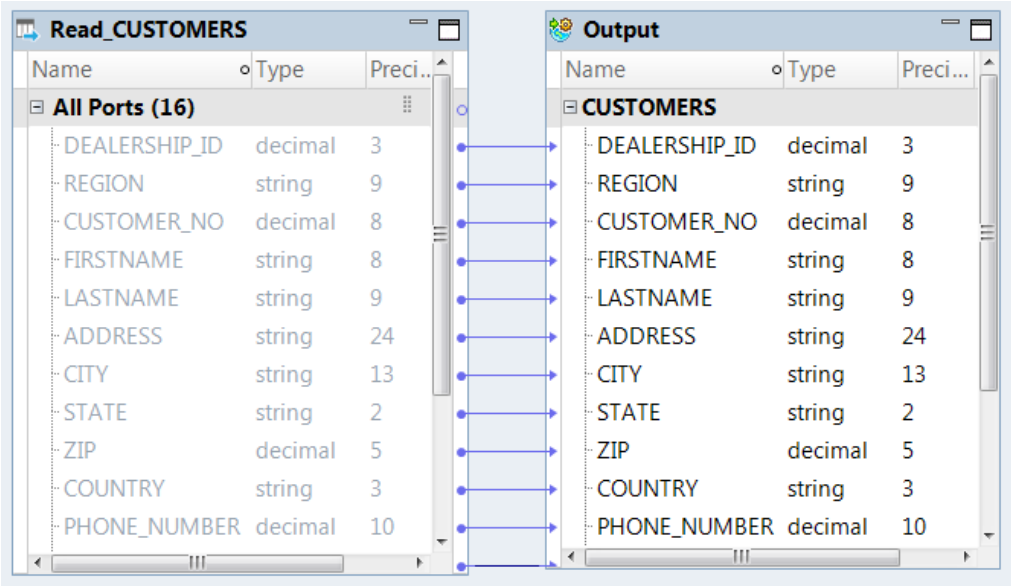
8. 可以展开**资源视图**中的组件以查看键、定义中的列和资源映射链接。

下图显示了资源键、定义列和资源映射链接：



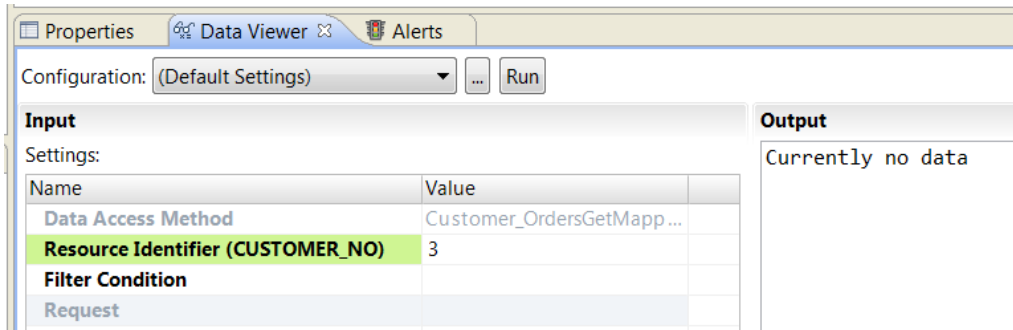
9. 单击资源映射链接以查看资源映射。

下图显示了默认资源映射：



10. 根据需要向映射添加转换。
11. 要测试 Web 服务，请导航到数据查看器视图。
12. 在输入设置中，在数据访问值字段中选择要测试的资源映射。
13. 要按键筛选输出，请在资源标识符字段中输入要搜索的键。

14. 要按其他元素筛选输出，请输入筛选条件。单击“值”字段以打开表达式编辑器。
可以在表达式编辑器中输入多个表达式。
下图显示了数据查看器视图中的输入设置：



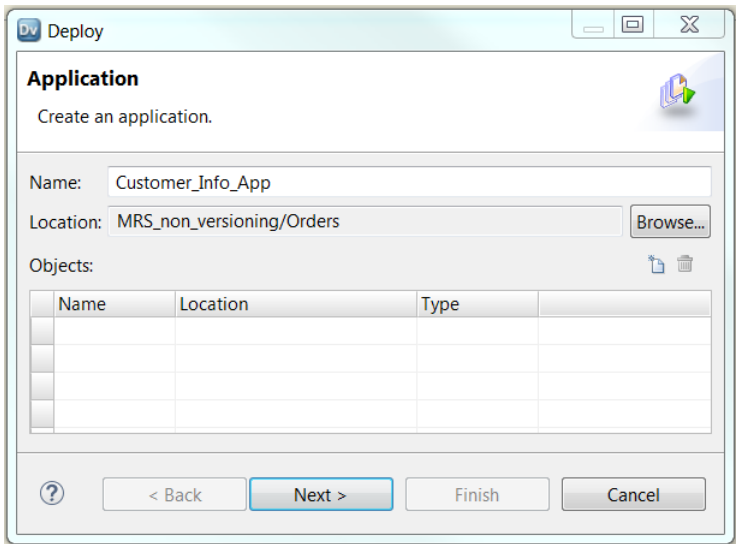
15. 单击**运行**以查看输出。
输出为 JSON 或 XML 格式，具体取决于 Web 服务配置。
16. 将 Web 服务部署到数据集成服务。

如何将数据对象部署为 REST Web 服务

可以将关系数据对象或平面文件数据对象部署为 REST Web 服务。

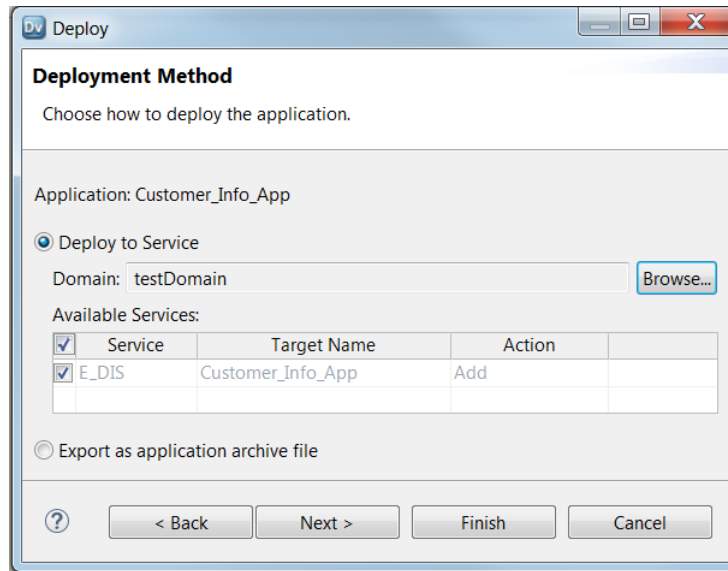
1. 在对象浏览器中，右键单击要在 Web 服务中访问的数据对象。
2. 单击**部署 > 部署为 REST Web 服务**。

此时将显示**部署应用程序**对话框。



3. 输入应用程序的名称，然后单击**下一步**。请勿添加任何对象。
此时将显示**部署方法**对话框。
4. 选择要将应用程序部署到的数据集成服务。

下图显示了部署方法对话框：

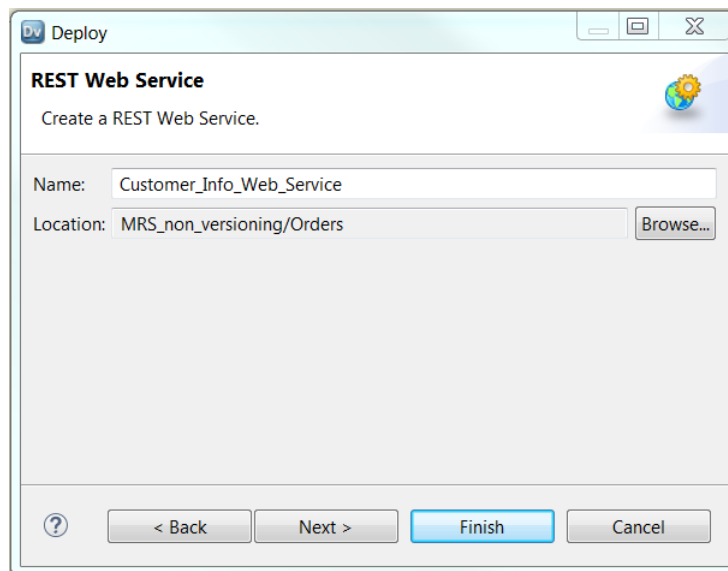


5. 单击**下一步**。

此时将显示 **REST Web 服务**对话框。

6. 输入 Web 服务的名称。

下图显示了 **REST Web 服务**对话框：



7. 单击**下一步**可选择要包含在资源中的列或添加更多资源。单击**完成**可创建包含数据对象中所有列的资源。

如果单击**下一步**，将显示 **REST 资源**对话框。

8. 在“REST 资源”对话框上，输入资源的名称并选择要包含在定义中的列。

默认情况下，所有列均处于选中状态。请勿清除任何字段。已部署的 Web 服务需要默认映射。

下图显示了“REST 资源”对话框：

REST Resource
Create REST resources and define mappings on REST resources.

Name: type filter text
CUSTOMERS

Name: CUSTOMERS
Data Object: CUSTOMERS
Description: Return customer information by customer ID

Select columns to include in definition

Name	Type	Key	Mi...	M...	Desc
DEALERSHIP_ID	xs:decimal	<input type="radio"/>	1	1	
REGION	xs:string	<input type="radio"/>	1	1	
CUSTOMER_NO	xs:decimal	<input checked="" type="radio"/>	1	1	
FIRSTNAME	xs:string	<input type="radio"/>	1	1	
LASTNAME	xs:string	<input type="radio"/>	1	1	
ADDRESS	xs:string	<input type="radio"/>	1	1	
CITY	xs:string	<input type="radio"/>	1	1	
STATE	xs:string	<input type="radio"/>	1	1	
ZIP	xs:decimal	<input type="radio"/>	1	1	
COUNTRY	xs:string	<input type="radio"/>	1	1	
PHONE_NUMBER	xs:decimal	<input type="radio"/>	1	1	
GENDER	xs:string	<input type="radio"/>	1	1	
INCOME	xs:decimal	<input type="radio"/>	1	1	

Select All
Deselect All

HTTP method(s):
☒ Get

< Back Next > Finish Cancel

9. 单击**完成**以部署 Web 服务。
请勿添加资源或更改资源定义。
10. 要向 Web 服务添加更多资源，请单击**新建**。
可以从数据对象创建资源，也可以在资源中手动输入列。

第 12 章

REST Web 服务使用者转换

本章包括以下主题：

- [REST Web 服务使用者转换概览, 129](#)
- [REST Web 服务使用者转换配置, 130](#)
- [HTTP 方法, 132](#)
- [REST Web 服务使用者转换端口, 134](#)
- [REST Web 服务使用者转换输入映射, 136](#)
- [REST Web 服务使用者转换输出映射, 138](#)
- [REST Web 服务使用者转换高级属性, 140](#)
- [REST Web 服务使用者转换创建, 141](#)

REST Web 服务使用者转换概览

REST Web 服务使用者转换是一种主动转换，其作为 Web 服务客户端连接到 REST Web 服务，以访问或转换数据。使用 REST Web 服务使用者转换可连接到 REST Web 服务。REST Web 服务使用者转换可以向 REST Web 服务发送请求以及从 REST Web 服务接收响应。

REST Web 服务使用者转换通过您在该转换中或在 HTTP 连接中定义的 URL 连接到 Web 服务。还可以使用 HTTPS 连接。REST Web 服务使用者转换可以使用 TLS 1.2、TLS 1.1 或 TLS 1.0。

对于 Web 服务所支持的每一个操作，REST Web 服务均包含一个 HTTP 方法。当数据集成服务连接到 REST Web 服务时，其可以发送获取数据、发布数据、放置数据或删除数据的请求。该请求可对单个资源或资源集合执行。数据集成服务发送请求消息后，其从 Web 服务接收响应消息。

请求和响应消息包含 XML 或 JSON 数据，其中包含的元素可形成一个层次结构。如果请求或响应消息包含多次出现元素，则元素组将形成 XML 或 JSON 层次结构中的级别。如果一个级别嵌在另一个级别中，则组之间产生关联。

在 REST Web 服务使用者转换中，方法输入和方法输出定义请求和响应消息的结构。方法输入和方法输出包含一些映射，这些映射定义如何将消息元素映射到输入和输出端口。

REST Web 服务使用者转换支持代理服务器。还可以将 Microsoft SharePoint 应用程序与 REST Web 服务使用者转换连接。

示例

联机存储定义产品数据库的资源。数据库通过部件号标识每个产品。

Web 服务客户端通过 REST Web 服务访问产品详细信息。Web 服务使用以下 URL：

<http://www.HypoStores.com/products/ProductDetails>

需要检索有关特定产品的详细信息（如说明和单价），并将详细信息传递给映射中的转换下游。创建 REST Web 服务使用者转换以检索有关产品的详细信息，并将这些信息传递给其他转换。

下表显示了可以配置的转换详细信息：

转换详细信息	值
HTTP 方法	Get
基本 URL	http://www.HypoStores.com/products/ProductDetails
输入参数端口	Part_No
输出端口	说明, Unit_Price
方法输出	<响应消息的结构。>

方法输出包括输出映射，可定义响应消息中的元素如何映射到输出端口。

当数据集成服务向 Web 服务发送请求时，其将参数端口中的值附加到基础 URL。例如，为检索有关部件 0716 的详细信息，数据集成服务使用以下 URL：

http://www.HypoStores.com/products/ProductDetails?Part_No=0716

当数据集成服务接收响应时，其将响应消息中的产品说明和单价转换为输出端口的数据。

也可以将 Part_No 作为参数传递，并在运行映射时置换值中游。

REST Web 服务使用者转换流程

REST Web 服务使用者转换根据输入端口和方法输出中的数据创建请求消息。其基于方法输出将响应消息中的元素转换为输出端口的数据。

REST Web 服务使用者转换的输入端口包含来自映射中的上游转换的关系数据。数据集成服务使用方法输入将来自输入端口的数据转换为请求消息中的元素。

为连接 Web 服务，数据集成服务读取您在转换属性或 HTTP 连接中配置的基础 URL。其标识要获取、发布、放置或删除的资源，方法是将来自 URL 端口或参数端口值附加到基础 URL。

当数据集成服务接收响应时，其将响应消息中的数据传递给转换的输出端口。数据集成服务根据方法输出的配置方式传递数据。输出端口包含关系数据。数据集成服务将输出端口中的数据发送到映射中的下游转换或者发送到目标中。

REST Web 服务使用者转换配置

创建 REST Web 服务使用者转换时，选择 HTTP 方法并定义方法输入和输出。如果选择 Get 方法，则不定义方法输入。

HTTP 请求消息中的输入元素映射到输入端口。HTTP 响应消息中的输出元素映射到输出端口。Developer 工具为第一级元素创建端口。

配置转换时，完成以下任务：

1. 选择 HTTP 方法。

2. 配置用以表示请求消息和响应消息的表头和正文中的元素的端口。
3. 配置输入映射。
4. 配置输出映射。
5. 配置高级属性，如 Web 服务的连接和 URL。

如果 REST Web 服务需要进行身份验证，应创建 HTTP 连接对象。

消息配置

数据集成服务生成请求消息，并基于方法输入和输出以及您在 REST Web 服务使用者转换中配置的端口来解释响应消息。

输入端口代表请求消息的不同部分。可以添加标识要检索或更改的资源的输入端口。还可以添加代表请求消息中的 HTTP 表头、cookie 信息和元素的输入端口。

输出端口代表响应消息中要发送到下游转换或发送到映射中的目标的元素。可以代表响应消息中的 HTTP 表头、cookie 信息、响应代码和元素的输出端口。

资源标识

为标识 HTTP 请求中的资源，数据集成服务会将特定输入端口中的值附加到基础 URL。在 HTTP 连接或转换属性中定义基础 URL。使用 URL 或参数端口标识特殊资源。

如果 Web 服务通过唯一的字符串标识资源，则使用 URL 端口。

例如，HypoStores REST Web 服务通过以下 URL 使用部件号标识部件：

`http://www.HypoStores.com/products/ProductDetails/<Part_No>`

要标识部件，应定义以下转换详细信息：

1. 将基础 URL 设置为以下 URL：
`http://www.HypoStores.com/products/ProductDetails`
2. 定义 URL 端口，并通过 URL 端口将部件号传递给转换。

如果映射将部件号 500 传递给 URL 端口，则数据集成服务将在请求消息中使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails/500`

如果 Web 服务通过参数标识资源的位置，则使用参数端口。

例如，您想通过“Part_No”参数将部件号传递给 HypoStores REST Web 服务。

要标识部件，应定义以下转换详细信息：

1. 将基础 URL 设置为以下 URL：
`http://www.HypoStores.com/products/ProductDetails`
2. 创建参数名为“Part_No”的参数端口，并通过该参数端口将部件号传递给转换。

如果映射将部件号 600 传递给参数端口，则数据集成服务在请求消息中使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails?Part_No=600`

创建多个参数端口可定义多个参数。数据集成服务使用 & 符号分隔各个参数。

例如，您要从 REST Web 服务检索员工详细信息，并通过“First_Name”和“Last_Name”参数传递员工的名字和姓氏。创建参数名为“First_Name”和“Last_Name”的参数端口。如果映射将名字“John Smith”传递给转换，数据集成服务将在请求消息中使用类似以下内容的 URL：

`http://www.HypoStores.com/employees/EmpDetails?First_Name=John&Last_Name=Smith`

如果不指定 URL 或参数端口，数据集成服务将使用转换属性的基础 URL 或 HTTP 连接标识资源。HTTP 连接中的基础 URL 将替代转换中的基础 URL。

HTTP 方法

创建 REST Web 服务使用者转换时，选择数据集成服务在请求消息中使用的 HTTP 方法。创建转换后无法更改 HTTP 方法。

将转换配置为使用以下 HTTP 方法之一：

Get

从 Web 服务检索资源或资源集合。例如，可以检索产品表或检索有关一个产品的信息。

Post

向 Web 服务发送数据。使用 Post 方法创建资源或资源集合。例如，可以添加新存储事务的详细信息。

放置

替换资源或资源集合。如果数据不存在，Put 方法将发布数据。例如，可以更新客户配送地址。

删除

删除资源或资源集合。例如，可以删除不再为组织工作的员工的记录。

HTTP Get 方法

数据集成服务使用 HTTP Get 方法从 REST Web 服务检索数据。使用 Get 方法可以检索资源或资源集合。

如果将 REST Web 服务使用者转换配置为使用 Get 方法，则应配置输入端口、方法输出和输出端口。无需配置方法输入。

示例

您要检索 HypoStores 产品数据库中部件号 500 的说明和价格。Web 服务使用以下 URL 标识部件：

http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>

输入以下基础 URL：

http://www.HypoStores.com/products/ProductDetails

下表显示了您可定义的输入端口：

端口类型	参数名	输入值
参数	Part_No	500

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	Part_Desc	...<desc>ACME 圆珠笔， 12-pk， 黑色， 0.7 mm</desc>...
输出	Price_USD	...<price>9.89</price>...

HTTP Post 方法

数据集成服务使用 HTTP Post 方法将数据发送到 REST Web 服务。Web 服务决定 Post 方法执行的实际功能。可以使用 Post 方法创建资源或者资源集合。

配置 REST Web 服务使用者转换以使用 Post 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您希望将新部件 501 发布到 HypoStores 产品数据库中。Web 服务对部件 501 使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails/501`

输入以下基础 URL：

`http://www.HypoStores.com/products/ProductDetails`

下表显示了可定义的输入端口：

端口类型	端口名称	输入值
URL	URL_Part_No	501
输入	Part_Desc	ACME 圆珠笔, 12-pk, 黑色, 0.5 mm
输入	Price_USD	9.89

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

HTTP Put 方法

数据集成服务使用 HTTP Put 方法通过 REST Web 服务更新数据。使用 Post 方法更新资源或资源集合。如果数据不存在，数据集成服务将创建资源或资源集合。

配置 REST Web 服务使用者转换以使用 Put 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您要更新 HypoStores 产品数据库中部件号 501 的单价。Web 服务对部件 501 使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails/501`

输入以下基础 URL：

`http://www.HypoStores.com/products/ProductDetails`

下表显示了可定义的输入端口：

端口类型	端口名称	输入值
URL	URL_Part_No	501
输入	Price_USD	9.99

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

HTTP Delete 方法

数据集成服务使用 HTTP Delete 方法通过 REST Web 服务删除数据。使用 Delete 方法可以删除资源或资源集合。

配置 REST Web 服务使用者转换以使用 Delete 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您希望从 HypoStores 产品数据库中删除部件号 502。Web 服务使用以下 URL 标识部件：

`http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>`

输入以下基础 URL：

`http://www.HypoStores.com/products/ProductDetails`

下表显示了您可定义的输入端口：

端口类型	参数名	输入值
参数	Part_No	502

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

REST Web 服务使用者转换端口

一个 REST Web 服务使用者转换可以包含多个输入端口和多个输出端口。您可以根据 XML 或 JSON 层次结构的结构成组创建端口。

查看转换端口时，如果不需要查看 XML 或 JSON 层次结构，则显示端口。显示端口时，可以定义组、定义端口以及将元素从方法输入和方法输出映射到输入端口和输出端口。

一个 REST Web 服务使用者转换可以包含多个输入组和多个输出组。创建端口后，可以创建组，然后将这些端口添加到组中。基于 XML 或 JSON 中输入或输出层次结构的结构在组层次结构中定义端口。通过添加键，可以将子组与父组关联。

在层次结构中，除了最下面的组外，其他所有组都必须具有主键。在层次结构中，除了根组以外，其他所有组都必须具有外键。

该转换包含一个名为 RequestInput 的根输入组。必须在这个根输入组中添加主键。主键必须为 String、Bigint 或 Integer 类型。可以将根输入组中的任何端口配置为传递端口。

要将元素映射到端口，可单击位置列中的字段，然后展开选择位置对话框中的层次结构。接着，从层次结构选择一个元素。

输入端口

输入端口代表您要传递给 Web 服务的上游转换或源中的数据。可以配置多个输入端口。每个输入端口都映射到请求消息中的一个元素。

要添加输入端口，请选择输入组，单击**新建**按钮旁边的箭头，然后选择**字段**。

输出端口

输出端口代表响应消息中要传递给下游转换或传递给目标的元素。可以配置多个输出端口。每个输出端口都映射到响应消息中的一个元素。

要添加输出端口，请选择输出组，单击**新建**按钮旁边的箭头，然后选择**字段**。

传递端口

这些传递端口会通过转换传递数据而不进行更改。可以将根输入组中的任何端口配置为传递端口。

要添加传递端口，可在根输入组中添加一个端口。然后右键单击该端口并选择**映射**。

参数端口

如果资源的 URL 使用参数，则可以使用参数端口标识该资源。在根输入组中添加参数端口。

参数端口具有端口名称和参数名称。如果参数名称包含不允许在端口名称中使用的字符，请输入与端口名称不同的参数名称。例如，您要传递参数“Cust-ID”到 Web 服务，但数据集成服务不允许在端口名称中使用短线字符 (-)。输入“Cust-ID”作为参数名称，但输入“CustID”作为端口名称。

数据集成服务将每个参数端口的参数名称和值以“名称=值”对的形式附加到基础 URL。可以配置多个参数端口。数据集成服务使用 & 符号分隔请求中的多个参数。

例如：

`http://www.HypoStores.com/customers/CustDetails?Last_Name=Jones&First_Name=Mary`

如果在转换中定义参数端口和 URL 端口，数据集成服务会将 URL 端口值附加到基础 URL，后接参数名称和参数值。

要添加参数端口，可右键单击根输入组，然后选择**新建 > 参数端口**。输入参数名称和端口名称。

URL 端口

URL 端口允许您通过静态 URL 标识资源。数据集成服务将 URL 端口的值附加到基本 URL，以标识资源。

例如：

`http://www.HypoStores.com/products/ProductDetails/<URL_port_value>`

将 URL 端口添加到根输入组。

可以配置多个 URL 端口。数据集成服务使用斜杠字符 (/) 分隔每个 URL 端口中的值。如果您在转换中定义了 URL 端口和参数端口，数据集成服务会将 URL 端口值附加到基本 URL，后跟参数名称和值。

要添加 URL 端口，请右键单击根输入组，然后选择**新建 > URL 端口**。

HTTP 表头端口

HTTP 表头端口代表请求消息中的 HTTP 表头。可以配置多个 HTTP 表头端口。

要将表头信息传递给请求中的 Web 服务，可将端口添加到根输入组中。可以为根输入组配置一个 HTTP 表头端口。如果将 HTTP 表头添加到根输入组，可以将其配置为传递端口。

HTTP 表头端口具有端口名称和 HTTP 表头名称。如果 HTTP 表头名称包含不允许在端口名称中使用的字符，请输入与端口名称不同的 HTTP 表头名称。例如，您要将表头名称“Content-Type”传递到 Web 服务，但数据集成服务不允许在端口名称中使用短线字符 (-)。输入“Content-Type”作为 HTTP 表头名称，但输入“ContentType”作为端口名称。

要添加 HTTP 表头端口，可右键单击根输入组，然后选择**新建 > HTTP 表头**。输入表头名称和端口名称。

Cookie 端口

可以配置 REST Web 服务使用者转换以使用 Cookie 身份验证。远程 Web 服务器会根据 cookie 跟踪 Web 服务使用者用户。某个映射多次调用一个 Web 服务时，可以提高性能。

要将 cookie 信息传递给请求中的 Web 服务，可将端口添加到根输入组中。可以为根输入组配置一个 cookie 端口。如果将 cookie 端口添加到根输入组，可以将其配置为传递端口。

要从响应提取 cookie 信息，可在输出组中添加 cookie。可以为每个输出组配置一个 cookie 端口。

将 cookie 端口映射到 Web 服务请求消息时，Web 服务提供程序将在响应消息中返回 cookie 值。可以将该 cookie 值传递到映射中的其他转换下游，也可以将该 cookie 值保存在文件中。如果将 cookie 值保存在文件中，可以将该 cookie 配置为 REST Web 服务使用者转换的输入。

要添加 cookie 端口，可右键单击根输入组，然后选择**新建 > 其他端口**。接着，选择 **Cookie**，并单击**确定**。

输出 XML 端口

输出 XML 端口代表 Web 服务的响应。输出 XML 端口是字符串端口。

将输出 XML 端口添加到输出组。可以为每个输出组配置一个输出 XML 端口。

右键单击根输入组，然后选择**新建 > 其他端口**。接着，选择**输出 XML**，并单击**确定**。

响应代码端口

响应代码端口代表 Web 服务的 HTTP 响应代码。响应代码端口是 Integer 端口。

将响应代码端口添加到输出组。可以为每个输出组配置一个响应代码端口。

要添加响应代码端口，可右键单击根输入组，然后选择**新建 > 其他端口**。接着，选择**响应代码**，并单击**确定**。

REST Web 服务使用者转换输入映射

查看转换端口时，显示输入映射以查看方法输入层次结构。显示输入映射时，可以定义输入组，定义输入端口，并将输入端口映射到方法输入元素。

输入映射包含以下区域：

端口

可在**端口**区域中创建转换输入组和输入端口。

方法输入

方法输入区域显示 REST Web 服务使用者转换发送到 Web 服务的请求消息中的元素。如果使用架构对象创建转换，该架构对象将定义方法输入层次结构。

创建输入端口后，将**端口**区域的输入端口映射到**方法输入**区域中的元素。将输入端口映射到方法输入中的元素时，端口的位置显示在**方法输入**区域的位置列中。

如果选择映射输入第一级层次结构，Developer tool 会将方法输入的第一级中的元素映射到输入端口。Developer tool 还会创建端口以执行映射。如果第一级层次结构中包含的多次出现父元素含有一个或多个多次出现子元素，则 Developer tool 不映射第一级层次结构。

可以选择查看将输入端口连接到方法输入中的元素的行。

将输入端口映射到元素的规则和指导

将输入端口映射到方法输入层次结构中的元素时，请查看以下规则：

- 可以将一个输入端口映射到层次结构中的一个元素。可以将同一端口映射到层次结构中任意数量的键。
 - 输入端口和元素的数据类型必须兼容。
 - 可以将一个输入组中的端口映射到方法输入中的多个层次结构级别。
 - 必须将输入端口映射到方法输入中的键。映射到键的任何端口的数据类型都必须为字符串、整型或长整型。将数据映射到方法输入中位于请求消息所含层次结构级别之上的所有级别中的键。对于映射级别及之上的所有级别，需要包括外键。
- 注意:** 如果只映射方法输入层次结构的最低级别，则无需将输入端口映射到键。
- 必须将 RequestInput 根元素映射到方法输入定义的 Rest_Consumer_input 组的子元素。
 - 可以将多个 String、Bigint 或 Integer 输入端口映射到**方法输入**区域中的键以创建复合键。为复合键单击**位置**字段时，可以重新对输入端口进行排序，或者删除其中一个端口。
 - 如果 Web 服务生成 JSON 文档，应确保 xmlRoot 是响应层次结构中的第一个节点。如果 xmlRoot 不是具有 JSON 响应的 Web 服务的第一个节点，则可能会显示空值。

将输入端口映射到方法输入

显示转换输入映射时，可以定义输入组，定义输入端口，以及将输入端口映射到方法输入元素。

1. 打开 REST Web 服务使用者转换。
2. 在**端口**视图中，显示输入映射。
3. 为根输入组定义主键。
4. 要将输入组或端口添加到**端口**区域，请使用以下方法之一：

方法	说明
拖动元素。	将一个组或一个子元素从 方法输入 区域拖动到 端口 区域的一个空列中。如果将组拖动到 端口 区域，Developer tool 会添加一个不包含端口的组。
手动添加组或端口。	要添加组，请单击 新建 按钮旁边的箭头，然后单击 组 。要添加端口，请单击 新建 按钮旁边的箭头，然后单击 字段 。
从其他转换中拖动端口。	在编辑器中，将端口从其他转换拖动到 REST Web 服务使用者转换。

方法	说明
复制端口。	从其他转换中选择端口，并将其复制到 端口 区域。要复制端口，可以使用键盘快捷键，或者使用 Developer tool 中的 复制 和 粘贴 按钮。
选择 映射第一级层次结构 。	Developer tool 会将方法输入第一级中的元素映射到输入端口和组。Developer tool 还会创建输入端口和组以执行映射。

- 如果手动创建端口或从其他转换复制端口，请单击**方法输入**区域中的**位置**列，然后从列表中选择端口。
- 要将输入端口映射为复合键，请使用以下方法之一：

方法	说明
拖动输入端口。	选择两个或多个输入端口，然后将它们拖动到方法输入层次结构的键中。
从 选择位置 对话框中选择输入端口。	在方法输入层次结构中单击键的 位置 列，然后选择输入端口。

- 要清除元素的位置，请使用以下方法之一：

方法	说明
单击 清除 。	在 方法输入 区域中选择一个或多个元素，然后单击 清除 。
删除将端口连接到元素的行。	在方法输入中选择将输入端口连接到元素的一个或多个行，然后按 删除 。

REST Web 服务使用者转换输出映射

查看转换端口时，系统将显示输出映射，从中可以查看方法输出层次结构。显示输出映射时，可以定义输出组，定义输出端口，并将方法输出元素映射到输出端口。

输出映射包含以下区域：

方法输出

方法输出区域显示了 Web 服务返回给 REST Web 服务使用者转换的响应消息中的元素。如果使用架构对象创建转换，该架构对象将定义方法输出层次结构。

端口

可在**端口**区域中创建转换输出组和端口。

创建输出端口后，将**方法输出**区域中的元素映射到**端口**区域中的端口。将方法输出中的元素映射到输出端口时，元素的位置将显示在**端口**区域的**位置**列中。

选择映射输出层次结构的第一级时，Developer 工具会将方法输出的第一级中的元素映射到输出端口。开发程序工具还会创建端口以执行映射。如果层次结构的第一级中包含的多次出现父元素含有一个或多个多次出现子元素，则 Developer 工具不映射第一级层次结构。

可以选择在层次结构中显示输出端口。每个子组都显示在父组下方。还可以选择查看将方法输出中的元素连接到输出端口的行。

如果从存储库中删除了关联的架构对象，则 Developer 工具会将方法元素的位置保留在输出映射中。显示输出映射时，**端口**区域仍将在输出端口的**位置**列中显示方法元素的位置。如果将其他架构与转换关联，则 Developer 工具会检查是否每个位置都有效。如果方法元素位置不再有效，则 Developer 工具将在输出映射的**端口**区域中清除该位置。

将元素映射到输出端口的规则和准则

将方法输出层次结构中的元素映射到输出端口时，请查看以下规则：

- 方法输出元素和输出端口的数据类型必须兼容。
- 不能将一个元素映射到一个组中的多个输出端口。
- 除非端口是传递端口，否则每个输出端口都必须具有有效位置。
- 如果要将多次出现的子元素拖动到空的输出端口，则必须将该组与其他输出组关联。选择组后，开发程序工具将创建用于关联组的键。
- 将多次出现的元素拖动到包含父元素的组时，可以配置要包含的子元素的出现次数。也可以将父组替换为转换输出中多次出现的子组。
- 如果 Web 服务生成 JSON 文档，应确保 xmlRoot 是响应层次结构中的第一个节点。如果 xmlRoot 不是具有 JSON 响应的 Web 服务的第一个节点，则输出端口中可能会显示空值。

自定义视图选项

可以更改方法输出层次结构，以便在**方法输出**区域中显示 cookie 端口、传递端口和键。还可以显示用于定义元素排序方式的分组结构。

在**方法输出**区域中单击**自定义视图**。启用以下任意选项：

“序列”、“选项”和“全部”

显示一行，指示元素定义是“序列”、“选项”还是“全部”。

序列组中的元素顺序必须是在层次结构中指定的顺序。

选项组中至少有一个元素必须显示在响应消息中。

“全部”组中的元素必须全部包含在响应消息中。

键

在**方法输出**区域中查看键。**方法输出**区域包含每个组的键。可以在**端口**区域中将键添加到输出端口。

传递端口

方法输出区域显示传递端口。传递端口是指通过转换传递数据而不更改数据的端口。可以将方法输出中的传递端口映射到任何 REST Web 服务使用者转换输出组。传递端口会一次性接收数据，因此，该端口处于响应消息中的根级别。

将方法输出映射到输出端口

显示转换输出映射时，可以定义输出组，定义输出端口，并将方法输出元素映射到输出端口。

1. 打开 REST Web 服务使用者转换。
2. 在**端口**视图中，显示输出映射。

3. 要将输出组或端口添加到端口区域，请使用以下方法之一：

方法	说明
拖动元素。	将方法输出区域中的一个组或一个子元素拖动到端口区域的一个空列中。如果将组拖动到端口区域，Developer tool 会添加一个不包含端口的组。
手动添加组或端口。	要添加组，请单击新建按钮旁边的箭头，然后单击组。要添加端口，请单击新建按钮旁边的箭头，然后单击字段。
从其他转换中拖动端口。	在编辑器中，将端口从其他转换拖动到 REST Web 服务使用者转换。
复制端口。	从其他转换中选择端口，并将其复制到端口区域。要复制端口，可以使用键盘快捷键，或者使用 Developer tool 中的复制和粘贴按钮。

4. 如果手动创建端口或从其他转换复制端口，请单击端口区域中的位置列，然后从列表中选择元素。

5. 要清除端口的位置，请使用以下方法之一：

方法	说明
单击清除。	在端口区域中选择一个或多个端口，然后单击清除。
删除将元素连接到端口的行。	在方法输出中选择将元素连接到输出端口的一个或多个行，然后按删除。

REST Web 服务使用者转换高级属性

配置有助于确定数据集成服务如何为 REST Web 服务使用者转换处理数据的属性。

在高级选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。可以选择精简、普通、详细初始化或详细数据。默认值为“普通”。

连接

标识连接到 Web 服务的 HTTP 连接对象。在 Developer tool 中创建和编辑 HTTP 连接。配置 HTTP 连接时，应配置基础 URL、Web 服务所需的安全类型和连接超时时限。

REST Web 服务使用者转换使用 URL 连接到 Web 服务。可以在转换属性或 HTTP 连接中定义 URL。

在以下情况下配置 HTTP 连接：

- 不使用 URL 输入端口。
- Web 服务需要进行 HTTP 身份验证或使用 SSL 证书。
- 您希望更改默认连接超时时限。

XML 架构验证

在运行时验证响应消息。选择 XML 无效错误或不验证。

已排序输入

允许数据集成服务在未处理所有输入数据的情况下生成输出。当输入数据按 XML 输入层次结构中的键排序时，请启用已排序输入。

URL

REST Web 服务的基础 URL。HTTP 连接中的基础 URL 将替代该值。

格式

Web 服务响应的格式。选择 **XML** 或 **JSON**，具体视 Web 服务响应而定。

REST Web 服务使用者转换创建

可以创建可重用或不可重用的 REST Web 服务使用者转换。可重用转换可存在于多个映射中。不可重用转换存在于单个映射内。

创建 REST Web 服务使用者转换时，可以手动定义元素和 XML 层次结构，或者可以从架构对象导入元素和层次结构。架构对象可以是 XML 文件或文本文件。

创建 REST Web 服务使用者转换

创建 REST Web 服务使用者转换时，选择方法并基于所选方法定义方法输入和方法输出。

1. 要创建 REST Web 服务使用者转换，请使用以下方法之一：

方法	说明
可重用	在“对象浏览器”视图中选择项目或文件夹。单击 文件 > 新建 > 转换 。选择 REST Web 服务使用者转换，然后单击 下一步 。
不可重用	在映射或 Mapplet 中，将 REST Web 服务使用者转换从转换选项板拖动到映射或 Mapplet 编辑器中。

2. 输入转换名称，并选择位置和 HTTP 方法。

3. 单击**下一步**。

4. 要定义方法输入，请使用以下方法之一：

方法	说明
创建为空	手动定义 XML 元素和层次结构。
从架构对象中的元素创建	从架构对象导入 XML 元素和层次结构。

方法输入定义区域显示转换输入组和输入端口。**输入映射**区域显示请求消息层次结构。

5. 定义输入组和输入端口，并将输入端口映射到输入元素。

6. 单击**下一步**。

7. 要定义方法输出，请选择**创建为空**或**从架构对象中的元素创建**。

方法输出定义区域显示转换输入组和输入端口。**输出映射**区域显示请求消息层次结构。

8. 定义输出组和输出端口，并将元素映射到输出端口。

9. 单击**完成**。

第 13 章

管理 REST Web 服务和 SOAP Web 服务

本章包括以下主题：

- [Web 服务管理概览, 142](#)
- [Web 服务属性配置, 142](#)
- [Web 服务安全管理, 146](#)
- [Web 服务日志, 150](#)
- [Web 服务监视, 150](#)

Web 服务管理概览

可以在 Administrator 工具中管理 REST Web 服务或 SOAP Web 服务。可以配置 Web 服务安全，配置 Web 服务，查看 Web 服务日志以及监视 Web 服务请求。您必须具有相应特权才能执行这些任务。

将 Web 服务部署到数据集成服务之后，可以执行以下任务：

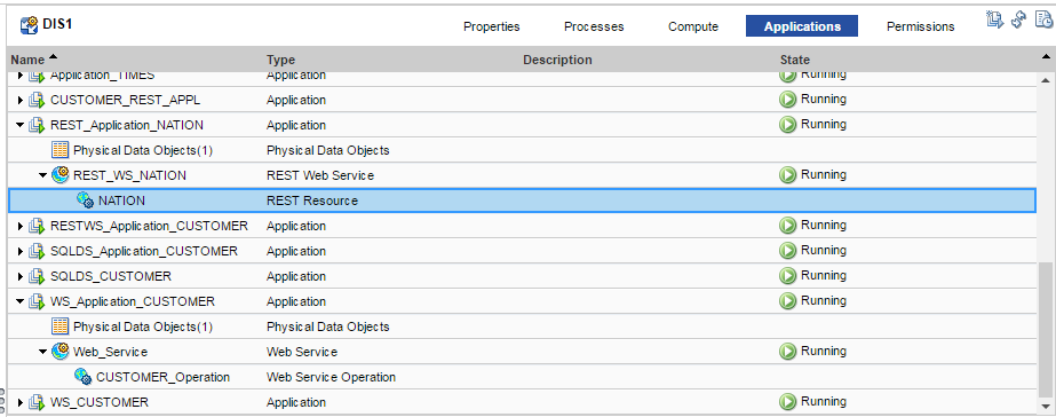
- 配置 Web 服务安全。启用 Web 服务安全并分配对 Web 服务的权限。
- 配置 Web 服务属性和 Web 服务操作属性。
- 查看 Web 服务日志。查看 Web 服务的数据集成服务日志。查看 Web 服务运行时日志目录中的 Web 服务运行时日志。
- 监视 Web 服务。使用 Administrator 工具或 Monitoring 工具监视 Web 服务请求。

Web 服务属性配置

在 Administrator 工具中，可以配置部署到数据集成服务的每个 Web 服务的 Web 服务属性。

在 Administrator 工具中，可以在数据集成服务的**应用程序**视图中编辑 Web 服务的属性。展开应用程序名称并选择一个 Web 服务或 REST Web 服务。属性将显示在**属性**视图中。

下图显示了数据集成服务的**应用程序**视图：



还可以编辑 SOAP Web 服务操作或 REST Web 服务资源的属性。选择操作名称或资源名称可查看属性。

Web 服务属性

REST Web 服务和 SOAP Web 服务属性包括只读常规属性和数据集成服务运行 Web 服务时使用的属性。

在“应用程序”视图的顶部面板中展开 Web 服务或 REST Web 服务时，可以访问 Web 服务中的 Web 服务操作或资源。

“应用程序”视图会显示 Web 服务、Web 服务操作或 Web 服务资源的只读常规属性。视图中显示的属性取决于对象类型。

下表介绍了每种类型的 Web 服务和 Web 服务操作或资源的只读常规属性：

属性	说明
名称	选定对象的名称。对所有对象显示。
说明	选定对象的简短说明。对所有对象显示。
类型	选定对象的类型。所有对象类型均会显示此属性。
位置	选定对象的位置。这包括域和数据集成服务名称。对所有对象显示。
URL	用于连接到 Web 服务的 URL。对 Web 服务显示。

下表介绍了 Web 服务的可配置的 Web 服务属性：

属性	说明
启动类型	决定允许 Web 服务在应用程序启动时运行还是在您启动 Web 服务时运行。
跟踪级别	<p>写入到运行时 Web 服务日志中的错误消息级别。选择以下消息级别之一：</p> <ul style="list-style-type: none"> - 关闭。DTM 进程不将消息写入 Web 服务运行时日志。 - 严重。“严重”消息包括可能导致 Web 服务停止运行的错误。 - 警告。“警告”消息包括可恢复的故障或警告。DTM 进程将“警告”和“严重”消息写入到 Web 服务运行时日志中。 - 信息。“信息”消息包括 Web 服务状态消息。DTM 进程将“信息”、“警告”和“严重”消息写入到 Web 服务运行时日志中。 - 细致。“细致”消息包括 Web 服务请求的数据处理错误。DTM 进程将“细致”、“信息”、“警告”和“严重”消息写入到 Web 服务运行时日志中。 - 最细致。“最细致”消息用于调试。DTM 进程将“最细致”、“细致”、“信息”、“警告”和“严重”消息写入到 Web 服务运行时日志中。 - 全部。DTM 进程将“最细致”、“细致”、“信息”、“警告”和“严重”消息写入到 Web 服务运行时日志中。 <p>默认值为“信息”。</p>
请求超时	数据集成服务在 Web 服务请求超时之前运行操作映射的最长时间（单位为毫秒）。默认值为 3,600,000。
并发请求数上限	一个 Web 服务一次可以处理的最大请求数。默认值为 10。
排序顺序	数据集成服务在 Unicode 模式下运行时用于对数据进行排序和比较的排序顺序。
启用传输层安全	指示 Web 服务必须使用 HTTPS。如果数据集成服务未配置为使用 HTTPS，则 Web 服务将不会启动。

下表包含 REST Web 服务特有的属性：

属性	说明
是否需要进行身份验证	启用 REST Web 服务的基本身份验证。基本身份验证需要 Web 服务请求的用户名和密码。默认为已禁用。
输入精度	数据集成服务在请求消息中解析的最大字符数。请求消息超出输入精度时，Web 服务请求会失败。默认值为 10,000。
输出精度	数据集成服务为响应消息生成的最大字符数。响应消息超出输出精度时，数据集成服务会将其截断。默认值为 3,000。

下表包含 SOAP Web 服务特有的属性：

属性	说明
启用 WS-Security	启用数据集成服务，以验证用户凭据以及验证用户是否具有运行每个 Web 服务操作的权限。仅限 SOAP Web 服务。
优化级别	数据集成服务应用到对象的优化器级别。请输入与您要配置的优化器级别关联的数值。您可以输入以下数值之一： <ul style="list-style-type: none">- 0. 数据集成服务不应用优化。- 1. 数据集成服务应用早期计划优化方法。- 2. 数据集成服务应用早期计划、早期选择、推入和谓词优化方法。- 3. 数据集成服务应用基于成本、早期计划、早期选择、推入、谓词和半联接优化方法。
DTM 保持活动的时间	DTM 实例在完成最后一个请求之后保持打开状态的毫秒数。针对相同的操作发出的 Web 服务请求可以重用处于打开状态的实例。当处理请求所需的时间少于 DTM 实例的初始化时间时，使用“保持活动的时间”属性来提高性能。如果请求失败，DTM 实例将终止。 必须为整数。负整数值表示使用了数据集成服务的 DTM 保持活动的时间。0 表示数据集成服务不将 DTM 实例保留在内存中。默认值为 -1。
SOAP 输出精度	数据集成服务为响应消息生成的最大字符数。当响应消息超出 SOAP 输出精度时，数据集成服务会将其截断。默认值为 200,000。
SOAP 输入精度	数据集成服务在请求消息中解析的最大字符数。Web 服务请求文件在请求消息超出 SOAP 输入精度时失败。默认值为 200,000。

Web 服务操作和资源属性

配置数据集成服务运行 Web 服务操作或 Web 服务资源时所使用的设置。

下表介绍了 SOAP Web 服务操作或 REST Web 服务资源的可配置属性：

属性	说明
结果集缓存到期时间段	结果集缓存可供使用的毫秒数。如果设置为 -1，缓存将永不过期。如果设置为 0，将禁用结果集缓存。对到期时间段所做的更改不会应用于现有缓存。如果您希望所有缓存使用相同的到期时间段，请在更改到期时间段后清除结果集缓存。默认值为 0。

Web 服务结果集缓存

结果集缓存使数据集成服务能够使用 Web 服务请求的已缓存结果。在短时间内运行相同查询的用户可能希望使用结果集缓存来缩短相同查询的运行时间。

配置结果集缓存时，数据集成服务将缓存与每个 Web 服务请求关联的 DTM 进程的结果。数据集成服务将在您配置的到期时间段内缓存结果。外部客户端在缓存过期之前运行相同的查询或请求时，数据集成服务将返回已缓存的结果。如果缓存不存在或已过期，数据集成服务会启动 DTM 实例来处理该请求。

当缓存中的数据量超出最大缓存大小内存时，数据集成服务会将结果集存储在一个加密的缓存文件中，该文件位于 `<Informatica_install_dir>/tomcat/bin/disTemp/<Service_Name>/<Node_Name>/`。

Web 服务使用 WS-Security 时，数据集成服务按用户存储 Web 服务的结果集缓存。数据集成服务会按在 Web 服务请求的用户名标志中提供的用户名存储缓存。数据集成服务按用户缓存结果时，仅将已缓存的结果返回给发送 Web 服务请求的用户。

要在 Administrator 工具中配置结果集缓存，请执行以下步骤：

1. 在数据集成服务进程属性中配置结果集缓存属性。

下表介绍了结果集缓存属性：

属性	说明
总磁盘大小上限	结果集缓存存储总量允许的最大字节数。默认值为零。
每个缓存的内存大小上限	内存中分配给单个结果集缓存实例的最大字节数。默认值为零。
总内存大小上限	内存中分配给总结果集缓存存储的最大字节数。默认值为零。
缓存数上限	数据集成服务允许的结果集缓存最大实例数。默认值为零。

2. 在 SOAP Web 服务操作属性或 REST Web 服务资源属性中配置缓存到期时间段。
如果结果集可供使用的毫秒数，则结果集缓存到期时间段如果设置为 -1，缓存将永不过期。如果设置为零，将禁用结果集缓存。对到期时间段进行的更改不会应用到现有缓存。要将所有缓存设置为使用相同的到期时间段，请在更改到期时间段后清除结果集缓存。默认值为零。
3. 要使数据集成服务按用户缓存结果，请在 Web 服务属性中启用 WS-Security。

要在将 Web 服务操作配置为缓存结果集时对 Web 服务请求禁用结果集缓存，请在 SOAP 请求的 HTTP 表头中添加以下语法：`WebServiceOptions.disableResultSetCache=true`

Web 服务安全管理

HTTP 客户端筛选器、传输层安全和消息层安全可以为 Web 服务提供安全数据传输和授权数据访问。配置消息层安全时，数据集成服务可以将凭据传递给连接。

可以为 REST Web 服务配置以下安全选项：

是否需要进行身份验证

启用 REST Web 服务的基本身份验证。基本身份验证要求每个 Web 服务请求包括域的用户名和密码。启用 Administrator 工具中的数据集成服务的属性。单击 **应用程序 > ApplicationNameREST Web 服务 > isAuthenticationRequired**。需要身份验证时，每个 GET 请求都需要用户名和密码，然后 REST Web 服务才能返回响应。默认为已禁用。

可以为 SOAP Web 服务配置以下安全选项：

HTTP 客户端筛选器

如果希望数据集成服务根据 Web 服务客户端的主机名或 IP 地址接受请求，请使用 Administrator 工具配置 HTTP 客户端筛选器。默认情况下，在任何计算机上运行的 Web 服务客户端均可发送请求。

消息层安全

如果希望数据集成服务对 SOAP 请求中的用户凭据进行身份验证，可使用 Administrator 工具启用 WS-Security 并配置 Web 服务权限。数据集成服务可以验证作为 SOAP 请求中的用户名标志提供的用户凭据。如果用户名标志无效，数据集成服务会拒绝该请求并向 Web 服务客户端发送系统定义的故障。如果用户没有执行 Web 服务操作的权限，数据集成服务会拒绝该请求并向 Web 服务客户端发送系统定义的故障。

传输层安全 (TLS)

如果希望 Web 服务和 Web 服务客户端使用 HTTPS URL 通信，请使用 Administrator 工具为 Web 服务启用传输层安全 (TLS)。Web 服务运行所在的数据集成服务也必须使用 HTTPS 协议。HTTPS URL 使用 SSL 为 Web 服务与 Web 服务客户端之间的数据传输提供安全连接。

传递安全

如果操作映射需要连接凭据，数据集成服务可以把 SOAP 请求中用户名标志中的凭据传递给该连接。要将数据集成服务配置为向连接传递凭据，可使用 Administrator 工具将数据集成服务配置为对连接使用传递安全，并为 Web 服务启用 WS-Security。

注意: 如果用户名标志包含哈希或摘要式密码，则无法使用传递安全。

Web 服务权限

权限控制当 Web 服务要求进行用户身份验证时用户对 REST Web 服务或 SOAP Web 服务具有的访问级别。可以使用 Administrator 工具来配置 REST Web 服务或 SOAP Web 服务的权限。还可以设置资源或操作的权限。

要分配权限，请从数据集成服务的**应用程序**视图中选择 Web 服务、资源或操作。单击**用户权限**或**组权限**。

管理员可将 Web 服务权限分配给以下类型的用户和组：

- Web 服务使用者。向 Web 服务发送请求并接收来自 Web 服务的响应的本地域用户。该用户必须具有对 Web 服务的执行权限。
- Web 服务管理员。可以执行以下操作的用户：登录到 Administrator 工具，编辑 Web 服务属性以及向其他用户授予权限。
- Web 服务操作员。可以执行以下操作的用户：登录到 Administrator 工具，监视 Web 服务以及启动或停止 Web 服务。

可以向用户和组分配以下权限：

- 授予权限。用户可以使用 Administrator 工具或使用 *infacmd* 命令行程序管理对 Web 服务对象的权限。
- 执行权限。用户可以发送 Web 服务请求并接收 Web 服务响应。

下表介绍了每个 SOAP Web 服务对象的权限：

对象	授予权限	执行权限
SOAP Web 服务	授予和撤销对 Web 服务以及 Web 服务内的所有 Web 服务操作的权限。	发送 Web 服务请求以及接收来自 Web 服务内的所有 Web 服务操作的 Web 服务响应。
SOAP Web 服务操作	授予、撤销和拒绝对 Web 服务操作的权限。	发送 Web 服务请求以及接收来自 Web 服务操作的 Web 服务响应。

下表介绍了每个 REST Web 服务对象的权限：

对象	授予权限	执行权限
REST Web 服务	授予和撤销对 REST Web 服务以及 Web 服务内的所有 Web 服务资源的权限。	发送 Web 服务请求以及接收来自 REST Web 服务中所有 Web 服务资源的 Web 服务响应。
REST 资源	授予、撤销和拒绝对 REST Web 服务资源的权限。	发送 Web 服务请求以及接收来自 REST Web 服务资源的 Web 服务响应。

SOAP 请求中的用户名标志

Web 服务要求用户身份验证时，Web 服务客户端必须在 SOAP 请求中提供用户名标志表头。当 Web 服务不要求用户身份验证时，数据集成服务将忽略 SOAP 请求中提供的用户名标志表头。

SOAP 请求中的用户名标志元素可以包含以下任一密码类型：

- 纯文本
- 哈希
- 摘要式

注意：当用户名标志包含哈希或摘要式密码时，不能使用 LDAP 身份验证。

在 UsernameToken 元素的密码元素中添加用户密码。密码元素包含一个“类型”属性，用于指示所用的密码安全类型。

纯文本密码

如果无需加密用户密码，请将纯文本密码包含在 SOAP 请求的用户名标志表头中。数据集成服务可以处理 UsernameToken 元素中的纯文本密码。

如果密码为纯文本，则 UsernameToken 元素包含以下子元素：

用户名元素

包含本地安全域或任何 LDAP 安全域中的用户名。默认安全域为本地安全域。如果用户名属于本地安全域，Username 不需要安全域的名称。如果用户名属于 LDAP 安全域，用户名之前必须有安全域名称和斜杠 (/)。

密码元素

包含纯文本形式的密码。将密码元素的“类型”属性设置为“PasswordText”。

下面的示例 SOAP 表头中包含使用纯文本密码的 UsernameToken 元素示例：

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">Administrator</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

哈希密码

必须加密用户密码时，可在 SOAP 请求的用户名标志表头中包含哈希密码。数据集成服务可以处理 UsernameToken 元素中的哈希密码。

使用哈希密码时，UsernameToken 元素可以包含以下子元素：

用户名元素

包含一个位于本地安全域中的用户名。

密码元素

包含一个哈希密码。必须使用 MD5 或 SHA-1 哈希函数对该密码进行哈希处理，并将其转换为 Base64 编码。将密码元素的“类型”属性设置为“PasswordText”。

下面的示例 SOAP 表头中包含使用哈希密码的 UsernameToken 元素示例：

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">Ntm58Cxf7SB0QAz30lsTq1nv-D7</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

摘要式密码

当用户密码为使用临时值和时间戳进行散列计算的加密密码时，可在 SOAP 请求的用户名标志表头中添加摘要式密码。数据集成服务可以处理 UsernameToken 元素中的摘要式密码。

使用摘要式密码时，UsernameToken 元素可以包含以下子元素：

用户名元素

包含一个位于本地安全域中的用户名。

密码元素

包含一个摘要式密码。密码是对由 Nonce 元素的临时值和 Created 元素中的时间戳串联的密码进行散列计算得出的值。密码必须使用 SHA-1 散列函数进行散列计算，且必须编码为 Base64。为保证摘要式密码安全，请将密码元素的“类型”属性设置为“PasswordDigest”。

Nonce 元素

包含一个临时值，而且这是一个只能使用一次的随机值。默认情况下，该值的有效时间为 300 秒，自申请创建之时算起（可根据 Created 元素中的值判断创建时间）。客户端应用程序必须在临时值的有效时间内发送请求。例如，Created 值显示请求创建时间为 10:00 a.m.，则请求的有效时间为从 10:00 a.m. 至 10:05 a.m.。如果客户端应用程序在 10:00 a.m. 之前或 10:05 a.m. 之后将请求发送至 Web 服务，则请求和临时值无效，请求将失败。

Created 元素

包含一个指示请求创建时间的时间戳值。该时间戳采用 UTC 格式 yyyy-MM-dd'T'HH:mm:ss.SSS'Z'。例如：2008-08-11T18:06:32.425Z。

摘要式密码使用标准 OASIS 密码摘要算法：

Password_Digest = Base64 (SHA-1 (nonce + created + password))

您可以使用任何工具生成临时值、时间戳以及摘要式密码。

下面的示例 SOAP 表头中包含使用摘要式密码的 UsernameToken 元素示例：

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">Ntm58Cxf7SB0QAz30lsTq1nv-D7</wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">zwELHdoAzNjQQ9xzLIwFZA==</wsse:Nonce>
      <wsu:Created>2010-10-15T20:56:18.633Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

Web 服务日志

您可以在数据集成服务日志和 Web 服务运行时日志中查看 Web 服务日志。

数据集成服务日志中可能会包含 Web 服务状态变化或妨碍 Web 服务运行的错误等这类 Web 服务消息。在 Administrator 工具的**日志**选项卡上查看数据集成服务日志。

当数据集成服务的 DTM 进程运行 Web 服务操作映射时，有关该 Web 服务操作映射的消息将显示在 Web 服务运行时日志中。数据集成服务的 DTM 进程会为每个 Web 服务请求创建一个日志文件。可以在数据集成服务进程日志位置中的 ws 目录中查看 Web 服务运行时日志。默认情况下，您可以在以下目录查看 Web 服务运行时日志：
<InformaticaInstallationDir>/tomcat/bin/disLogs/ws

Web 服务跟踪级别

在 Administrator 工具中，配置**跟踪级别**属性，以指出 DTM 进程写入到 Web 服务运行时日志中的消息的跟踪级别。可以配置 Web 服务请求中的 HTTP 表头，以替代为 Web 服务设置的跟踪级别。

每个 Web 服务请求均基于 Web 服务跟踪级别配置生成运行时日志。要替代 Web 服务请求的 Web 服务跟踪级别，请在 Web 服务 SOAP 请求的 HTTP 表头中加入以下条目：WebServiceOptions.traceLevel= <trace_level>。例如，要使用 HTTP 表头将跟踪级别设置为“细致”，请使用以下文本：WebServiceOptions.traceLevel= fine

Web 服务监视

要监视 REST Web 服务或 SOAP Web 服务，请查看其属性、运行时统计信息和运行时报告以及有关每个 Web 服务请求的信息。

可以在以下位置监视 Web 服务：

- 监视工具。在 Developer tool 中，单击**进度**视图中的**菜单**按钮，并选择**监视作业**。选择运行 Web 服务的数据集成服务，然后单击**确定**。此时将打开 Monitoring 工具。
- Administrator 工具。要在 Administrator 工具中监视 Web 服务，请单击**监视**选项卡。

监视 Web 服务时，可以查看该服务的摘要统计信息或执行统计信息。

摘要统计信息视图显示与 Web 服务分布和状态相关的图形信息。如果选择一个时间范围并展开**请求和连接**面板，“摘要统计信息”将显示该时间范围的 Web 服务请求。可以查看请求的详细信息或摘要。查看图形分布或表格式摘要。“摘要统计信息”视图会使用存储在模型存储库中的数据显示统计信息。必须先在“监视配置”中配置模型存储库，然后才能查看摘要统计信息。

执行统计信息视图显示与应用程序中部署的特定 Web 服务相关的信息。可以查看有关 SOAP Web 服务或 REST Web 服务的以下统计信息：

- 请求总数。
- 已完成的请求。
- 已中止的请求。在中止模式下再次应用或禁用了数据集成服务时被中止的请求。
- 已失败的请求。

要查看 Web 服务的属性，请在导航器中展开一个应用程序并选择 **Web Services** 文件夹或 **Rest Services** 文件夹。此时内容面板中将显示 Web 服务列表。内容面板将显示有关每个 Web 服务的属性，如名称、说明和状态。

选择 Web 服务时，详细信息面板将显示以下视图：

- 属性视图。属性视图显示 Web 服务的常规属性和运行时统计信息。

- 报告视图。报告视图显示有关选定 Web 服务的监视报告。
- 操作视图。操作视图显示 Web 服务中包含的每个操作的名称和说明。该视图还显示关于每个操作的属性、请求和报告。
- 请求视图。请求视图显示有关每个 Web 服务请求的属性，如请求 ID、用户名、状态、开始时间、已用时间和结束时间。可以筛选请求列表。

Web 服务的“属性”视图

属性视图显示 Web 服务的常规属性和运行时统计信息。

在**属性**视图的内容面板中选择 Web 服务后，可以查看相应的常规属性和监视统计信息。

Web 服务的常规属性

可以查看 Web 服务的常规属性，如对象的名称和类型。

Web 服务的统计信息

可以查看特定时间段内 Web 服务请求的运行时统计信息。**统计信息**部分显示已完成的 Web 服务请求数、失败的 Web 服务请求数和总 Web 服务请求数。

Web 服务的“报告”视图

报告视图显示有关选定 Web 服务的监视报告。

监视 Web 服务时，**报告**视图会显示有关 Web 服务的报告。例如，您可以查看“最活跃的 Web 服务客户端 IP”报告来确定特定时间段内接收到最多 Web 服务请求的 IP 地址。

REST 或 SOAP Web 服务的“操作”视图

操作视图显示 Web 服务中包含的每个操作或资源的名称和说明。该视图还显示关于每个操作的属性、请求和报告。

在内容面板中选择 Web 服务操作时，详细信息面板显示**属性**视图、**请求**视图和**报告**视图。

“属性”视图

属性视图显示有关选定的 Web 服务操作或资源的常规属性和统计信息。常规属性包括操作名称或资源名称和对象的类型。该视图还显示特殊时间段内的 Web 服务操作的相关统计信息。统计信息包括 Web 服务请求的已完成数量、失败数量和总数。

“请求”视图

请求视图显示每个 Web 服务操作的属性，如请求 ID、用户名、状态、开始时间、已用时间和结束时间。可以筛选请求列表。还可以查看选定 Web 服务请求的日志。

SOAP Web 服务的“报告”视图

报告视图显示有关 SOAP Web 服务操作的报告。

Web 服务的“请求”视图

请求视图显示有关每个 Web 服务请求的属性，如请求 ID、用户名、状态、开始时间、已用时间和结束时间。可以筛选请求列表。

在内容面板中选择 Web 服务请求后，可以在详细信息面板中查看有关该请求的日志。详细信息面板显示有关选定 Web 服务请求的常规属性和统计信息。统计信息包括 Web 服务请求的已完成数量、失败数量和总数。

还可以从**请求**视图中止 Web 服务请求。要终止 Web 服务请求，可在内容面板中选择工作流请求并单击**操作 > 中止选定请求**。

附录 A

数据类型兼容性

本附录包括以下主题：

- [数据类型引用概览, 153](#)
- [XML 和转换数据类型, 153](#)
- [小数, 155](#)

数据类型引用概览

创建映射时，会创建一组指令，数据集成服务使用这组指令从源读取数据、转换数据并将数据写入目标。数据集成服务根据映射中的数据流（从映射中的第一个转换开始）和分配给映射中的每个端口的数据类型转换数据。

Developer 工具显示两种数据类型：

- 本地数据类型。特定于用作物理数据对象的关系表或平面文件。本地数据类型显示在物理数据对象列属性中。
- 转换数据类型。在转换中显示的一组数据类型。它们是基于 ANSI SQL-92 通用数据类型的内部数据类型，数据集成服务可使用这些数据类型在平台间移动数据。转换数据类型显示在映射中的所有转换中。

数据集成服务读取源数据时，它会先将本地数据类型转换为可比的转换数据类型，然后再转换数据。数据集成服务向目标写入时，它会将转换数据类型转换为可比的本地数据类型。

指定多字节字符集时，数据类型在数据库中分配额外空间，用以存储最多三个字节的字符。

XML 和转换数据类型

XML 数据类型映射到数据集成服务用来跨平台移动数据的转换数据类型。

数据集成服务支持 W3C 于 2001 年 5 月 2 日建议中指定的所有 XML 数据类型。但数据集成服务可能不支持整个 XML 值范围。有关 XML 数据类型的详细信息，请参阅 XML 数据类型的 W3C 规范，网址为：

<http://www.w3.org/TR/xmlschema-2>.

下表将 XML 数据类型与转换数据类型进行了比较：

数据类型	转换	范围
anyURI	String	1 至 104,857,600 个字符
base64Binary	Binary	1 到 104,857,600 个字节
boolean	String	1 至 104,857,600 个字符
byte	Integer	-2,147,483,648 到 2,147,483,647
date	Date/Time	公元 0001 年 1 月 1 日到公元 9999 年 12 月 31 日（精确到纳秒）
dateTime	Date/Time	公元 0001 年 1 月 1 日到公元 9999 年 12 月 31 日（精确到纳秒）
decimal	Decimal	精度为 1 到 28，小数位数为 0 到 28
双精度型	Double	精度为 15 位数字
duration	String	1 至 104,857,600 个字符
ENTITIES	String	1 至 104,857,600 个字符
ENTITY	String	1 至 104,857,600 个字符
float	Double	精度为 15 位数字
gDay	String	1 至 104,857,600 个字符
gMonth	String	1 至 104,857,600 个字符
gMonthDay	String	1 至 104,857,600 个字符
gYear	String	1 至 104,857,600 个字符
gYearMonth	String	1 至 104,857,600 个字符
hexBinary	Binary	1 到 104,857,600 个字节
ID	String	1 至 104,857,600 个字符
IDREF	String	1 至 104,857,600 个字符
IDREFS	String	1 至 104,857,600 个字符
int	Integer	-2,147,483,648 到 2,147,483,647
integer	Integer	-2,147,483,648 到 2,147,483,647
language	String	1 至 104,857,600 个字符
long	Bigint	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
名称	String	1 至 104,857,600 个字符
NCName	String	1 至 104,857,600 个字符

数据类型	转换	范围
negativeInteger	Integer	-2,147,483,648 到 2,147,483,647
NMTOKEN	String	1 至 104,857,600 个字符
NMTOKENS	String	1 至 104,857,600 个字符
nonNegativeInteger	Integer	-2,147,483,648 到 2,147,483,647
nonPositiveInteger	Integer	-2,147,483,648 到 2,147,483,647
normalizedString	String	1 至 104,857,600 个字符
NOTATION	String	1 至 104,857,600 个字符
positiveInteger	Integer	-2,147,483,648 到 2,147,483,647
QName	String	1 至 104,857,600 个字符
short	Integer	-2,147,483,648 到 2,147,483,647
string	String	1 至 104,857,600 个字符
time	Date/Time	公元 0001 年 1 月 1 日到公元 9999 年 12 月 31 日（精确到纳秒）
token	String	1 至 104,857,600 个字符
unsignedByte	Integer	-2,147,483,648 到 2,147,483,647
unsignedInt	Integer	-2,147,483,648 到 2,147,483,647
unsignedLong	Bigint	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
unsignedShort	Integer	-2,147,483,648 到 2,147,483,647

小数

如果 Web 服务操作映射包含精度超过 28 位数的小数数据类型的输入或输出转换，则数据集成服务会将小数数据类型转换为双精度数据类型。

索引

A

- anyAttribute 属性
 - Web 服务使用者转换 [82](#), [85](#)
- anyType
 - 映射端口 [75](#)
- anyType 元素
 - 解析 [62](#)
 - Web 服务使用者转换 [82](#), [85](#)
- attributeFormDefault
 - 架构对象 [28](#)
- 安全
 - Web 服务安全 [146](#)
 - 用户名标志 [148](#)

B

- 绑定
 - WSDL 文件元素 [79](#)
- 表头
 - 创建 [43](#)
 - HTTP POST [43](#)
 - 说明 [15](#)
- 并发 Web 服务请求消息
 - 在 Web 服务使用者转换中启用 [87](#)

C

- 操作
 - 说明 [15](#)
 - WSDL 文件元素 [79](#)
- 操作故障
 - 说明 [15](#)
- 操作区域
 - Web 服务转换 [68](#)
- 操作输出
 - 说明 [15](#)
 - “操作输出”区域
 - 自定义 Web 服务使用者转换 [86](#)
- 操作输入
 - 说明 [15](#)
- 操作输入区域
 - 自定义 Web 服务使用者转换 [83](#)
- 成员类型
 - 架构对象 [26](#)
- 抽象属性
 - 架构对象 [25](#)
- 创建
 - 故障转换 [52](#)
- 传输层安全
 - REST Web 服务使用者转换 [79](#)
 - Web 服务使用者转换 [79](#)
- Cookie 身份验证
 - REST Web 服务使用者转换 [136](#)

- Cookie 身份验证 (续)
 - Web 服务使用者转换 [82](#)
- 操作映射
 - 测试 [57](#)
 - 故障转换 [51](#)
 - 输出转换 [48](#)
 - 输入转换 [46](#)
 - 概览 [45](#)
- 创建 REST Web 服务
 - 部署数据对象 [126](#)
 - 从数据对象创建 [123](#)

D

- 代码
 - Web 服务故障处理 [54](#)
- 动态 URL
 - Web 服务使用者转换 [82](#)
- 端点 URL
 - REST Web 服务使用者转换 [135](#)
 - Web 服务使用者转换 [82](#)
- 端口
 - 未规范化的 Web 服务输入 [73](#)
 - 映射至 SOAP 消息 [70](#)
- 端口选项卡
 - 输入转换 [47](#)
- “端口”选项卡
 - 故障转换 [51](#)
 - 输出转换 [49](#)

E

- elementFormDefault
 - 架构对象 [28](#)

F

- 复合键
 - REST Web 服务使用者转换 [137](#)
 - Web 服务使用者转换 [82](#)
- 服务
 - WSDL 文件元素 [79](#)
- 复杂类型
 - 高级属性 [27](#)
- 副作用
 - Web 服务使用者转换 [91](#)

G

- 高级属性
 - REST Web 服务使用者转换 [140](#)
 - Web 服务使用者转换 [87](#)

- “高级”选项卡
 - 输出转换 [49](#)
- 跟踪级别
 - Web 服务 [150](#)
- 固定值属性
 - 架构对象 [25](#)
- 故障代码
 - Web 服务故障处理 [54](#)
- 故障触发者
 - Web 服务故障处理 [54](#)
- 故障元素
 - SOAP 1.1 [54](#)
 - SOAP 1.2 [54](#)
- 故障转换
 - 创建 [52](#)
 - “端口”选项卡 [51](#)
 - 说明 [51](#)
 - 通用故障 [52](#)
 - 以 XML 形式映射到输出 [51](#)
 - 映射第一级层次结构 [51](#)
 - 映射规则和指导 [52](#)
 - 预定义故障 [52](#)
 - 在 Web 服务中配置 [53](#)
 - 自定义视图选项 [57](#)
- 故障字符串
 - Web 服务故障处理 [54](#)
- GZip
 - 压缩 SOAP 消息 [90](#)

H

- HTTP 表头
 - 添加到 Web 服务使用者转换 [81](#)
 - 添加到 REST Web 服务使用者转换 [136](#)
- HTTP 错误输出
 - 在 Web 服务使用者转换中启用 [87](#)
- HTTP 连接
 - REST Web 服务 [140](#)
- HTTP POST
 - 表头 [43](#)
- HTTP 响应代码
 - REST Web 服务使用者转换 [136](#)

I

- isAuthenticationRequired
 - REST Web 服务 [146](#)

J

- 架构对象
 - 元素高级属性 [26](#)
 - attributeFormDefault [28](#)
 - 编辑架构文件 [29](#)
 - 抽象属性 [25](#)
 - 导入 [29](#)
 - elementFormDefault [28](#)
 - 复杂元素 [27](#)
 - 复杂元素高级属性 [27](#)
 - 概览 [23](#)
 - 概览视图 [23](#)
 - 架构视图 [24](#)
 - 架构文件 [24](#)
 - 简单类型 [26](#)
 - 继承自属性 [27](#)

- 架构对象 (续)
 - 块属性 [26](#)
 - 命名空间 [24](#)
 - 设置默认编辑器 [31](#)
 - 特性属性 [28](#)
 - 同步 [29](#)
 - 文件位置 [28](#)
 - 由属性继承 [27](#)
 - 元素属性 [25](#)
 - 置换组 [26](#)
- 架构视图
 - 架构对象 [24](#)
 - 简单类型高级属性 [27](#)
 - REST Web 服务 [100](#)
- 架构文件
 - 编辑 [31](#)
 - 从架构对象中删除 [24](#)
 - 设置默认编辑器 [31](#)
 - 添加到架构对象 [24](#)
- 键
 - SOAP 消息层次结构 [69](#)
- 简单类型
 - 架构对象 [27](#)
- 简单类型视图
 - 架构对象 [26](#)
- 将故障视为错误
 - 在 Web 服务使用者转换中启用 [87](#)
- 基本属性
 - 架构对象 [26](#)
- 继承自属性
 - 架构对象 [27](#)
- 节点
 - Web 服务故障处理 [54](#)
- 角色
 - Web 服务故障处理 [54](#)
- 基本身份验证
 - REST Web 服务 [146](#)
- 结果集缓存
 - Web 服务操作属性 [145](#)
 - 配置 [145](#)

K

- 可空属性
 - 架构对象 [25](#)
- 块属性
 - 架构对象 [26](#)

L

- 联合元素
 - 解析 SOAP 消息 [65](#)
 - 说明 [77](#)
- 连接
 - REST Web 服务 [140](#)
 - Web 服务 [87](#)
- 列表元素
 - 解析 SOAP 消息 [65](#)
 - 说明 [76](#)

M

- 枚举属性
 - 架构对象 [25](#)

密码

纯文本 [148](#)

哈希 [148](#)

命名空间

更改生成的前缀 [24](#)

架构对象 [24](#)

模式属性

架构对象 [25](#)

P

派生类型

解析 SOAP 消息 [63](#)

Web 服务 [74](#)

派生类型元素

Web 服务使用者转换 [82](#), [85](#)

Q

Qname 元素

解析 SOAP 消息 [64](#)

全部组

在 REST Web 服务使用者转换中查看 [139](#)

在 Web 服务使用者转换中查看 [83](#), [86](#)

请求消息

REST Web 服务 [104](#)

R

任何元素

Web 服务使用者转换 [82](#), [85](#)

REST Web 服务使用者转换

安全 [79](#)

不可重用 [141](#)

参数端口 [135](#)

传递端口 [135](#)

创建 [141](#)

传输层安全 [79](#)

cookie 端口 [136](#)

代理服务器支持 [129](#)

Delete 方法 [134](#)

端口 [134](#)

概览 [129](#)

高级属性 [140](#)

跟踪级别 [140](#)

Get 方法 [132](#)

HTTP 方法 [132](#)

HTTP 表头端口 [136](#)

Internet 媒体类型 [140](#)

进程 [130](#)

可重用 [141](#)

连接属性 [140](#)

配置 [130](#)

Post 方法 [133](#)

Put 方法 [133](#)

RequestInput 端口 [134](#)

设置基本 URL [140](#)

输出 XML 端口 [136](#)

输出端口 [135](#)

输出映射 [138](#)

输出映射规则 [139](#)

输入端口 [135](#)

输入映射 [136](#)

输入映射规则 [137](#)

URL 端口 [135](#)

REST Web 服务使用者转换 (续)

响应代码端口 [136](#)

消息配置 [131](#)

XML 架构验证 [140](#)

映射输出 [129](#)

映射输出端口 [139](#)

映射输入 [129](#)

映射输入端口 [137](#)

映射元素至端口 [134](#)

已排序输入 [140](#)

自定义输出映射视图 [139](#)

资源标识 [131](#)

REST Web 服务

“数据查看器”视图 [107](#)

按资源键进行查询 [106](#)

创建 Web 服务 [109](#)

多次出现的数据 [104](#)

概览 [97](#), [105](#)

架构视图 [100](#)

进程 [98](#)

默认映射 [101](#)

请求消息格式 [104](#)

输出转换 [103](#)

同步资源 [100](#)

响应消息格式 [106](#)

资源键 [101](#)

资源映射 [101](#)

自定义映射 [102](#)

REST 和 SOAP

比较 [14](#)

如何

从数据对象创建 REST Web 服务 [123](#)

将数据对象部署为 REST Web 服务 [126](#)

S

筛选器端口

Web 服务使用者转换 [91](#)

身份验证

用户名标志 [148](#)

生成的键

Web 服务输出组 [61](#)

生成的前缀

为命名空间更改 [24](#)

输出映射

REST Web 服务使用者转换 [138](#)

REST Web 服务 [103](#)

Web 服务使用者转换 [85](#)

多个组 [104](#)

输出转换

“端口”选项卡 [49](#)

“高级”选项卡 [49](#)

说明 [48](#)

以 XML 形式映射到输出 [48](#)

映射第一级层次结构 [49](#)

映射规则和准则 [49](#)

在 Web 服务中配置 [49](#)

自定义视图选项 [57](#)

数据类型

概览 [153](#)

XML [153](#)

输入端口区域

生成 SOAP 消息 [67](#)

输入映射

REST Web 服务使用者转换 [136](#)

Web 服务使用者转换 [82](#)

输入转换

- 端口选项卡 [47](#)
- 说明 [46](#)
- 映射层次结构第一级 [47](#)
- 映射规则和指导 [47](#)
- 在 Web 服务中配置 [47](#)
- 自定义视图选项 [57](#)
- 作为 XML 映射到输入 [46](#)

SOAP 1.1

- 故障元素 [54](#)

SOAP 1.2

- 故障元素 [54](#)

SOAP 操作

- 在 Web 服务使用者转换中替代 [87](#)

SOAP 层次结构

- 与输入端口的关系 [68](#)

SOAP Web 服务

- 开发 [16](#)
- 创建概览 [34](#)

SOAP 消息

- 概览 [78](#)
- 键 [69](#)
- 将端口映射到联合元素 [77](#)
- 解析 anyType 元素 [62](#)
- 解析列表元素 [65](#)
- 解析选项元素 [65](#)
- 解析置换组 [65](#)
- 映射端口 [70](#)
- 映射多次出现的节点 [60](#)
- 映射多个输入端口 [71](#)
- 映射列表元素 [76](#)
- 映射选项元素 [76](#)
- 转换数据 [72](#)

SOAP 消息解析

- 联合元素 [65](#)
- 派生类型 [63](#)
- Qname 元素 [64](#)
- 说明 [59](#)
- 未规范化输出 [61](#), [62](#)
- 已转换的输出 [62](#)

SOAP 压缩

- Web 服务使用者转换 [90](#)

SOAP 和 REST

- 比较 [14](#)

数据访问方法

- REST Web 服务预览 [107](#)

T

特性属性

- 架构对象 [28](#)

替代 SOAP 操作

- Web 服务使用者转换 [87](#)

通用 SOAP 故障

- Web 服务使用者转换 [89](#)

通用故障

- Web 服务故障处理 [56](#)

通用故障输出

- 在 Web 服务使用者转换中启用 [87](#)

推入优化

- Web 服务使用者转换 [91](#)

同步资源

- REST Web 服务 [100](#)

U

UsernameToken

- 纯文本密码 [148](#)
- 哈希密码 [148](#)

W

Web 服务连接

- 概览 [87](#)

Web 服务使用者转换

- 安全 [79](#)
- 并发 Web 服务请求消息 [87](#)
- 操作 [79](#)
- 查看键 [83](#), [86](#)
- 创建 [93](#)
- 传输层安全 [79](#)
- Cookie 身份验证 [82](#)
- 错误处理 [89](#)
- 动态 Web 服务 URL [82](#)
- 动态 WS-Security 名称 [82](#)
- 端点 URL [82](#)
- 概览 [78](#)
- 高级属性 [87](#)
- 启用 HTTP 错误输出 [87](#)
- 启用通用故障输出 [87](#)
- 启用推入优化 [92](#)
- 筛选器优化 [91](#)
- 输出映射 [85](#)
- 输入映射 [82](#)
- SOAP 消息 [78](#)
- SOAP 压缩 [90](#)
- 添加 HTTP 表头 [81](#)
- 通用 SOAP 故障 [89](#)
- 推入优化 [91](#)
- 映射输出节点 [85](#)
- 映射输入端口 [82](#)
- 早期选择优化 [91](#)

Web 服务转换

- 位置列 [68](#)

Web 服务

- 部署 [16](#)
- 操作 [15](#)
- 创建表头 [43](#)
- 创建元素 [42](#)
- 创建预定义故障 [43](#)
- 从 WSDL 创建 [37](#)
- 概览视图 [35](#)
- 跟踪级别 [150](#)
- 故障处理 [54](#)
- 将端口映射至 anyTypes [75](#)
- 派生类型 [74](#)
- 配置故障转换 [53](#)
- 配置输出转换 [49](#)
- 配置输入转换 [47](#)
- 日志 [150](#)
- 输入转换 [46](#)
- SOAP 示例 [17](#)
- 通用故障 [56](#)
- WSDL [16](#)
- WSDL 关联 [37](#)
- WSDL 视图 [37](#)
- WSDL URL [16](#)
- 系统定义的故障 [55](#)
- 预定义故障 [56](#)
- 置换组 [75](#)
- 安全 [146](#)

Web 服务 (续)

- 操作属性 [145](#)
- 概览 [13](#)
- 监视 [150](#)
- 进程 [14](#)
- 配置结果集缓存 [145](#)
- 权限 [147](#)
- 权限类型 [147](#)
- 手动创建 [39](#)
- 属性 [143](#)
- 属性配置 [142](#)
- 在没有 WSDL 的情况下创建 [39](#)
- 资源属性 [145](#)
- 组件 [15](#)

Web 服务安全

- HTTP 客户端筛选器 [146](#)
- HTTPS [146](#)
- isAuthenticationRequired [146](#)
- 传递安全 [146](#)
- 传输层安全 [146](#)
- 权限 [146](#)
- 身份验证 [146](#)
- 授权 [146](#)
- 消息层安全 [146](#)

Web 服务使用者

- 进程 [14](#), [98](#)

未规范化的输入

- Web 服务端口 [73](#)

未规范化输出

- SOAP 消息解析 [62](#)

位置列

- Web 服务转换 [68](#)

WS-Security 用户名

- 动态端口 [82](#)

WSDL

- 从 WSDL 创建 Web 服务 [37](#)
- 说明 [16](#)
- 与 Web 服务的关联 [37](#)

WSDL 数据对象

- 创建 [20](#)
- 导入 [19](#)
- 概览视图 [20](#)
- 高级视图 [20](#)
- 架构视图 [19](#)
- 同步 [21](#)

WSDL URL

- 说明 [16](#)

WSDL 文件

- 绑定元素 [79](#)
- 操作元素 [79](#)
- 端口元素 [79](#)
- 服务元素 [79](#)

X

详细信息元素

- Web 服务故障处理 [54](#)

响应代码

- REST Web 服务使用者转换 [136](#)

系统定义的故障

- Web 服务故障处理 [55](#)

选项元素

- 解析 SOAP 消息 [65](#)
- 说明 [76](#)
- 在 REST Web 服务使用者转换中查看 [139](#)
- 在 Web 服务使用者转换中查看 [83](#), [86](#)

序列组

- 在 REST Web 服务使用者转换中查看 [139](#)

响应消息

- REST Web 服务 [106](#)

Y

以 XML 形式映射到输出

- 故障转换 [51](#)

- 输出转换 [48](#)

映射层次结构第一级

- 输入转换 [47](#)

映射第一级层次结构

- 故障转换 [51](#)

- 输出转换 [49](#)

- 说明 [103](#)

已转换的输出

- SOAP 消息解析 [62](#)

已转换的数据

- SOAP 消息 [72](#)

用户名标志

- 密码类型 [148](#)

- 密码摘要 [149](#)

由属性继承

- 架构对象 [27](#)

元素

- 创建 [42](#)

- 联合 [77](#)

原因

- Web 服务故障处理 [54](#)

预定义故障

- 创建 [43](#)

- Web 服务故障处理 [56](#)

Z

早期选择优化

- Web 服务使用者转换 [91](#)

折叠空格属性

- 架构对象 [26](#)

证书

- 非信任证书 [22](#), [32](#)

- 管理证书 [22](#), [32](#)

- 添加非信任证书 [22](#), [33](#)

- 证书属性 [22](#), [33](#)

置换组

- 架构对象 [26](#)

- 解析 SOAP 消息 [65](#)

- Web 服务 [75](#)

- Web 服务使用者转换 [82](#), [85](#)

种类属性

- 架构对象 [26](#)

自定义视图选项

- 说明 [57](#)

最大长度

- 架构对象 [25](#)

最多出现次数

- 架构对象 [25](#)

最少出现次数

- 架构对象 [25](#)

最小长度

- 架构对象 [25](#)

作为 XML 映射到输入

- 输入转换 [46](#)

资源

- REST Web 服务 [97](#), [98](#)

资源标识符
 在数据预览中筛选 [107](#)
资源键
 REST Web 服务 [101](#)
 搜索 [106](#)
 在请求 URI 中引用 [106](#)
资源映射
 “高级” 选项卡 [106](#)

资源映射 (续)
 REST Web 服务 [101](#)
 默认值 [101](#)
 说明 [97](#)
 自定义 [102](#)
自定义资源映射
 说明 [102](#)