



Informatica® PowerCenter  
10.5.6

# 웹 서비스 공급자 가이드

이 소프트웨어와 설명서는 사용 및 공개에 대한 제한 사항이 포함되어 있는 별도의 사용권 계약에 따라서는 제공됩니다. 본 문서의 어떤 부분도 Informatica LLC의 사전 통지 없이 어떠한 형태나 수단(전자적, 사진 복사, 녹음 등)으로 복제되거나 전송될 수 없습니다.

Informatica, Informatica 로고 및 PowerCenter는 미국과 전 세계 여러 관할 국가에서 Informatica LLC의 상표 또는 등록 상표입니다. Informatica 상표의 현재 목록은 <https://www.informatica.com/trademarks.html>에서 확인할 수 있습니다. 다른 회사 및 제품명은 해당 소유자의 상표 또는 등록 상표일 수 있습니다.

이 소프트웨어 및/또는 설명서 중 일부는 타사 저작권의 적용을 받으며, 이에 국한되지 않습니다. 저작권 DataDirect Technologies. 모든 권리 보유. 저작권 (c) Sun Microsystems. 모든 권리 보유. 저작권 (c) RSA Security Inc. 모든 권리 보유. 저작권 (c) Ordinal Technology Corp. 모든 권리 보유. 저작권 (c) Aandacht c.v. 모든 권리 보유. 저작권 Genivia, Inc. 모든 권리 보유. 저작권 Isomorphic Software. 모든 권리 보유. 저작권 (c) Meta Integration Technology, Inc. 모든 권리 보유. 저작권 (c) Intalio. 모든 권리 보유. 저작권 (c) Oracle. 모든 권리 보유. 저작권 (c) Adobe Systems Incorporated. 모든 권리 보유. 저작권 (c) DataArt, Inc. 모든 권리 보유. 저작권 (c) ComponentSource. 모든 권리 보유. 저작권 (c) Microsoft Corporation. 모든 권리 보유. 저작권 (c) Rogue Wave Software, Inc. 모든 권리 보유. 저작권 (c) Teradata Corporation. 모든 권리 보유. 저작권 (c) Yahoo! Inc. 모든 권리 보유. 저작권 (c) Glyph & Cog, LLC. 모든 권리 보유. 저작권 (c) Thinkmap, Inc. 모든 권리 보유. 저작권 (c) Clearpace Software Limited. 모든 권리 보유. 저작권 (c) Information Builders, Inc. 모든 권리 보유. 저작권 (c) OSS Nokalva, Inc. 모든 권리 보유. 저작권 Edifecs, Inc. 모든 권리 보유. 저작권 Cleo Communications, Inc. 모든 권리 보유. 저작권 (c) International Organization for Standardization 1986. 모든 권리 보유. 저작권 (c) ej-technologies GmbH. 모든 권리 보유. 저작권 (c) Jaspersoft Corporation. 모든 권리 보유. 저작권 (c) International Business Machines Corporation. 모든 권리 보유. 저작권 (c) yWorks GmbH. 모든 권리 보유. 저작권 (c) Lucent Technologies. 모든 권리 보유. 저작권 (c) University of Toronto. 모든 권리 보유. 저작권 (c) Daniel Veillard. 모든 권리 보유. 저작권 (c) Unicode, Inc. 저작권 IBM Corp. 모든 권리 보유. 저작권 (c) MicroQuill Software Publishing, Inc. 모든 권리 보유. 저작권 (c) PassMark Software Pty Ltd. 모든 권리 보유. 저작권 (c) LogiXML, Inc. 모든 권리 보유. 저작권 (c) 2003-2010 Lorenzi Davide, 모든 권리 보유. 저작권 (c) Red Hat, Inc. 모든 권리 보유. 저작권 (c) The Board of Trustees of the Leland Stanford Junior University. 모든 권리 보유. 저작권 (c) EMC Corporation. 모든 권리 보유. 저작권 (c) Flexera Software. 모든 권리 보유. 저작권 (c) Jinfonet Software. 모든 권리 보유. 저작권 (c) Apple Inc. 모든 권리 보유. 저작권 (c) Telerik Inc. 모든 권리 보유. 저작권 (c) BEA Systems. 모든 권리 보유. 저작권 (c) PDFlib GmbH. 모든 권리 보유. 저작권 (c) Orientation in Objects GmbH. 모든 권리 보유. 저작권 (c) Tanuki Software, Ltd. 모든 권리 보유. 저작권 (c) Ricebridge. 모든 권리 보유. 저작권 (c) Sencha, Inc. 모든 권리 보유. 저작권 (c) Scalable Systems, Inc. 모든 권리 보유. 저작권 (c) jQWidgets. 모든 권리 보유. 저작권 (c) Tableau Software, Inc. 모든 권리 보유. 저작권 (c) MaxMind, Inc. 모든 권리 보유. 저작권 (c) TMate Software s.r.o. 모든 권리 보유. 저작권 (c) MapR Technologies Inc. 모든 권리 보유. 저작권 (c) Amazon Corporate LLC. 모든 권리 보유. 저작권 (c) Highsoft. 모든 권리 보유. 저작권 (c) Python Software Foundation. 모든 권리 보유. 저작권 (c) BeOpen.com. 모든 권리 보유. 저작권 (c) CNRI. 모든 권리 보유.

이 제품에는 Apache Software Foundation(<http://www.apache.org/>)에서 개발한 소프트웨어 및/또는 Apache License의 다양한 버전("라이선스")에 따라 사용이 허가된 기타 소프트웨어가 포함되어 있습니다. <http://www.apache.org/licenses/>에서 이러한 라이선스의 복사본을 얻을 수 있습니다. 관련 법규 또는 서면 동의에 명시되어 있지 않은 경우, 이러한 라이선스에 따라 배포되는 소프트웨어는 어떠한 종류의 명시적이거나 묵시적인 보증 또는 조건 없이 "있는 그대로" 배포됩니다. 사용 권한에 대한 특정 언어별 라이선스 및 해당 라이선스에 따른 제한 사항을 참조하십시오.

이 제품에는 Mozilla(<http://www.mozilla.org/>)에서 개발한 소프트웨어, JBoss Group, LLC(저작권 JBoss Group, LLC, 모든 권리 보유.)가 저작권을 소유한 소프트웨어, Bruno Lowagie and Paulo Soares(저작권 (c) 1999-2006 by Bruno Lowagie and Paulo Soares)가 저작권을 소유한 소프트웨어 및 GNU Lesser General Public License Agreement(<http://www.gnu.org/licenses/lgpl.html>)의 다양한 버전에 따라 라이선스가 부여된 기타 소프트웨어가 포함되어 있습니다. 해당 정보는 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 이에 국한되지 않는 어떠한 종류의 명시적이거나 묵시적인 보증 없이 "있는 그대로" 제공되며, Informatica는 어떠한 책임도 지지 않습니다.

이 제품에는 Douglas C. Schmidt와 Washington University, University of California, Irvine, Vanderbilt University의 연구팀(저작권 ((c)) 1993-2006, 모든 권리 보유.)이 저작권을 소유한 ACE(TM) 및 TAO(TM) 소프트웨어가 포함되어 있습니다.

이 제품에는 OpenSSL Toolkit(저작권 The OpenSSL Project. 모든 권리 보유)에서 사용할 수 있도록 OpenSSL Project에서 개발한 소프트웨어가 포함되어 있으며 이 소프트웨어의 재배포는 <http://www.openssl.org> 및 <http://www.openssl.org/source/license.html>의 조항에 따라 변경될 수 있습니다.

이 제품에는 Curl 소프트웨어(저작권 1996-2013, Daniel Stenberg, <[daniel@haxx.se](mailto:daniel@haxx.se)>. 모든 권리 보유.)가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한 사항은 <http://curl.haxx.se/docs/copyright.html>에 명시된 조항에 따라 변경될 수 있습니다. 위와 같은 저작권 고지 및 이러한 허가 고지가 모든 제품에 표시되어 있는 경우 목적 및 사용료 유무에 관계없이 이 소프트웨어를 사용, 복사, 수정 및 배포할 수 있는 사용 권한이 부여됩니다.

이 제품에는 MetaStuff, Ltd(저작권 2001-2005 ((C)) MetaStuff, Ltd. 모든 권리 보유.)가 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 <http://www.dom4j.org/license.html>의 조항에 따라 변경될 수 있습니다.

이 제품에는 Per Bothner(저작권 (c) 1996-2006 Per Bothner. 모든 권리 보유.)가 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이러한 정보를 사용할 수 있는 권리는 <http://www.gnu.org/software/kawa/Software-License.html>의 라이선스에 설명되어 있습니다.

이 제품에는 OSSP UUID 소프트웨어(저작권 (c) 2002 Ralf S. Engelschall, 저작권 (c) 2002 The OSSP Project 저작권 (c) 2002 Cable & Wireless Deutschland)가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 <http://www.opensource.org/licenses/mit-license.php>의 조항에 따라 변경될 수 있습니다.

이 제품에는 Boost(<http://www.boost.org/>)에서 개발하거나 Boost 소프트웨어 라이선스에 따라 개발된 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 [http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt)의 조항에 따라 변경될 수 있습니다.

이 제품에는 University of Cambridge(저작권 (c) 1997-2007 University of Cambridge)가 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 <http://www.pcre.org/license.txt>의 조항에 따라 변경될 수 있습니다.

이 제품에는 Eclipse Foundation(저작권 (c) 2007 The Eclipse Foundation. 모든 권리 보유.)이 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 <http://www.eclipse.org/org/documents/epl-v10.php> 및 <http://www.eclipse.org/org/documents/edl-v10.php>의 조항에 따라 변경될 수 있습니다.

이 제품에는 <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/hsqLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.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/license.html>; <http://www.jgraph.com/jgraphdownload.html>; <http://www.jcraft.com/jsch/LICENSE.txt>; [http://jotm.objectweb.org/bsd\\_license.html](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/license.html>; <http://forge.ow2.org/projects/javaservice/>, <http://www.postgresql.org/about/license.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/IODBC/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://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/>, <https://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> 및 <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>.

이 제품에는 Academic Free License(<http://www.opensource.org/licenses/afl-3.0.php>), Common Development and Distribution License(<http://www.opensource.org/licenses/cddl1.php>), Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), Sun Binary Code License Agreement Supplemental License Terms, BSD License(<http://www.opensource.org/licenses/bsd-license.php>), 새 BSD License(<http://opensource.org/licenses/BSD-3-Clause>), MIT License(<http://www.opensource.org/licenses/mit-license.php>), Artistic License(<http://www.opensource.org/licenses/artistic-license-1.0>) 및 Initial Developer's Public License 버전 1.0(<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>)에 따라 라이선스가 부여된 소프트웨어가 포함되어 있습니다.

이 제품에는 Joe Walnes와 XStream Committers(저작권 (c) 2003-2006 Joe Walnes, 2006-2007 XStream Committers. 모든 권리 보유.)가 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한은 <http://xstream.codehaus.org/license.html>의 조항에 따라 변경될 수 있습니다. 이 제품에는 Indiana University Extreme! Lab에서 개발한 소프트웨어가 포함되어 있습니다. 자세한 내용을 확인하려면 <http://www.extreme.indiana.edu/>를 방문하십시오.

이 제품에는 Frank Balluffi 및 Markus Moeller(저작권 (c) 2013 Frank Balluffi and Markus Moeller. 모든 권리 보유.)가 저작권을 소유한 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 사용 권한 및 제한 사항은 MIT license에 명시된 조항에 따라 변경될 수 있습니다.

<https://www.informatica.com/legal/patents.html>에서 특허를 참조하십시오.

고지 사항: Informatica LLC는 비침해, 상품성 또는 특정 목적에 따른 사용에 대한 묵시적 보증을 포함하여 이에 국한되지 않는 어떠한 종류의 명시적이거나 묵시적인 보증 없이 이 문서를 "있는 그대로" 제공합니다. Informatica LLC는 이 소프트웨어나 문서에 오류가 없음을 보장하지 않습니다. 이 소프트웨어나 설명서에 제공된 정보에는 기술적 오류나 인쇄 오류가 있을 수 있습니다. 이 소프트웨어 및 설명서의 정보는 언제든지 예고 없이 변경될 수 있습니다.

고지 사항

이 Informatica 제품(이하 "소프트웨어")에는 Progress Software Corporation(이하 "DataDirect")의 운영 회사인 DataDirect Technologies의 특정 드라이버(이하 "DataDirect Drivers")가 포함되어 있습니다. 이러한 드라이버에는 다음 조건이 적용됩니다.

1. DataDirect Drivers는 상품성, 특정 목적에의 적합성 및 비침해에 대한 묵시적 보증을 포함하여 이에 국한되지 않는 어떠한 종류의 명시적이거나 묵시적인 보증 없이 "있는 그대로" 제공됩니다.
2. DataDirect 또는 그 타사 공급자는 손해의 발생 가능성을 사전에 알고 있었는지 여부에 관계없이 ODBC 드라이버의 사용으로 발생하는 직접, 간접, 부수적, 특별, 결과적 또는 기타 손해에 대해 어떠한 경우에도 최종 사용자에게 책임을 지지 않습니다. 이러한 제한 사항은 계약 위반, 보증 불이행, 과실, 무과실 책임, 허위 진술 및 기타 불법 행위를 포함하여 이에 국한되지 않는 모든 소송 사유에 적용됩니다.

이 설명서의 정보는 예고 없이 변경될 수 있습니다. 이 문서에서 문제가 발견되는 경우 [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)으로 보고해 주십시오.

Informatica 제품은 제품이 제공될 당시의 계약 조건에 따라 보증됩니다. Informatica는 상품성과 특정 목적에의 적합성에 대한 보증 그리고 비침해에 대한 보증 또는 조건을 포함하여 어떠한 종류의 명시적이거나 묵시적인 보증 없이 이 문서의 정보를 "있는 그대로" 제공합니다.

발행 날짜: 2024-09-20

# 목차

<b>서문</b>	<b>10</b>
Informatica 리소스	10
Informatica 네트워크	10
Informatica 기술 자료	10
Informatica 설명서	10
Informatica Product Availability Matrix	11
Informatica Velocity	11
Informatica Marketplace	11
Informatica 글로벌 고객 지원 센터	11
<b>장 1: 웹 서비스 개념</b>	<b>12</b>
웹 서비스 개념 개요	12
SOAP(Simple Object Access Protocol)	13
WSDL(Web Services Description Language)	13
<b>장 2: 웹 서비스 공급자 이해</b>	<b>15</b>
웹 서비스 공급자 이해 개요	15
웹 서비스 협	15
일괄 웹 서비스	16
실시간 웹 서비스	16
웹 서비스 공급자 아키텍처	16
성능 및 확장성	17
여러 개의 리포지토리를 웹 서비스 협과 연결	18
리포지토리를 여러 개의 웹 서비스 협 서비스와 연결	18
웹 서비스 워크플로우의 다중 인스턴스 실행	18
그리드에서 웹 서비스 세션 또는 워크플로우 실행	18
웹 서비스 협 보안	19
웹 서비스 협 로그	19
로그 구성	19
로그 보기	19
SOAP 결함 처리	20
사용자 정의 결함	20
시스템 결함	20
시스템 결함 스키마	20
<b>장 3: 웹 서비스 협 콘솔 사용</b>	<b>22</b>
웹 서비스 협 콘솔 사용 개요	22
웹 서비스 협 콘솔에 연결	22
웹 서비스 협 콘솔 이해	23
탐색기	23

웹 서비스 및 작업 섹션. . . . .	23
속성 섹션. . . . .	25
웹 서비스 테스트. . . . .	26
입력 메시지. . . . .	26
공용 웹 서비스 또는 일괄 웹 서비스 작업 테스트. . . . .	27
보안 실시간 웹 서비스 테스트. . . . .	27

## 장 4: 일괄 웹 서비스 작업. . . . . 29

일괄 웹 서비스 작업 개요. . . . .	29
메타데이터 웹 서비스 작업. . . . .	29
getAllDIServers. . . . .	30
getAllFolders. . . . .	30
getAllRepositories. . . . .	30
getAllTaskInstances. . . . .	30
getAllWorkflows. . . . .	31
Login. . . . .	31
Logout. . . . .	32
데이터 통합 웹 서비스 작업. . . . .	32
deinitializeDIServerConnection. . . . .	33
getDIServerProperties. . . . .	33
getNextLogSegment. . . . .	33
getSessionLog. . . . .	34
getSessionPerformanceData. . . . .	35
getSessionStatistics. . . . .	35
getTaskDetails. . . . .	36
getTaskDetailsEx. . . . .	36
getWorkflowDetails. . . . .	37
getWorkflowDetailsEx. . . . .	38
getWorkflowLog. . . . .	39
initializeDIServerConnection. . . . .	39
Login. . . . .	40
Logout. . . . .	41
monitorDIServer. . . . .	41
pingDIServer. . . . .	41
recoverWorkflow. . . . .	42
resumeWorkflow. . . . .	43
scheduleWorkflow. . . . .	44
startSessionLogFetch. . . . .	45
startTask. . . . .	45
startWorkflow. . . . .	46
startWorkflowEx. . . . .	47
startWorkflowFromTask. . . . .	48

startWorkflowLogFetch. . . . .	49
stopTask. . . . .	50
stopWorkflow. . . . .	51
unscheduleWorkflow. . . . .	52
waitTillTaskComplete. . . . .	53
waitTillWorkflowComplete. . . . .	54
<b>장 5: 클라이언트 응용 프로그램 기록. . . . .</b>	<b>56</b>
클라이언트 응용 프로그램 기록 개요. . . . .	56
일괄 웹 서비스용 클라이언트 응용 프로그램. . . . .	57
클라이언트 프록시 클래스 생성. . . . .	57
초기화. . . . .	57
세션 유지 관리. . . . .	58
작업 호출. . . . .	58
리소스 정리. . . . .	58
오류 처리. . . . .	58
프록시 개체. . . . .	58
일괄 웹 서비스용 <b>Java</b> 클라이언트 응용 프로그램. . . . .	59
Axis에서 클라이언트 프록시 클래스 생성. . . . .	59
Axis에서 초기화. . . . .	59
Axis에서 세션 유지 관리. . . . .	60
Axis에서 작업 호출 생성. . . . .	61
Axis에서 정리. . . . .	61
Axis에서 오류 처리. . . . .	61
일괄 웹 서비스용 <b>C#</b> 클라이언트 응용 프로그램. . . . .	61
.NET에서 클라이언트 프록시 클래스 생성. . . . .	62
.NET에서 초기화. . . . .	62
.NET에서 세션 유지 관리. . . . .	63
.NET에서 작업 호출 생성. . . . .	63
.NET에서 오류 처리. . . . .	63
실시간 웹 서비스용 클라이언트 응용 프로그램. . . . .	64
웹 서비스 워크플로우. . . . .	64
클라이언트 프록시 클래스 생성. . . . .	64
초기화. . . . .	65
작업 호출. . . . .	65
오류 처리. . . . .	65
실시간 웹 서비스용 <b>Java</b> 클라이언트 응용 프로그램. . . . .	65
실시간 웹 서비스용 클라이언트 응용 프로그램 작성. . . . .	66
매개 변수 배열 사용. . . . .	67
매개 변수 배열 정의. . . . .	67
매개 변수 배열 사용에 대한 규칙 및 지침. . . . .	69
클라이언트 요청에 보안 추가. . . . .	69

SOAP 요청의 UsernameToken. ....	70
일반 텍스트 암호. ....	70
해시된 암호. ....	71
요약 암호. ....	72
<b>장 6: 웹 서비스 소스 및 대상 작업.....</b>	<b>74</b>
웹 서비스 소스 및 대상 개요. ....	74
웹 서비스 소스 및 대상 이해. ....	74
XML 보기 및 그룹. ....	75
소스 정의. ....	75
대상 정의. ....	76
웹 서비스 소스 및 대상을 가져오거나 작성하기 위한 규칙 및 지침. ....	76
웹 서비스 소스 또는 대상 정의 가져오기. ....	77
가져오기 모드. ....	77
메시지 ID. ....	78
고급 옵션. ....	78
XML 보기를 작성하지 않고 WSDL에서 가져오기. ....	79
WSDL에서 웹 서비스 소스 또는 대상 정의 가져오기. ....	79
소스 또는 대상 정의 작성. ....	80
다중 발생 요소. ....	81
메시지 포트. ....	81
관계형 또는 플랫폼 파일 소스나 대상에서 소스 또는 대상 작성. ....	81
<b>장 7: 웹 서비스 소스 및 대상 편집.....</b>	<b>83</b>
웹 서비스 소스 및 대상 편집 개요. ....	83
디자이너 작업 공간에서 정의 편집. ....	83
테이블 탭. ....	84
열 탭. ....	84
특성 탭. ....	84
메타데이터 확장 탭. ....	84
웹 서비스 정의 탭. ....	84
WSDL 작업 공간에서 정의 편집. ....	85
WSDL 작업 공간에 대한 규칙 및 지침. ....	85
<b>장 8: 웹 서비스 매핑 작업.....</b>	<b>87</b>
웹 서비스 매핑 작업 개요. ....	87
웹 서비스 매핑 유형. ....	88
요청-응답 매핑. ....	88
준비된 매핑. ....	89
WSDL에서 매핑 생성. ....	89
관계형 또는 플랫폼 파일 소스나 대상에서 매핑 생성. ....	90
변환 또는 맵렛에서 매핑 생성. ....	90

재사용 가능 변환에서 매핑 생성.....	91
맷에서 매핑 생성.....	91
재사용 가능 변환 또는 맷에서 매핑 생성.....	92
웹 서비스 매핑에서 대상 인스턴스 편집.....	92
로드 범위.....	92
부분 로드 복구.....	93
첨부 파일.....	93
플랫 파일 또는 XML 소스 및 대상 첨부 파일.....	93
WSDL 첨부 파일.....	93

## 장 9: 웹 서비스 워크플로우에 대한 작업..... 95

웹 서비스 워크플로우에 대한 작업 개요.....	95
웹 서비스 워크플로우 작성 및 구성.....	96
웹 서비스 워크플로우 작성.....	96
웹 서비스 워크플로우 구성.....	96
웹 서비스 워크플로우 동시 실행.....	98
웹 서비스 공급자 판독기 및 기록기 구성.....	99
웹 서비스 공급자 판독기 구성.....	99
웹 서비스 공급자 기록기 구성.....	100
XML 및 플랫 파일 세션을 위한 판독기 및 기록기 구성.....	101
웹 서비스 세션을 위한 파티션 구성.....	101
웹 서비스 워크플로우 문제 해결.....	101

## 부록 A: 웹 서비스 샘플 클라이언트 응용 프로그램..... 103

웹 서비스 샘플 클라이언트 응용 프로그램 개요.....	103
일괄 웹 서비스 샘플 프로그램 사용.....	103
일괄 웹 서비스 샘플 프로그램 컴파일.....	104
일괄 웹 서비스 샘플 프로그램 실행.....	105
일괄 웹 서비스 예제.....	105
찾아보기.....	105
데이터 통합.....	107
다중 통합 서비스.....	108
다중 스테딩.....	108
웹 서비스 헵 테스트.....	109
실시간 웹 서비스 샘플 프로그램 사용.....	110
1단계. 조회 테이블 작성.....	111
2단계. 매핑 및 워크플로우 가져오기.....	112
3단계. SQL 변환을 위해 데이터베이스 및 데이터 유형 수정.....	112
4단계. 데이터베이스 연결 설정 수정.....	112
5단계. 실시간 웹 서비스 샘플 프로그램 컴파일.....	113
6단계. 실시간 웹 서비스 샘플 프로그램 실행.....	113
실시간 웹 서비스 예제.....	113



다중 행 조회.....	113
단일 행 조회.....	114
<b>부록 B: 웹 브라우저 구성.....</b>	<b>115</b>
웹 브라우저 구성.....	115
<b>인덱스.....</b>	<b>116</b>

# 서문

*PowerCenter® Web Services Provider* 가이드는 Web Services Provider에 대한 정보와 웹 서비스 협에서 호스팅하는 PowerCenter 웹 서비스에 대한 정보를 제공합니다. 이 가이드에서는 PowerCenter 워크플로우를 웹 서비스로 전환하는 방법과 웹 서비스 협에서 제공되는 웹 서비스를 사용하는 클라이언트 응용 프로그램을 생성하는 예제를 확인할 수 있습니다.

## Informatica 리소스

Informatica는 Informatica Network 및 기타 온라인 포털을 통해 다양한 범위의 제품 리소스를 제공합니다. 리소스를 통해 Informatica 제품 및 솔루션을 최대한 활용하고 다른 Informatica 사용자 및 주제별 전문가로부터 배울 수 있습니다.

### Informatica 네트워크

Informatica Network는 Informatica 기술 자료, Informatica 글로벌 고객 지원 센터 등 여러 리소스로 연결되는 관문입니다. Informatica Network를 시작하려면 <https://network.informatica.com>을 방문하십시오.

Informatica Network 멤버인 경우 다음 옵션이 가능합니다.

- 기술 자료에서 제품 리소스를 검색할 수 있습니다.
- 제품 사용 가능 여부에 대한 정보를 봅니다.
- 지원 사례를 생성하고 검토할 수 있습니다.
- 거주 지역의 Informatica 사용자 그룹 네트워크를 검색하고 동료와 협업 관계 유지

### Informatica 기술 자료

Informatica 기술 자료를 사용하여 사용 방법 문서, 모범 사례, 비디오 자습서, 자주 묻는 질문에 대한 답변 등 제품 리소스를 확인할 수 있습니다.

기술 자료를 검색하려면 <https://search.informatica.com>을 방문하십시오. 기술 자료에 대한 질문, 의견 또는 아이디어가 있는 경우 [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com)을 통해 Informatica 기술 자료 팀에 문의해 주시기 바랍니다.

### Informatica 설명서

Informatica 설명서 포털에서 확장된 설명서 라이브러리를 탐색하여 현재 및 최근 제품 릴리스를 확인할 수 있습니다. 설명서 포털을 탐색하려면 <https://docs.informatica.com>을 방문하십시오.

제품 설명서에 대한 질문, 의견 또는 아이디어가 있는 경우 [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)에서 Informatica 설명서 팀에 문의해 주시기 바랍니다.

## Informatica Product Availability Matrix

PAM(Product Availability Matrix)은 제품 릴리스에서 지원하는 운영 체제 버전, 데이터베이스 및 데이터 소스 유형과 대상을 나타냅니다.

<https://network.informatica.com/community/informatica-network/product-availability-matrices>에서 Informatica PAM을 찾을 수 있습니다.

## Informatica Velocity

Informatica Velocity는 수백 가지 데이터 관리 프로젝트의 실제 경험을 토대로 Informatica 전문 서비스업에서 개발한 팁과 모범 사례 모음입니다. Informatica Velocity는 전 세계의 조직과 협력하여 성공적인 데이터 관리 솔루션을 계획, 개발, 배포 및 유지 관리하는 Informatica 컨설턴트의 포괄적인 지식을 보여줍니다.

Informatica Velocity 리소스는 <http://velocity.informatica.com>에서 확인할 수 있습니다. Informatica Velocity에 대한 질문, 주석 또는 아이디어가 있으시면 Informatica 전문 서비스업([ips@informatica.com](mailto:ips@informatica.com))에 문의하십시오.

## Informatica Marketplace

Informatica Marketplace는 Informatica 구현을 확대 및 개선하기 위한 솔루션을 찾을 수 있는 포럼입니다. Marketplace에서 Informatica 개발자와 파트너가 제공하는 수백 개의 솔루션을 활용하여 생산성을 향상시키고 프로젝트의 구현에 걸리는 시간을 줄일 수 있습니다. <https://marketplace.informatica.com>에서 Informatica Marketplace를 찾을 수 있습니다.

## Informatica 글로벌 고객 지원 센터

전화 또는 Informatica 네트워크를 통해 글로벌 지원 센터에 문의할 수 있습니다.

해당 지역의 Informatica 글로벌 고객 지원 전화 번호는 Informatica 웹 사이트 (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>)를 방문하여 찾을 수 있습니다.

Informatica 네트워크에 대한 온라인 지원 리소스를 찾으려면 <https://network.informatica.com>으로 이동하고 eSupport 옵션을 선택하십시오.

# 제 1 장

## 웹 서비스 개념

이 장에 포함된 항목:

- [웹 서비스 개념 개요, 12](#)
- [SOAP\(Simple Object Access Protocol\), 13](#)
- [WSDL\(Web Services Description Language\), 13](#)

## 웹 서비스 개념 개요

웹 서비스는 웹을 통해 운영되는 비즈니스 기능입니다. 표준화된 XML 메시징을 통해 네트워크에서 액세스할 수 있는 작업의 컬렉션을 나타냅니다. PowerCenter 웹 서비스 공급자를 사용하면 PowerCenter 메타데이터와 데이터 통합 기능을 통합하여 웹 서비스로 제공할 수 있습니다. 모든 언어 또는 플랫폼의 통합 서비스와 통신할 수 있는 응용 프로그램을 기록할 수 있습니다. 이러한 응용 프로그램을 기존 구성 요소와 제품에 쉽게 포함시킬 수 있습니다.

웹 서비스는 개방형 표준(예: XML, SOAP 및 WSDL)을 기반으로 하며 전통적인 독점적 응용 프로그램보다 더 탁월한 상호 운용성을 제공합니다.

웹 서비스의 예에는 주식, 비행기 시간표 및 신용 조사와 같은 비즈니스 서비스가 포함됩니다.

웹 서비스를 활성화하는 구성 요소에는 다음이 포함됩니다.

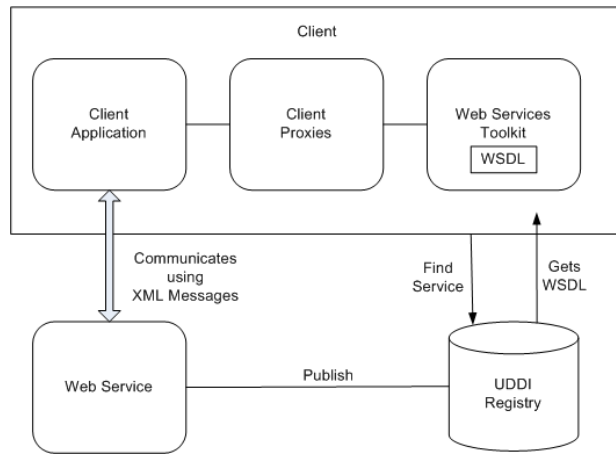
- **SOAP(Simple Object Access Protocol).** SOAP는 웹 서비스에 필요한 통신 프로토콜입니다. 웹 서비스 메시지의 XML 형식을 정의하는 사양입니다.
- **WSDL(Web Service Definition Language).** WSDL은 웹 서비스 작업을 설명하는 XML 문서입니다.
- **레지스트리.** 게시된 웹 서비스의 디렉터리입니다. 일부 웹 서비스 공급자는 UDDI(Universal Description, Discovery, and Integration)에 서비스를 게시합니다. UDDI에 웹 서비스를 등록하는 것은 선택 사항입니다.

**참고:** PowerCenter 웹 서비스 공급자는 UDDI 레지스트리를 사용하지 않습니다.

PowerCenter 웹 서비스 공급자에 대한 웹 서비스 클라이언트를 빌드하려면 상호 작용하려는 웹 서비스를 선택하고 선택한 웹 서비스에 대한 WSDL을 검색합니다. 웹 서비스 툴킷(예: Axis)을 사용하여 클라이언트 프록시를 생성합니다. 클라이언트 프록시에는 웹 서비스와 상호 작용하는 데 필요한 모든 함수 호출이 포함됩니다.

WSDL을 검토하여 웹 서비스에서 제공하는 함수, 웹 서비스에 필요한 데이터 및 서비스 위치를 결정할 수 있습니다. WSDL은 웹 서비스 인터페이스와 서비스에 사용할 수 있는 작업을 설명합니다. WSDL에 있는 정보를 통해 서비스를 사용할 클라이언트 응용 프로그램을 빌드합니다.

다음 그림은 웹 서비스의 구성 요소를 보여줍니다.



## SOAP(Simple Object Access Protocol)

SOAP는 웹 서비스의 통신 프로토콜입니다. 웹 서비스 메시지의 형식을 정의합니다. SOAP 인코딩은 데이터 구조(예: Java)에서 SOAP XML로 변환하는 방법을 SOAP 런타임 환경에 알려주는 데 사용됩니다. SOAP 및 WSDL은 웹 서비스와 해당 클라이언트 사이의 통신을 지시합니다.

SOAP 메시지에는 다음 섹션이 포함되어 있습니다.

- **SOAP 엔벨로프.** 엔벨로프는 메시지의 콘텐츠, 메시지를 처리해야 하는 사용자 또는 구성 요소 및 메시지가 필수 또는 선택 사항인지 여부를 비롯하여 메시지의 프레임워크를 정의합니다.
- **SOAP 헤더.** 헤더는 사용자가 분산형 방식으로 SOAP 메시지에 피처를 추가할 수 있게 해주는 SOAP 엔벨로프의 요소입니다.
- **SOAP 본문.** 본문은 대상 수신자와 정보를 교환하기 위한 메커니즘을 제공하는 필수 정보의 컨테이너입니다.

인증 및 트랜잭션 관리는 헤더 항목으로 구현할 수 있는 일반적인 확장 예제입니다. SOAP 헤더는 SOAP 메시지의 본문에 있는 데이터를 처리하는 데 도움을 줍니다. 인증 또는 트랜잭션과 관련된 정보는 SOAP 메시지 본문을 전송한 항목과 해당 정보를 처리할 컨텍스트를 식별하기 때문에 일반적으로 헤더에 포함됩니다.

SOAP 툴킷을 사용하여 SOAP 메시지를 생성하고 구문 분석합니다. SOAP 툴킷은 다른 언어의 함수 호출을 SOAP 메시지로 변환합니다. 예를 들어 Apache Axis 툴킷은 Java 함수 호출을 SOAP로 변환합니다.

SOAP를 사용하면 조직 내부와 외부에서 웹 서비스를 다양한 플랫폼에 구현할 수 있습니다. 각 SOAP 구현에서는 서로 다른 함수 호출 및 매개 변수를 지원합니다. 따라서 한 툴킷에서 작동하는 함수는 다른 툴킷에서 작동하지 않을 수 있습니다.

## WSDL(Web Services Description Language)

WSDL은 웹 서비스에서 사용하는 프로토콜 및 형식을 설명하는 XML 문서입니다.

서비스 요청을 보내는 사람과 받는 사람이 모두 교환되는 데이터를 이해할 수 있도록 웹 서비스에 전달되는 데이터에 대한 설명이 WSDL에 포함되어 있습니다. WSDL 요소에는 해당 데이터에서 수행되는 작업에 대한 설명(메

시지를 받는 사람에게 메시지 처리 방법을 알림) 및 바인딩 또는 전송에 대한 바인딩(메시지를 보내는 사람에게 메시지 전송 방법을 알림)이 포함되어 있습니다.

웹 서비스 협 콘솔에서 **PowerCenter** 웹 서비스 공급자가 호스팅하는 웹 서비스용 **WSDL** 파일을 보고 다운로드할 수 있습니다.

#### 관련 항목:

- [“웹 서비스 협 콘솔 사용” 페이지 22](#)

## 제 2 장

# 웹 서비스 공급자 이해

이 장에 포함된 항목:

- [웹 서비스 공급자 이해 개요, 15](#)
- [웹 서비스 공급자 아키텍처, 16](#)
- [성능 및 확장성, 17](#)
- [웹 서비스 협 보안, 19](#)
- [웹 서비스 협 로그, 19](#)
- [SOAP 결함 처리, 20](#)

## 웹 서비스 공급자 이해 개요

웹 서비스 공급자는 PowerCenter 웹 서비스 프레임워크의 공급자 항목이며 이 항목은 웹 서비스를 통해 외부 클라이언트가 PowerCenter 워크플로우 및 데이터 통합 기능에 액세스할 수 있게 만듭니다.

웹 서비스 공급자는 다음 구성 요소로 구성됩니다.

- **웹 서비스 협.** PowerCenter 도메인의 응용 프로그램 서비스이며 SOAP 표준을 사용하여 웹 서비스 클라이언트와 요청 및 응답을 받고 보냅니다. 웹 서비스 협은 통합 서비스 및 리포지토리 서비스와 상호 작용하여 요청을 처리하고 응답을 생성합니다.
- **일괄 웹 서비스.** 웹 서비스 공급자는 통합 서비스 프로세스와 리포지토리 메타데이터에 대한 액세스를 허용하는 웹 서비스 작업 집합을 제공합니다.
- **실시간 웹 서비스.** PowerCenter 워크플로우를 웹 서비스로 활성화하는 경우 실시간 웹 서비스를 작성합니다. PowerCenter 워크플로우를 웹 서비스로 변환하면 웹 서비스 클라이언트에서 워크플로우를 실행할 수 있습니다.

## 웹 서비스 협

웹 서비스 협은 PowerCenter 도메인의 웹 서비스 게이트웨이이며 이를 통해 클라이언트 응용 프로그램이 웹 서비스 표준 및 프로토콜을 사용하여 PowerCenter 기능에 액세스할 수 있습니다. 웹 서비스 협을 사용하면 PowerCenter 워크플로우를 웹 서비스로 활성화할 수 있습니다. PowerCenter 프로세스를 모니터링하고 리포지토리 정보를 가져올 수도 있습니다.

웹 서비스 협은 데이터 통합 프로세스가 PowerCenter 프레임워크 내에서 유지되도록 허용하지만 웹 서비스 기술을 사용하여 요청 및 응답을 처리합니다. 웹 서비스 협은 웹 서비스 클라이언트에서 SOAP 메시지 양식으로 요청을 받고 통합 서비스에 전달합니다. 통합 서비스는 리포지토리 서비스와 함께 작동하여 요청을 처리하고 결과를 웹 서비스 협에 보냅니다. 웹 서비스 협은 웹 서비스 클라이언트에 SOAP 메시지 양식으로 응답을 보냅니다.

웹 서비스 협은 사용자가 웹 서비스를 관리하고 웹 서비스용 WSDL을 보고 다운로드할 수 있는 웹 서비스 협 콘솔을 제공합니다. WSDL 파일을 사용하여 웹 서비스에 액세스할 클라이언트 응용 프로그램을 작성할 수 있습니다.

PowerCenter 설치에는 웹 서비스 협이 포함되어 있습니다. PowerCenter를 설치한 후 도메인의 다른 응용 프로그램 서비스와 마찬가지로 Informatica Administrator를 사용하여 웹 서비스 협을 작성하고 활성화합니다.

## 일괄 웹 서비스

웹 서비스 공급자는 워크플로우를 실행 및 모니터링하고 메타데이터 정보에 액세스할 수 있는 웹 서비스 작업을 제공합니다. 웹 서비스 작업을 포괄적으로 일괄 웹 서비스라고 합니다. 일괄 웹 서비스 작업은 웹 서비스 협과 연결된 리포지토리의 개체에 대한 정보를 제공하거나, 통합 서비스에 연결하여 워크플로우 및 태스크 실행을 관리하거나, 워크플로우 및 세션에 대한 정보를 가져올 수 있습니다.

일괄 웹 서비스는 다음 범주로 나뉩니다.

- **데이터 통합 웹 서비스.** 데이터 통합 웹 서비스를 사용하여 통합 서비스에 연결하고 PowerCenter 워크플로우를 실행하거나 모니터링합니다. 데이터 통합 웹 서비스는 통합 서비스에 대한 세부 정보를 가져오고, 워크플로우를 예약 및 실행하고, 워크플로우의 태스크를 시작 및 중지하고, 세션의 통계를 모니터링하거나 가져올 수 있는 작업을 제공합니다.
- **메타데이터 웹 서비스.** 메타데이터 웹 서비스는 PowerCenter 리포지토리에서 메타데이터를 검색하는 작업을 제공합니다. 메타데이터 웹 서비스를 사용하여 리포지토리 개체(예: 폴더, 워크플로우 및 워크플로우 태스크)에 대한 정보를 가져오면 리포지토리에서 워크플로우를 실행하고 모니터링하는 데 도움이 됩니다.

## 실시간 웹 서비스

설치 후에 웹 서비스 협을 처음으로 시작하는 경우 실시간 웹 서비스를 사용할 수 없습니다. 실시간 웹 서비스는 PowerCenter 워크플로우를 웹 서비스로 노출하는 경우에 작성합니다. 클라이언트를 작성하여 웹 서비스 워크플로우를 실행하고 워크플로우 프로세스의 결과를 가져올 수 있습니다. 웹 서비스는 SOAP 메시지 요청을 수신하고 SOAP 메시지 응답을 생성합니다.

서비스 매핑을 작성하여 웹 서비스 클라이언트에서 메시지를 수신하고, 메시지를 변환하고, PowerCenter에서 지원하는 모든 대상에 메시지를 기록할 수 있습니다. 웹 서비스 소스 및 대상 정의가 둘 다 포함된 웹 서비스 매핑을 작성하여 웹 서비스 클라이언트에서 메시지 요청을 수신하고, 데이터를 변환하고, 웹 서비스 클라이언트에 응답을 보낼 수도 있습니다. 소스 및 대상 정의는 서비스 작업을 나타냅니다. 소스는 사용자 요청을 정의하고 대상은 응답을 정의합니다.

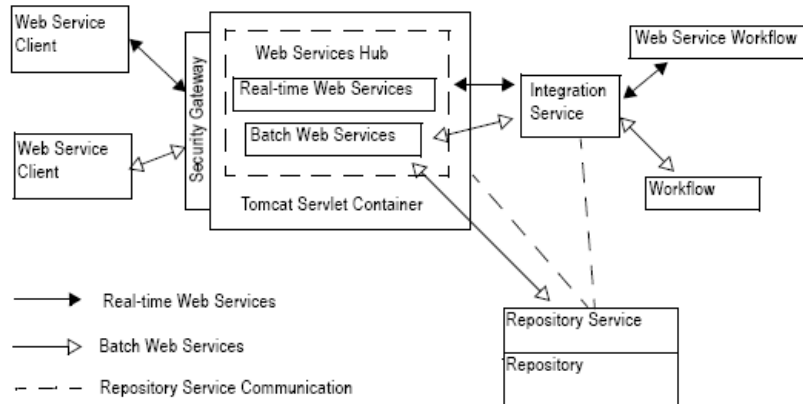
매핑을 작성한 후 웹 서비스 워크플로우를 작성하여 웹 서비스 매핑에 정의된 프로세스를 실행할 수 있습니다. 웹 서비스 워크플로우는 웹 서비스로 활성화된 워크플로우입니다. 웹 서비스 워크플로우를 구성하고 워크플로우에 세션을 추가합니다. 워크플로우를 저장하면 웹 서비스 협이 웹 서비스를 웹 서비스 협 콘솔에 게시합니다. 통합 서비스는 요청-응답 및 단방향 서비스를 둘 다 병렬로 처리할 수 있습니다.

## 웹 서비스 공급자 아키텍처

웹 서비스 공급자는 웹 서비스 협과 웹 서비스 협에서 호스팅하는 일괄 및 실시간 웹 서비스로 구성됩니다. 웹 서비스 협은 통합 서비스 및 리포지토리 서비스와 함께 작동하여 웹 서비스 요청을 처리합니다.



다음 그림은 웹 서비스 공급자 아키텍처를 보여줍니다.



웹 서비스 협은 실시간 웹 서비스와 일괄 웹 서비스에 대한 요청을 유사한 방식으로 처리합니다.

다음 프로세스는 웹 서비스 협이 웹 서비스 요청을 처리하는 방식을 설명합니다.

1. 웹 서비스 클라이언트가 **SOAP** 메시지를 웹 서비스 협에 보내서 웹 서비스를 실행합니다.
2. 일괄 웹 서비스의 경우 웹 서비스 협이 로그인 중에 생성된 세션 ID를 기반으로 웹 서비스 클라이언트를 인증합니다.  
보안 실시간 웹 서비스의 경우 웹 서비스 협이 사용자 이름 토큰을 기반으로 웹 서비스 클라이언트를 인증합니다.
3. 웹 서비스 협이 요청에 대한 메시지 ID를 생성합니다.  
요청이 실시간 웹 서비스에 대한 요청인 경우 웹 서비스 협이 통합 서비스에 메시지를 보냅니다.  
요청이 일괄 웹 서비스 작업에 대한 요청인 경우 웹 서비스 협이 요청 유형에 따라 통합 서비스 또는 리포지토리 서비스에 메시지를 보냅니다. 예를 들어 요청이 워크플로우를 실행하거나 통합 서비스를 시작 또는 중지하는 요청인 경우 웹 서비스 협이 통합 서비스에 요청을 처리하도록 메시지를 보냅니다. 요청이 리포지토리의 워크플로우 목록을 가져오는 요청인 경우 웹 서비스 협이 리포지토리 서비스에 요청을 처리하도록 메시지를 보냅니다.
4. 통합 서비스 또는 리포지토리 서비스가 요청을 처리합니다.  
요청이 실시간 웹 서비스에 대한 요청인 경우 통합 서비스가 메시지 ID를 사용하여 요청을 응답과 연관시키는 웹 서비스 협에 처리된 데이터를 보냅니다.
5. 웹 서비스 협이 웹 서비스 클라이언트에 **SOAP** 응답을 보냅니다.

통합 서비스와 웹 서비스 협은 프로세스 전체에서 리포지토리 서비스와 통신합니다.

## 성능 및 확장성

단일 노드에서 2개 이상의 웹 서비스 협을 실행할 수 있습니다. 하나의 노드에서 여러 개의 웹 서비스 협을 실행하면 노드에서 실행할 수 있는 웹 서비스 수가 증가되며 리소스 사용률이 최대화됩니다.

도메인에서 웹 서비스를 실행하도록 웹 서비스 협을 구성하는 경우 다음 옵션을 사용하여 성능을 향상시키고 유연성 및 확장성을 제공할 수 있습니다.

- 여러 개의 리포지토리를 웹 서비스 협과 연결합니다.
- 리포지토리를 여러 개의 웹 서비스 협과 연결합니다.

- 웹 서비스 워크플로우의 다중 인스턴스를 실행합니다.
- 그리드에서 웹 서비스 세션 또는 워크플로우를 실행합니다.

## 여러 개의 리포지토리를 웹 서비스 협과 연결

둘 이상의 리포지토리를 웹 서비스 협과 연결할 수 있습니다. 여러 개의 리포지토리를 웹 서비스 협과 연결하면 웹 서비스 협은 연결된 모든 리포지토리에 있는 웹 서비스 워크플로우를 실행할 수 있습니다. 이렇게 하면 하나의 웹 서비스 협을 사용하여 서로 다른 시간에 여러 사용자가 액세스할 수 있는 웹 서비스를 실행할 수 있으며 웹 서비스 협 사용률이 최대화됩니다.

## 리포지토리를 여러 개의 웹 서비스 협 서비스와 연결

리포지토리를 2개 이상의 웹 서비스 협과 연결할 수 있습니다. 하나의 리포지토리를 여러 개의 웹 서비스 협 서비스와 연결하면 여러 개의 웹 서비스 협 서비스가 동일한 웹 서비스를 실행할 수 있습니다.

웹 서비스 협 서비스 전체에서 서비스 요청 로드 균형형을 맞출 수 있도록 타사 로드 균형 조정기를 사용하여 웹 서비스 협 서비스에 대한 요청을 관리하고 배포합니다. 프로덕션 환경에서 사용되는 하드웨어 로드 균형 조정기는 웹 서비스의 성능을 최적화할 수 있습니다. **Administrator** 도구에서 웹 서비스 협 서비스를 작성할 때 로드 균형 조정기에 대한 URL을 설정합니다.

일괄 웹 서비스를 사용하는 경우 로드 균형 조정기에서 여러 웹 서비스 협 서비스에 요청을 분산할 수 없습니다. 웹 서비스 협이 웹 서비스 클라이언트를 인증할 때 사용하는 세션 ID는 로그인 작업을 호출하는 웹 서비스 협에 대해서만 유효합니다. 동일한 노드 또는 다른 노드의 여러 웹 서비스 협 서비스에서 동일한 세션 ID를 일괄 웹 서비스 작업에 사용할 수 없습니다.

## 웹 서비스 워크플로우의 다중 인스턴스 실행

2개 이상의 인스턴스에서 실행되도록 워크플로우를 구성하는 경우 웹 서비스 협은 웹 서비스의 새 인스턴스를 동적으로 시작하여 가능한 많은 웹 서비스 요청을 처리할 수 있습니다. 웹 서비스 협은 웹 서비스 사용량을 모니터링하여 리소스 사용량과 웹 서비스 처리 시간을 결정합니다. 웹 서비스 협이 웹 서비스에 대한 요청을 처리하는 데 사용할 수 있는 최대 시간에 대한 임계값을 설정할 수 있습니다. 처리 시간이 임계값을 초과하면 웹 서비스 협이 웹 서비스 워크플로우의 다른 인스턴스를 시작하여 새 요청을 처리합니다.

서비스 요청 수가 감소하면 웹 서비스 협이 웹 서비스 인스턴스를 동적으로 종료하여 리소스 사용량을 줄입니다.

### 관련 항목:

- [“웹 서비스 워크플로우 동시 실행” 페이지 98](#)

## 그리드에서 웹 서비스 세션 또는 워크플로우 실행

**PowerCenter** 도메인에 그리드가 포함되어 있는 경우 그리드에서 웹 서비스 워크플로우를 실행할 수 있습니다. **Administrator** 도구에서 그리드를 작성하고 통합 서비스를 그리드와 연결합니다. 그런 다음 웹 서비스 워크플로우를 실행할 통합 서비스를 할당합니다.

클라이언트 응용 프로그램의 그리드에서 웹 서비스 워크플로우를 실행하려면 그리드와 연결된 통합 서비스에서 웹 서비스 워크플로우를 실행합니다.

그리드에서 실행할 세션을 활성화할 수도 있습니다. 세션이 그리드에서 실행되는 경우 통합 서비스는 그리드의 노드 전체에 세션 스레드를 배포합니다. 그리드에서 세션을 실행하려면 웹 서비스 소스 및 대상 정의에 메시지 ID를 추가합니다. 통합 서비스는 이 메시지 ID를 사용하여 노드 전체에서 웹 서비스 입력 및 출력 메시지를 연결합니다.

# 웹 서비스 협 보안

웹 서비스 협에는 다음과 같은 보안 수준이 있습니다.

- **암호화.** 웹 서비스 협은 리포지토리에 연결하는 데 사용되는 구성 파일에 리포지토리 로그인 정보를 암호화합니다. 보안 모드로 웹 서비스 협을 실행하고 웹 서비스 클라이언트 요청의 암호화를 위해 SSL 프로토콜을 사용할 수도 있습니다.
- **인증.** 일괄 웹 서비스의 경우 웹 서비스 클라이언트 응용 프로그램은 다른 작업을 호출하기 전에 **Login** 작업을 호출해야 합니다. 웹 서비스 협은 세션 ID를 기반으로 요청을 인증합니다.  
보안 실시간 웹 서비스의 경우 웹 서비스 협이 사용자 이름 토큰을 기반으로 웹 서비스 클라이언트를 인증합니다. 웹 서비스 클라이언트는 웹 서비스 협에 보내는 모든 **SOAP** 요청에 사용자 이름 토큰을 포함시켜야 합니다. 사용자 이름 토큰에는 일반 텍스트 암호, 해시된 암호 또는 요약 암호가 포함될 수 있습니다.  
웹 서비스 협은 공용 실시간 웹 서비스에 대한 웹 서비스 요청을 인증하지 않습니다.
- **권한 부여.** 리포지토리 액세스 권한이 있는 웹 서비스 클라이언트는 서비스를 실행할 폴더에 대한 권한을 가지고 있어야 합니다. 보안 실시간 웹 서비스의 경우 폴더에 대한 적절한 사용 권한이 있는 웹 서비스 클라이언트가 서비스 구성에 따라 해당 폴더에서 서비스를 실행할 수 있습니다. 예를 들어 서비스가 실행 가능하지 않은 경우 웹 서비스 클라이언트는 해당 서비스를 시작할 수 없지만 웹 서비스 워크플로우가 실행 중이면 서비스를 해당 호출할 수 있습니다.

관련 항목:

- [“클라이언트 요청에 보안 추가” 페이지 69](#)

# 웹 서비스 협 로그

웹 서비스 협은 서비스 초기화, 태스크 실행 및 연결 상태와 같은 태스크 관련 상태 및 오류 메시지에 대한 로그를 작성합니다. 로그에는 클라이언트의 IP 주소, 클라이언트가 호출하는 서비스 및 연결된 워크플로우가 포함됩니다. 이 로그의 오류 메시지를 검토하여 문제를 해결할 수 있습니다.

Administrator 도구에서 웹 서비스 협에 대한 로그를 보고 구성할 수 있습니다.

**참고:** 또한 웹 서비스 협은 요청을 처리할 수 없는 경우 **SOAP** 응답의 **fault** 요소에 메시지를 기록합니다.

관련 항목:

- [“SOAP 결함 처리” 페이지 20](#)

## 로그 구성

PowerCenter 도메인의 로그 관리자는 웹 서비스 협을 포함하여 도메인의 전체 서비스에 대한 모든 로깅 기능을 처리합니다.

Administrator 도구에서 웹 서비스 협 로그의 크기 및 위치와 로그에 포함될 오류 수준을 구성할 수 있습니다.

## 로그 보기

Administration Console 로그 뷰어에서 웹 서비스 협 로그 이벤트를 볼 수 있습니다. 로그 이벤트를 필터링하여 웹 서비스 협에 대한 로그 이벤트 목록만 가져올 수 있습니다. 로그 뷰어에서 로그 이벤트를 보는 경우 로그 관리자는 도메인 관리자가 설정한 로그 디렉터리의 생성된 파일에서 로그 이벤트를 표시합니다.

# SOAP 결함 처리

웹 서비스 협은 SOAP 결함 메시지로 오류 응답을 보냅니다. 웹 서비스 협은 다음 유형의 결함 응답을 생성할 수 있습니다.

- 사용자 정의 결함
- 시스템 결함

## 사용자 정의 결함

오류 데이터를 대상에 보내기 위해 대상 정의에서 결함 보기를 정의할 수 있습니다. 웹 서비스 매핑의 변환 논리에 따라 오류 데이터를 대상에 보내는 경우 통합 서비스는 메시지를 결함 대상에 기록합니다. 특정 오류를 포착하고 해결하려는 경우 오류 데이터를 대상에 보냅니다. 예를 들어 응답의 데이터 유형을 문자열로 예상합니다. 웹 서비스 워크플로우에서 숫자 응답을 보내는 경우 해당 응답을 결함 대상에 보낼 수 있습니다. 그런 다음 응답을 평가하고 오류를 해결할 수 있습니다.

## 시스템 결함

웹 서비스 협이 시스템 오류를 발견하면 오류 유형에 따라 결함 메시지를 생성하고 응답을 웹 서비스 클라이언트에 보냅니다. 결함 메시지는 웹 서비스 협이 오류를 발견할 때 수행하는 태스크를 기반으로 합니다.

- 웹 서비스 협이 SOAP 요청 메시지의 **header** 요소를 처리할 수 없는 경우 SOAP 응답 **header** 요소의 하위 요소에 SOAP 요청의 헤더 항목과 관련된 오류 정보를 반환합니다.
- 웹 서비스 협이 SOAP 요청의 **header** 요소에서 오류를 발견하는 경우 **body** 요소를 처리하지 않습니다. 요청에 대한 SOAP 응답에는 SOAP 헤더의 **header fault** 요소와 **detail** 요소가 없는 SOAP fault 요소가 포함됩니다.
- 웹 서비스 협이 **body** 요소의 콘텐츠를 처리할 수 없는 경우 SOAP 응답 메시지의 SOAP fault 요소에는 오류 정보가 있는 **detail** 요소가 포함됩니다.
- 웹 서비스 협은 다음과 같은 시스템 오류를 발견하는 경우 **detail** 요소에 오류 정보가 있는 SOAP 결함 응답을 생성합니다.
  - 통합 서비스가 실행 중이지 않으며 웹 서비스 협이 입력 메시지를 처리할 수 없습니다.
  - 웹 서비스 협의 제한 시간이 초과되었습니다.
  - 보안 웹 서비스가 올바른 사용자 이름 토큰을 제공하지 않습니다.
- 웹 서비스 협은 다음 상황에서 웹 서비스 요청에 대한 응답을 반환하지 않습니다.
  - 서비스 요청의 콘텐츠 형식이 잘못되었거나 구문 분석 오류가 발생합니다.
  - 워크플로우에서 요청을 필터링하여 제외합니다.

## 시스템 결함 스키마

기본적으로 시스템 결함 메시지에는 접두사 및 코드 번호가 있는 메시지 코드와 메시지 텍스트가 포함됩니다. 예를 들어 메시지 코드 WSH\_95002는 빈 워크플로우 이름이 포함된 올바른지 않은 요청과 관련이 있습니다.

메시지 코드는 SOAP 결함의 **detail** 요소에 있는 **ErrorCode** 요소이며 메시지 텍스트는 SOAP 결함의 **faultstring** 요소입니다.

## SOAP 결함 헤더

웹 서비스 협은 SOAP 응답 헤더의 **header fault** 요소에 있는 헤더 관련 오류를 보고합니다.

이 요소의 스키마는 아래에 나열되어 있습니다.

```
<ns1:HeaderFault xmlns:ns1="http://www.informatica.com/wsh">
  <ErrorCode>
    error code
  </ErrorCode>
  <ErrorMessage>
    error message
  </ErrorMessage>
</ns1:HeaderFault>
```

## SOAP 결함 본문

SOAP 결함 본문에는 다음 하위 요소가 포함됩니다.

- **Faultcode.** faultcode는 오류의 원인이 웹 서비스 클라이언트 또는 통합 서비스인지를 결정합니다. 메시지가 잘못된 구조를 가지고 있는 경우 오류의 원인은 웹 서비스 클라이언트일 수 있습니다.
- **Faultstring.** faultstring은 오류에 대한 설명을 제공합니다. faultstring 값은 오류의 원인이 통합 서비스, 웹 서비스 헵 또는 리포지토리 서비스인지를 나타냅니다.
- **세부 정보.** detail 요소는 오류 코드가 포함된 오류 정보를 제공하며 extended details 요소는 faultstring이 웹 서비스 헵 또는 리포지토리 서비스 오류인 경우 자세한 오류 정보를 제공합니다.

웹 서비스 헵이 사용하는 SOAP 결함 스키마는 다음과 같습니다.

```
<SOAP-ENV: Fault>
  <faultcode> Client/Server </faultcode>
  <faultstring>Brief Description of Error</faultstring>
  <detail>
    <ns:WSHFaultDetails xmlns:ns="www.informatica.com/wsh">
      <ErrorCode>
        Error Code
      </ ErrorCode >
      <ExtendedDetails>
        Actual Error
      </ ExtendedDetails >
    </ns:WSHFaultDetails>
  </detail>
</SOAP-ENV: Fault>
```

## 제 3 장

# 웹 서비스 협 콘솔 사용

이 장에 포함된 항목:

- [웹 서비스 협 콘솔 사용 개요, 22](#)
- [웹 서비스 협 콘솔에 연결, 22](#)
- [웹 서비스 협 콘솔 이해, 23](#)
- [웹 서비스 테스트, 26](#)

## 웹 서비스 협 콘솔 사용 개요

웹 서비스 협 콘솔은 웹 서비스 협에서 사용 가능한 실시간 웹 서비스 및 일괄 웹 서비스 작업을 보고 테스트하는데 사용하는 PowerCenter 응용 프로그램입니다. 웹 서비스 협 콘솔을 사용하여 다음 태스크를 수행할 수 있습니다.

- **실시간 웹 서비스의 속성 보기.** 웹 서비스가 보호되는지 여부와 같은 속성과 웹 서비스에 대한 설명을 볼 수 있습니다. 웹 서비스가 포함된 폴더와 리포지토리도 볼 수 있습니다.
- **실시간 웹 서비스용 WSDL 보기.** WSDL을 다운로드하려면 하드 디스크의 파일에 WSDL을 저장합니다.
- **실시간 웹 서비스 테스트.** 웹 서비스 협 콘솔에서 시험관 클라이언트 응용 프로그램을 사용하여 올바른 웹 서비스를 실행하고 응답을 봅니다. 일괄 웹 서비스의 작업을 테스트할 수도 있습니다.
- **일괄 웹 서비스 작업에 대한 설명 보기.** 일괄 웹 서비스 작업에 대한 설명을 볼 수 있습니다. 시험관 응용 프로그램을 사용하여 작업에 대한 매개 변수를 볼 수 있습니다.
- **데이터 통합 및 메타데이터 웹 서비스용 WSDL 보기.** WSDL을 다운로드하려면 하드 디스크의 파일에 WSDL을 저장합니다.
- **일괄 웹 서비스 작업 테스트.** 웹 서비스 협 콘솔에서 시험관 응용 프로그램을 사용하여 일괄 웹 서비스 작업을 실행하고 응답을 봅니다.

**참고:** 웹 서비스 협 콘솔에는 인증이 필요하지 않습니다. 로그인하지 않아도 웹 서비스 협 콘솔에 액세스할 수 있습니다. 보안을 유지하려면 보안 네트워크 환경 내에서 웹 서비스 협을 실행해야 합니다.

## 웹 서비스 협 콘솔에 연결

모든 브라우저에서 웹 서비스 협 콘솔에 연결할 수 있습니다.

다음 URL 중 하나를 사용하여 웹 서비스 헵 콘솔에 연결합니다.

```
http://<WebServicesHubHostName:PortNumber>/wsh  
http://<WebServicesHubHostName:PortNumber>/PowerCenter
```

컨텍스트 이름 /wsh 및 /PowerCenter는 대/소문자를 구분합니다.

HTTP를 실행하는 웹 서비스 헵의 기본 포트는 7333입니다. HTTPS를 통한 보안 연결을 사용하도록 웹 서비스 헵을 구성할 수도 있습니다. HTTPS를 실행하는 웹 서비스 헵의 기본 포트는 7343입니다. Administrator 도구에 서 웹 서비스 헵을 작성할 때 포트 번호를 설정할 수 있습니다.

Administrator 도구에서 웹 서비스 헵 콘솔에 연결할 수도 있습니다. 웹 서비스 헵의 세부 정보를 확인하고 서비스 URL을 클릭합니다. 웹 서비스 헵 콘솔에 연결하려면 웹 서비스 헵을 활성화해야 합니다.

## 웹 서비스 헵 콘솔 이해

웹 서비스 헵 콘솔은 다음 섹션으로 구성됩니다.

- **Navigator.** 탐색기에는 웹 서비스 헵 콘솔에서 볼 수 있는 서비스의 유형이 표시됩니다.
- **웹 서비스 또는 작업.** 실시간 웹 서비스의 경우 웹 서비스 섹션에는 올바른 웹 서비스와 올바르지 않은 웹 서비스가 표시됩니다. 일괄 웹 서비스의 경우 작업 섹션에는 메타데이터 웹 서비스와 데이터 통합 웹 서비스에 사용할 수 있는 작업이 표시됩니다.  
웹 서비스 섹션에서 웹 서비스를 테스트하거나 웹 서비스용 WSDL을 볼 수 있습니다.  
작업 섹션에서는 일괄 웹 서비스 작업을 테스트하거나 일괄 웹 서비스 WSDL을 보고 저장할 수 있습니다.
- **설명.** 설명 섹션에는 탐색기에서 선택한 웹 서비스 유형에 대한 정보가 제공됩니다.
- **속성.** 속성 섹션에는 웹 서비스 또는 작업 섹션에서 선택한 웹 서비스 또는 웹 서비스 작업의 속성이 표시됩니다.

## 탐색기

탐색기에서 정보를 표시하려는 웹 서비스의 유형을 스크롤하여 선택할 수 있습니다. 콘솔의 다른 섹션에 표시되는 정보는 탐색기에서 선택하는 웹 서비스 유형에 따라 달라집니다.

## 웹 서비스 및 작업 섹션

웹 서비스 헵 콘솔은 탐색기에서 선택하는 웹 서비스 유형에 따라 웹 서비스 섹션 또는 작업 섹션을 표시합니다.

탐색기에서 올바른 웹 서비스 또는 올바르지 않은 웹 서비스를 선택하면 웹 서비스 헵에서 실행되는 실시간 웹 서비스에 대한 정보가 웹 서비스 섹션에 표시됩니다.

탐색기에서 메타데이터 웹 서비스 또는 데이터 통합 웹 서비스를 선택하면 웹 서비스 헵에서 사용 가능한 일괄 웹 서비스 작업이 작업 섹션에 표시됩니다.

웹 서비스 또는 작업 목록을 정렬할 수 있습니다. 웹 서비스 또는 작업 목록을 정렬하려면 정렬하려는 열의 레이블을 클릭합니다. 웹 서비스 헵 콘솔은 클릭하는 열을 기준으로 웹 서비스 또는 작업을 사전 순으로 나열합니다. 열 레이블 옆에 있는 화살표는 목록의 정렬 순서(오름차순 또는 내림차순)를 나타냅니다.

시험판 응용 프로그램을 사용하여 웹 서비스 및 작업 섹션에 나열된 웹 서비스 작업을 테스트할 수 있습니다. 웹 서비스 작업을 테스트하려면 웹 서비스 작업의 입력 메시지에 매개 변수 값을 입력하고 응답을 확인합니다.

웹 서비스 섹션에서 웹 서비스용 WSDL을 표시할 수 있습니다. 작업 섹션에서는 일괄 웹 서비스용 WSDL을 표시할 수 있습니다. WSDL은 작업이 아닌 메타데이터 웹 서비스 또는 데이터 통합 웹 서비스에 대해 게시됩니다. 작업을 선택하고 WSDL을 클릭하면 웹 서비스 헵이 메타데이터 웹 서비스 또는 데이터 통합 웹 서비스용 WSDL을

표시합니다. WSDL을 사용하여 실시간 웹 서비스 또는 일괄 웹 서비스 작업을 호출하는 클라이언트 응용 프로그램을 작성합니다.

## 웹 서비스 섹션

웹 서비스 힙 콘솔에서 실시간 웹 서비스를 보려면 웹 서비스 워크플로우를 작성해야 합니다. 웹 서비스 워크플로우를 표시하도록 구성하면 웹 서비스 힙이 웹 서비스 및 WSDL을 웹 서비스 힙 콘솔에 게시합니다.

탐색기에서 올바른 웹 서비스 또는 올바르지 않은 웹 서비스를 선택하면 웹 서비스 힙 콘솔에 표시하도록 구성된 실시간 웹 서비스 목록이 웹 서비스 섹션에 표시됩니다.

리포지토리의 개체를 관리할 권한이 있는 경우 리포지토리와 연결된 모든 웹 서비스를 볼 수 있습니다. 다른 사용자가 작성한 웹 서비스는 볼 수 있지만 실행할 수는 없습니다. 예를 들어 **TestRepo** 리포지토리의 런타임 개체에 대한 작성, 편집 및 삭제 권한을 가지고 있습니다. **TestRepo** 리포지토리와 연결된 웹 서비스 힙에 대한 웹 서비스 힙 콘솔에서 **TestRepo** 리포지토리의 모든 웹 서비스를 볼 수 있습니다. 다른 사용자가 작성한 **TestRepo** 리포지토리의 웹 서비스는 볼 수 있지만 실행할 수는 없습니다.

다음 테이블에는 웹 서비스 섹션에서 사용할 수 있는 옵션이 설명되어 있습니다.

레이블	설명
시험판	선택한 웹 서비스를 테스트하는 데 사용할 수 있는 클라이언트 응용 프로그램입니다. 선택한 웹 서비스를 실행하려면 클릭합니다. 올바르지 않은 웹 서비스에는 사용할 수 없습니다.
WSDL	선택한 웹 서비스에 대한 WSDL입니다. 선택한 웹 서비스에 대한 WSDL을 보려면 클릭합니다. 섹션의 맨 위 또는 선택한 웹 서비스와 같은 행에 있는 WSDL 단추를 클릭할 수 있습니다. WSDL을 다운로드하려면 WSDL을 확인하고 로컬 시스템에 저장합니다. 올바르지 않은 웹 서비스에는 사용할 수 없습니다.
검색	웹 서비스를 검색합니다. 검색하려는 텍스트 문자열을 입력하고 실행을 클릭합니다. 텍스트를 포함하는 모든 웹 서비스 이름, 리포지토리 이름 또는 워크플로우 이름이 웹 서비스 섹션에 나열됩니다.
서비스 이름	웹 서비스 힙에서 실행할 수 있는 웹 서비스의 이름입니다.
리포지토리 이름	웹 서비스와 연결된 리포지토리의 이름입니다.
워크플로우 이름	웹 서비스를 구성하는 워크플로우의 이름입니다.

## 작업 섹션

작업 섹션에는 웹 서비스 힙에서 사용 가능한 일괄 웹 서비스 작업 목록이 표시됩니다. 클라이언트 응용 프로그램에서 이러한 작업을 호출하여 워크플로우를 실행 및 모니터링하고 **PowerCenter** 메타데이터에 액세스할 수 있습니다.

탐색기에서 메타데이터 웹 서비스를 선택하면 사용 가능한 메타데이터 웹 서비스 작업 목록이 작업 섹션에 표시됩니다. 탐색기에서 데이터 통합 웹 서비스를 선택하면 사용 가능한 데이터 통합 웹 서비스 작업 목록이 작업 섹션에 표시됩니다.



다음 테이블에는 작업 섹션의 아이콘과 정보가 설명되어 있습니다.

레이블	설명
시험판	선택한 작업을 테스트하는 응용 프로그램입니다. 선택한 메타데이터 웹 서비스 또는 데이터 통합 웹 서비스 작업을 실행하려면 클릭합니다.
WSDL	메타데이터 또는 데이터 통합 웹 서비스용 WSDL입니다. 선택한 웹 서비스 유형에 대해 웹 서비스 협에서 게시한 WSDL을 보려면 클릭합니다. 탐색기에서 메타데이터 웹 서비스를 선택하는 경우 WSDL을 클릭하여 메타데이터 웹 서비스용 WSDL을 볼 수 있습니다. 탐색기에서 데이터 통합 웹 서비스를 선택하는 경우 WSDL을 클릭하여 데이터 통합 웹 서비스용 WSDL을 볼 수 있습니다. WSDL을 다운로드하려면 WSDL을 확인하고 로컬 시스템에 저장합니다.
검색	작업을 검색합니다. 검색하려는 텍스트 문자열을 입력하고 실행을 클릭합니다. 텍스트를 포함하는 모든 작업 이름 또는 설명이 작업 섹션에 나열됩니다.
작업 이름	메타데이터 웹 서비스 또는 데이터 통합 웹 서비스 작업의 이름입니다.
설명	메타데이터 웹 서비스 또는 데이터 통합 웹 서비스 작업의 설명입니다.

## 속성 섹션

속성 섹션에는 웹 서비스 또는 작업 섹션에서 선택한 웹 서비스 또는 웹 서비스 작업에 대한 정보가 표시됩니다.

### 실시간 웹 서비스의 속성 섹션

웹 서비스 섹션에서 올바른 실시간 웹 서비스 또는 올바르지 않은 실시간 웹 서비스를 선택하면 선택한 웹 서비스의 속성이 속성 섹션에 표시됩니다.

다음 테이블에는 실시간 웹 서비스 속성이 설명되어 있습니다.

속성	설명
서비스 이름	웹 서비스의 이름입니다.
도메인 이름	웹 서비스 협이 포함된 PowerCenter 도메인의 이름입니다.
리포지토리 이름	웹 서비스 워크플로우가 포함된 리포지토리입니다.
폴더 이름	웹 서비스 워크플로우가 포함된 폴더의 이름입니다.
워크플로우 이름	웹 서비스와 연결된 워크플로우의 이름입니다.
설명	웹 서비스에 대한 설명입니다.
실행 가능	클라이언트 응용 프로그램이 웹 서비스를 시작할 수 있는지 여부를 나타냅니다. True인 경우 웹 서비스 클라이언트가 웹 서비스 워크플로우를 시작하거나 워크플로우가 실행 중인 동안 웹 서비스를 호출할 수 있습니다. False인 경우 워크플로우가 실행 중인 동안 웹 서비스 클라이언트가 웹 서비스를 호출할 수 있지만 워크플로우를 시작할 수 없습니다.

속성	설명
보호됨	웹 서비스가 보호되거나 공용으로 사용되는지를 나타냅니다. True인 경우 웹 서비스 클라이언트 요청은 인증을 통과해야 합니다. SOAP 요청은 올바른 사용자 이름 토큰을 헤더에 포함하고 있어야 합니다. False인 경우 모든 웹 서비스 클라이언트가 인증 없이 웹 서비스 요청을 실행할 수 있습니다.
단방향	웹 서비스가 단방향 또는 요청-응답 매핑을 사용하는지를 나타냅니다.

## 일괄 웹 서비스의 속성 섹션

작업 섹션에서 메타데이터 웹 서비스 작업 또는 데이터 통합 웹 서비스 작업을 선택하면 선택한 웹 서비스 작업의 이름과 설명이 속성 섹션에 표시됩니다.

# 웹 서비스 테스트

시험관 응용 프로그램은 웹 서비스 헵 콘솔에 나열된 실시간 또는 일괄 웹 서비스 작업을 실행하는 데 사용할 수 있는 클라이언트 응용 프로그램입니다. 웹 서비스 헵 콘솔에서 시험관 응용 프로그램을 사용하여 올바른 웹 서비스 작업을 테스트하고 결과를 봅니다. 시험관 응용 프로그램은 입력 메시지에 필요한 매개 변수를 확실히 알지 못하거나 특정 입력 메시지에 대한 응답을 보려는 경우에 사용할 수 있습니다.

시험관 응용 프로그램을 사용하면 WSDL을 다운로드할 필요 없이 작업을 호출하거나 웹 서비스를 실행할 수 있으며 클라이언트 응용 프로그램에 대한 클라이언트 프록시 클래스를 생성할 수 있습니다. 콘솔에서 응답을 보고 클라이언트 응용 프로그램이 웹 서비스의 응답을 처리해야 하는 방법을 결정할 수 있습니다.

올바른 실시간 웹 서비스 응용 프로그램 또는 일괄 웹 서비스 작업을 테스트할 수 있습니다. 시험관 응용 프로그램은 SOAP 첨부 파일이 포함된 WSDL을 사용하여 웹 서비스를 테스트하는 데 사용할 수 없습니다.

보안 실시간 웹 서비스를 사용하려면 인증이 필요합니다. 보안 웹 서비스 작업을 테스트하려면 올바른 사용자 이름 토큰을 제공하여 PowerCenter 리포지토리에 로그인합니다.

## 입력 메시지

시험관 응용 프로그램은 웹 서비스 요청을 작성하는 두 가지 방법을 제공합니다.

- XML 입력
- 양식 입력

요청 요구 사항에 가장 잘 맞는 방법을 사용합니다. 예를 들어 다중 발생 요소가 요청에 포함되어 있는 경우 XML 입력을 사용하여 요청 메시지를 작성합니다.

## XML 입력

XML 입력 탭을 선택하면 웹 서비스 헵이 서비스 요청 작업을 실행하는 데 필요한 요소가 포함된 SOAP 입력 메시지를 표시합니다. SOAP 메시지에 요소 값을 입력합니다. 또는 웹 서비스 헵 콘솔 외부에서 SOAP 메시지를 작성하여 XML 입력 섹션에 붙여넣을 수 있습니다.

웹 서비스 헵은 SOAP 입력 메시지를 사용하여 웹 서비스를 실행합니다. SOAP 출력 메시지로 응답이 표시됩니다.

## 양식 입력

양식 입력 탭을 선택하면 웹 서비스 협이 웹 서비스 요청에 대한 매개 변수 목록을 표시합니다. 매개 변수 값을 입력합니다. 웹 서비스 요청에 복합 유형 요소가 포함되는 경우 올바른 계층의 입력 매개 변수가 양식 입력 탭에 표시됩니다.

웹 서비스 협은 사용자가 입력하는 매개 변수 값을 사용하여 SOAP 입력 메시지를 작성하고 웹 서비스를 실행합니다. SOAP 출력 메시지로 응답이 표시됩니다.

## 공용 웹 서비스 또는 일괄 웹 서비스 작업 테스트

공용 또는 일괄 웹 서비스를 테스트하려면 웹 서비스 작업을 선택하고 웹 서비스 작업의 입력 메시지에 매개 변수 값을 입력합니다.

공용 웹 서비스 또는 일괄 웹 서비스 작업을 테스트하려면 다음을 수행합니다.

1. 웹 서비스 또는 작업 섹션에서 올바른 실시간 웹 서비스 또는 작업을 선택합니다.
2. 시험관을 클릭합니다.  
테스트할 수 있는 웹 서비스 작업 목록과 시험관 응용 프로그램의 실행 방법에 대한 지침이 시험관 응용 프로그램 창에 표시됩니다.
3. 테스트하려는 작업을 선택합니다.  
입력 메시지에 대한 매개 변수가 시험관 응용 프로그램 창에 표시됩니다.
4. XML 입력 탭을 클릭하여 SOAP 메시지 형식으로 입력 매개 변수를 입력합니다.  
또는 양식 입력 탭을 클릭하여 매개 변수 입력 양식으로 입력 매개 변수를 입력합니다.
5. 매개 변수 값을 입력합니다.  
WSDL에는 사용자 정의 데이터 유형이 포함될 수 있습니다. 오류 응답이 발생하지 않게 하려면 데이터 유형에 따라 매개 변수 값을 입력합니다.
6. 보내기를 클릭합니다.  
웹 서비스 협이 웹 서비스 작업을 실행하고 SOAP 메시지 응답과 성공 또는 실패를 나타내는 메시지를 표시합니다.
7. 매개 변수를 지우고 새 값을 입력하려면 재설정을 클릭합니다.
8. 웹 브라우저의 닫기 단추를 클릭하여 시험관 응용 프로그램을 종료하고 웹 서비스 협 콘솔의 기본 페이지로 돌아갑니다.

## 보안 실시간 웹 서비스 테스트

보안 실시간 웹 서비스를 테스트하려면 사용자 이름 토큰을 SOAP 헤더에 포함시킵니다. 양식 입력 탭에 사용자 이름 및 암호를 입력하거나 SOAP 메시지를 수정하여 사용자 이름 토큰의 모든 요소를 XML 입력 탭에 포함시킬 수 있습니다.

양식 입력 탭 또는 XML 입력 탭의 일반 텍스트 또는 해시된 암호를 사용하여 보안 웹 서비스를 테스트할 수 있습니다. 해시된 암호를 사용하여 보안 웹 서비스를 테스트하려면 웹 서비스를 테스트하기 전에 MD5 또는 SHA-1 해시 함수를 사용하여 암호를 암호화합니다. 암호화는 Base64로 인코딩해야 합니다. 암호화를 통해 해시된 값을 웹 서비스의 암호로 사용합니다.

XML 입력 탭의 요약 암호를 사용하여 보안 웹 서비스를 테스트할 수 있습니다. 요약 암호를 사용하여 보안 웹 서비스를 테스트하려면 UsernameToken 요소에 필요한 Password 특성 및 요소를 요약 암호에 대해 추가합니다.

보안 웹 서비스를 테스트하려면 다음을 수행합니다.

1. 실시간 웹 서비스의 웹 서비스 섹션에서 실행할 보안 웹 서비스를 선택하고 시험관을 클릭합니다.
2. 시험관 응용 프로그램 창에서 보안 웹 서비스에 대한 작업을 선택합니다.

3. 양식 입력을 사용하여 웹 서비스 작업을 테스트하려면 양식 입력 탭을 클릭합니다.

SOAP 헤더 섹션에 사용자 이름과 일반 텍스트 또는 해시된 암호를 입력합니다.

SOAP 본문 섹션에 보안 웹 서비스에 필요한 매개 변수 값을 입력합니다.

-또는-

XML 입력을 사용하여 웹 서비스 작업을 테스트하려면 XML 입력 탭을 클릭하고 UsernameToken 요소를 업데이트합니다.

일반 텍스트 또는 해시된 암호를 사용하는 보안 웹 서비스를 테스트하려면 Username 및 Password 하위 요소의 *[string]* 값을 올바른 사용자 이름과 암호로 바꿉니다.

```
<UsernameToken>
  <Username>[string]</Username>
  <Password>[string]</Password>
</UsernameToken>
```

요약 암호를 사용하는 보안 웹 서비스를 테스트하려면 Username 요소의 *[string]* 값을 올바른 사용자 이름으로 바꿉니다. Password 요소를 업데이트하고 Nonce 및 Created 요소를 적절한 값과 함께 추가합니다.

```
<UsernameToken>
  <Username>[string]</Username>
  <Password Type="PasswordDigest">[string]</Password>
  <Nonce>[NonceValue]</Nonce>
  <Created>[RequestCreationTimestamp]</Created>
</UsernameToken>
```

UsernameToken 요소에 대한 자세한 내용은 [“SOAP 요청의 UsernameToken” 페이지 70](#)을 참조하십시오.

SOAP 본문 섹션에 보안 웹 서비스에 필요한 매개 변수 값을 입력합니다.

4. 보내기를 클릭합니다.

웹 서비스 협이 보안 웹 서비스 작업을 실행하고 SOAP 메시지 응답을 콘솔에 표시합니다.

5. 웹 브라우저의 닫기 단추를 클릭하여 시험판 응용 프로그램을 종료하고 웹 서비스 협 콘솔의 기본 페이지로 돌아갑니다.

## 제 4 장

# 일괄 웹 서비스 작업

이 장에 포함된 항목:

- [일괄 웹 서비스 작업 개요, 29](#)
- [메타데이터 웹 서비스 작업, 29](#)
- [데이터 통합 웹 서비스 작업, 32](#)

## 일괄 웹 서비스 작업 개요

일괄 웹 서비스 작업을 사용하여 기존 워크플로우 및 태스크를 예약, 시작 또는 중지할 수 있습니다. 세션 통계와 성능 데이터를 가져올 수 있습니다. 워크플로우 및 세션 로그를 검색할 수 있습니다.

일괄 웹 서비스는 개별 WSDL에 정의된 다음 서비스 그룹으로 구성됩니다.

- **메타데이터 웹 서비스.** 메타데이터 웹 서비스에 대한 작업은 웹 서비스 헵 콘솔의 일괄 웹 서비스 페이지에 제공되는 메타데이터 WSDL에 정의됩니다.
- **데이터 통합 웹 서비스.** 데이터 통합 웹 서비스에 대한 작업은 웹 서비스 헵 콘솔의 일괄 웹 서비스 페이지에 제공되는 데이터 통합 WSDL에 정의됩니다.

이 장에서는 일괄 웹 서비스에서 제공하는 작업에 대해 설명합니다. 이러한 작업에 대한 요청 및 응답 XML 문서와 관련된 자세한 내용은 WSDL 파일을 참조하십시오.

**참고:** 일괄 웹 서비스 작업 호출에서 가져오는 로그 세그먼트는 통합 서비스 코드 페이지 또는 UTF-8 중 하나로 되어 있습니다.

## 메타데이터 웹 서비스 작업

메타데이터 웹 서비스에서 제공하는 작업을 사용하여 웹 서비스 헵과 연결된 PowerCenter 리포지토리에서 메타데이터를 검색합니다.

이러한 작업을 사용하여 리포지토리에 로그인하고 다음 리포지토리 개체의 목록을 가져올 수 있습니다.

- 웹 서비스 헵과 연결된 리포지토리의 모든 폴더
- 폴더의 모든 워크플로우
- 워크플로우의 모든 **worklet** 및 세션 태스크
- 리포지토리와 연결된 모든 통합 서비스
- 웹 서비스 헵과 연결된 모든 리포지토리

이 섹션에는 메타데이터 웹 서비스에 사용할 수 있는 모든 작업이 나열되어 있습니다.

## getAllIServers

리포지토리와 연결된 모든 통합 서비스의 이름을 검색합니다.

2개 이상의 통합 서비스를 리포지토리와 연결하여 워크플로우 및 세션을 실행할 수 있습니다. 다중 통합 서비스 환경에서는 연결된 각 서비스에 대해 설명적인 서비스 이름을 입력하여 사용자가 다양한 통합 서비스를 명확하게 구별할 수 있도록 도와주는 것이 중요합니다. 리포지토리와 연결된 각 통합 서비스는 서비스 이름뿐만 아니라 리포지토리와 연결된 서비스 간에 고유한 호스트 이름 및 포트 번호의 조합을 가지고 있어야 합니다.

이 작업은 지정한 리포지토리와 연결된 모든 통합 서비스의 이름을 반환합니다.

`getAllIServers` 작업은 `SessionID` 매개 변수와 함께 호출합니다. 세션 ID는 리포지토리에 로그인하면 생성됩니다.

## getAllFolders

리포지토리의 모든 폴더를 검색합니다.

`getAllFolders` 작업은 `SessionID` 매개 변수와 함께 호출합니다. 세션 ID는 리포지토리에 로그인하면 생성됩니다.

## getAllRepositories

웹 서비스 헵과 연결된 모든 리포지토리를 봅니다. `getAllRepositories` 작업에는 매개 변수가 필요하지 않습니다.

웹 서비스 헵 클라이언트 응용 프로그램에서 리포지토리를 사용할 수 있게 하려면 리포지토리를 웹 서비스 헵과 연결해야 합니다. `Administrator` 도구를 사용하여 리포지토리를 웹 서비스 헵과 연결할 수 있습니다.

**참고:** `getAllRepositories` 작업은 특정 리포지토리와 연결되지 않으므로 이 작업을 사용하기 위해 리포지토리에 로그인하지 않아도 됩니다. `Login` 작업을 호출하지 않고 `getAllRepositories` 작업을 호출할 수 있습니다.

## getAllTaskInstances

워크플로우에 있는 지정한 깊이의 모든 `worklet` 및 세션 태스크 인스턴스에 대한 정보를 가져옵니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
깊이	필수	워크플로우 태스크 계층 내에서 태스크 인스턴스에 대한 정보를 가져오는 수준 개수입니다.
WorkflowName	필수	worklet 및 세션이 포함된 워크플로우의 이름입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
IsValid	선택 사항	올바른 태스크 인스턴스를 가져올지 아니면 올바르지 않은 태스크 인스턴스를 가져올지를 지정합니다. 올바른 태스크 인스턴스를 가져오려면 <code>True</code> 로 설정합니다.

## getAllWorkflows

폴더의 모든 워크플로우에 대한 정보를 가져옵니다. 워크플로우는 세션, 전자 메일 알림 및 셀 명령 같은 태스크를 실행하는 방식을 통합 서비스에 알려 주는 지침의 집합입니다. 워크플로우 정보에는 워크플로우의 이름, 워크플로우가 있는 폴더의 이름 및 워크플로우가 올라온지 여부가 포함됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
이름	필수	워크플로우가 포함된 폴더의 이름입니다.

## Login

리포지토리에 로그인합니다. **Login** 작업은 지정한 리포지토리에 대해 사용자 이름 및 암호를 인증합니다. 클라이언트 응용 프로그램은 다른 작업을 호출하기 전에 이 작업을 호출해야 합니다. **Login** 작업을 호출한 후 웹 서비스 클라이언트 응용 프로그램은 모든 일괄 웹 서비스 작업을 호출할 수 있습니다.

**Login** 작업에는 리포지토리 이름, 사용자 이름 및 암호가 필요합니다. 이 작업은 암호화된 세션 ID를 반환합니다. 리포지토리는 웹 서비스 협과 동일한 도메인에 있어야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
RepositoryName	필수	로그인할 리포지토리의 이름입니다.
UserName	필수	리포지토리에 로그인하는 데 사용하는 사용자 이름입니다.
암호	필수	리포지토리에 로그인하는 데 사용하는 사용자 계정의 암호입니다.
RepositoryDomainName	선택 사항	리포지토리 서비스의 도메인 이름입니다.
UserNameSpace	조건	리포지토리에 로그인하는 데 사용하는 사용자 계정의 보안 도메인입니다. PowerCenter 도메인에 2개 이상의 보안 도메인이 있는 경우 필수입니다.

### 세션 만료 기간

세션 ID는 일정 시간 동안 작업을 수행하지 않으면 만료됩니다. 웹 서비스 협의 **SessionExpiryPeriod** 고급 속성을 구성하여 세션이 만료되기 전에 작업을 수행하지 않아도 되는 시간을 지정합니다. **Administrator** 도구에서 웹 서비스 협 고급 속성을 구성할 수 있습니다.

웹 서비스 협은 세션 시작 시 사용자 이름 및 암호를 캐시합니다. 웹 서비스 협은 세션이 만료될 때까지 캐시된 사용자 이름 및 암호를 로그인 인증에 사용합니다. 사용자 계정에 대한 변경 내용은 세션이 만료될 때까지 인증에 영향을 주지 않습니다. 사용자 계정을 삭제하거나 비활성화하는 경우 세션이 만료될 때까지 사용자 계정이 삭제되거나 비활성화되지 않습니다. 사용자 계정이 잠기는 경우 잠긴 사용자는 세션이 만료될 때까지 요청을 보낼 수 있습니다.

**참고:** **SessionExpiryPeriod** 값이 큰 경우 사용자 계정 변경 내용이 적용되려면 일정 시간이 소요됩니다. 보안 위반을 방지하려면 사용자 계정이 세션에서 사용 중일 때 해당 계정을 삭제하거나 수정하지 않아야 합니다.

## Logout

리포지토리에서 로그아웃합니다. **Logout** 작업은 리포지토리와 통합 서비스에서 연결을 끊습니다. 클라이언트 응용 프로그램 실행 종료 시 이 작업을 호출하여 리소스를 해제하고 리포지토리 서비스 및 웹 서비스 힙 프로세스에서 발생할 수 있는 메모리 누수를 예방합니다.

**Logout** 작업은 **SessionID** 매개 변수와 함께 호출합니다. 세션 ID는 리포지토리에 로그인하면 생성됩니다.

## 데이터 통합 웹 서비스 작업

데이터 통합 웹 서비스 작업을 사용하여 다음 작업을 수행할 수 있습니다.

- **리포지토리 서비스에 연결.** 다음 작업을 사용하여 리포지토리에 로그인하거나 리포지토리에서 로그아웃할 수 있습니다.
  - Login
  - Logout
- **통합 서비스에 연결하여 해당 서비스 관련 세부 정보 가져오기.** 다음 작업을 사용하여 통합 서비스가 실행 중인지 확인하고 통합 서비스에 연결하여 해당 서비스 관련 정보를 가져올 수 있습니다.
  - pingDI Server
  - getDI ServerProperties
  - initializeDI ServerConnection(더 이상 사용되지 않음)
  - deinitializeDI ServerConnection(더 이상 사용되지 않음)
- **워크플로우 예약 및 실행.** 다음 작업을 사용하여 워크플로우 실행을 관리할 수 있습니다.
  - startWorkflow
  - startWorkflowEx
  - stopWorkflow
  - scheduleWorkflow
  - startWorkflowFromTask
  - unscheduleWorkflow
  - waitTillWorkflowComplete
- **워크플로우의 태스크 시작 및 중지.** 다음 작업을 사용하여 워크플로우의 태스크를 제어할 수 있습니다.
  - recoverWorkflow
  - resumeWorkflow(더 이상 사용되지 않음)
  - startTask
  - stopTask
  - waitTillTaskComplete
- **세션에 대한 통계 가져오기.** 다음 작업을 사용하여 세션 또는 워크플로우 실행에 대한 세부 정보를 가져올 수 있습니다.
  - getNextLogSegment(더 이상 사용되지 않음)
  - getSessionLog
  - getSessionPerformanceData
  - getSessionStatistics



- getTaskDetails
- getTaskDetailsEx
- getWorkflowDetails
- getWorkflowDetailsEx
- getWorkflowLog
- monitorDIServer
- startSessionLogFetch(더 이상 사용되지 않음)
- startWorkflowLogFetch(더 이상 사용되지 않음)

데이터 통합 웹 서비스 작업은 **di.wsdl**에 정의됩니다.

## deinitializeDIServerConnection

더 이상 사용되지 않은 작업입니다. 명시적으로 통합 서비스에서 클라이언트 응용 프로그램의 연결을 끊지 않아도 됩니다. **Logout** 작업은 클라이언트 응용 프로그램에서 획득한 통합 서비스에 대한 연결을 해제하고 정리 작업을 수행합니다.

이 작업을 사용하려면 **initializeDIServerConnection** 작업과 **SessionID** 매개 변수가 필요합니다. 세션 ID는 리포지토리에 로그인하면 생성됩니다.

## getDIServerProperties

통합 서비스의 속성을 가져옵니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

통합 서비스 속성에는 다음 정보가 포함됩니다.

- 통합 서비스 이름
- 통합 서비스 버전
- 제품 이름
- 통합 서비스 시작 시간
- 통합 서비스와 연결된 리포지토리의 이름
- 데이터 이동 모드(ASCII 또는 유니코드)
- 통합 서비스가 매핑을 디버그할 수 있는지 여부

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
ServiceName	필수	속성을 가져올 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getNextLogSegment

더 이상 사용되지 않은 작업입니다. 이 작업은 세션 또는 워크플로우 로그의 정보를 증분적으로 가져옵니다.

**getNextLogSegment** 작업은 세션 또는 워크플로우 로그의 일부를 반환합니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

이 작업은 `startSessionLogFetch` 또는 `startWorkflowLogFetch` 작업과 함께 사용됩니다. 로그의 끝에 도달할 때까지 `startSessionLogFetch` 또는 `startWorkflowLogFetch` 작업에서 생성한 로그 핸들을 사용하여 `getNextLogSegment` 작업을 호출합니다.

하나의 작업으로 세션 로그 정보를 가져오려면 `getSessionLog` 작업을 사용합니다. 하나의 작업으로 워크플로우 로그 정보를 가져오려면 `getWorkflowLog` 작업을 사용합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
LogHandle	필수	<code>startSessionLogFetch</code> 또는 <code>startWorkflowLogFetch</code> 작업에서 생성한 로그 ID입니다. 이 작업을 호출하기 전에 <code>startSessionLogFetch</code> 또는 <code>startWorkflowLogFetch</code> 작업을 호출해야 합니다.
제한 시간	필수	이 작업을 수행하는 동안 클라이언트 요청이 웹 서비스 협과 연결 상태를 유지하는 시간입니다. 제한 시간 내에 작업이 완료되지 않으면 작업이 실패합니다. 제한 시간은 <code>startSessionLogFetch</code> 또는 <code>startWorkflowLogFetch</code> 작업을 호출할 때부터 시작됩니다.

## getSessionLog

하나의 작업으로 세션 로그의 모든 정보를 가져옵니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

서비스 세션이 실행되면 통합 서비스에서 프로세스 초기화, 세션 유효성 검사, 관독기 및 기록기 스레드에 대한 SQL 명령 작성, 발생한 오류 및 로드 요약과 같은 정보를 세션 로그에 기록합니다. 세션 로그의 세부 정보 양은 설정하는 추적 수준에 따라 다릅니다. `getSessionLog` 작업은 세션 로그의 정보를 반환합니다.

중분적으로 세션 로그 정보를 가져오려면 `getNextLogSegment` 작업을 사용합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	세션이 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>WorkletName.TaskName</i> 을 입력합니다. 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
제한 시간	필수	이 작업을 수행하는 동안 클라이언트 요청이 웹 서비스 협과 연결 상태를 유지하는 시간입니다. 제한 시간 내에 작업이 완료되지 않으면 작업이 실패합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getSessionPerformanceData

통합 서비스에서 실행 중인 세션의 성능 데이터를 검색합니다. 성능 세부 정보에는 세션 및 매핑 효율성을 이해하는 데 도움을 주는 카운터가 제공됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	세션이 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	세션의 위치를 지정하는 경로입니다. 세션이 워크플로우 내에 있는 경우 세션 이름만 입력합니다. 세션이 worklet 내에 있는 경우 <i>WorkletName.SessionName</i> 을 입력합니다. 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getSessionStatistics

통합 서비스에서 실행 중인 세션의 통계를 가져옵니다. 세션이 실행 중이지 않으면 이 작업은 가장 최근에 실행된 세션의 통계를 제공합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

세션 통계에는 폴더 및 워크플로우 이름, 세션 및 태스크 실행 상태, 오류 정보, 소스 및 대상에 대해 성공하고 실패한 행 수, 적용된 행 수, 거부된 행 수 및 영향을 받은 행 수가 포함됩니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	세션 태스크가 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	세션의 위치를 지정하는 경로입니다. 세션이 워크플로우 내에 있는 경우 세션 이름만 입력합니다. 세션이 worklet 내에 있는 경우 <i>WorkletName.SessionName</i> 을 입력합니다. 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getTaskDetails

통합 서비스에서 태스크의 세부 정보를 검색합니다. 상위 워크플로우가 실행 중이고 태스크가 이미 실행된 경우 이 작업은 실행 중인 워크플로우의 현재 태스크에 대한 세부 정보를 반환합니다. 상위 워크플로우가 실행 중이지 않은 경우 이 작업은 마지막으로 실행된 워크플로우의 태스크 세부 정보를 반환합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

태스크 세부 정보에는 폴더 및 워크플로우 이름, 태스크 이름 및 유형, 시작 시간, 실행 상태, 실행 오류 코드 및 메시지가 포함됩니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	태스크가 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 ID입니다. 이 매개 변수는 사용할 수 없습니다.
WorkflowRunInstanceName	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 이름입니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
RequestMode	선택 사항	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. <ul style="list-style-type: none"><li>- NORMAL. 복구하지 않고 세션을 다시 시작합니다.</li><li>- RECOVERY. 세션을 복구합니다.</li></ul>
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## getTaskDetailsEx

워크플로우의 다중 인스턴스가 동시에 실행되는 경우 통합 서비스에서 태스크의 세부 정보를 검색합니다.

**getTaskDetailsEx** 작업은 **getTaskDetails** 작업과 유사하지만 태스크의 모든 인스턴스에 대한 정보를 반환합니다. 상위 워크플로우가 실행 중이고 태스크가 이미 실행된 경우 이 작업은 실행 중인 워크플로우에 있는 태스크의 모든 인스턴스에 대한 세부 정보를 반환합니다. 상위 워크플로우가 실행 중이지 않은 경우 이 작업은 마지막

으로 실행된 워크플로우의 태스크 인스턴스에 대한 세부 정보를 반환합니다. 태스크 세부 정보는 작업에서 반환한 통합 서비스 세부 정보에 포함됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	태스크가 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 worklet 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getWorkflowDetails

지정된 워크플로우의 세부 정보를 가져옵니다. 워크플로우가 실행 중인 경우 이 작업은 실행 중인 워크플로우의 세부 정보를 반환합니다. 워크플로우가 실행 중이지 않은 경우 이 작업은 이 워크플로우의 마지막 실행에 대한 세부 정보를 반환합니다.

워크플로우 세부 정보에는 폴더, 워크플로우 및 워크플로우 로그 파일의 이름과 워크플로우를 실행하는 사용자의 이름이 포함됩니다. 워크플로우 실행 유형, 로그 파일 코드 페이지, 시작 및 종료 시간, 실행 통계, 실행 오류 코드 및 메시지도 포함됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	워크플로우 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.

매개 변수 이름	필수/선택	설명
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## getWorkflowDetailsEx

워크플로우의 다중 인스턴스가 동시에 실행되는 경우 워크플로우의 세부 정보를 가져옵니다.

이 작업은 `getWorkflowDetails` 작업과 유사하지만 모든 워크플로우 인스턴스에 대한 정보를 반환합니다. 워크플로우가 실행 중인 경우 이 작업은 실행 중인 워크플로우의 모든 인스턴스에 대한 세부 정보를 반환합니다. 워크플로우가 실행 중이지 않은 경우 이 작업은 워크플로우의 마지막 실행에 대한 세부 정보를 반환합니다. 워크플로우 세부 정보는 작업에서 반환한 통합 서비스 세부 정보에 포함됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.

매개 변수 이름	필수/선택	설명
WorkflowName	필수	워크플로우 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## getWorkflowLog

하나의 작업으로 워크플로우 로그의 모든 정보를 가져옵니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

웹 서비스 워크플로우가 실행되면 통합 서비스에서 프로세스 초기화, 워크플로우 태스크 실행 정보, 발생한 오류 및 워크플로우 실행 요약과 같은 정보를 워크플로우 로그에 기록합니다. 워크플로우 로그의 세부 정보 양은 추적 수준에 따라 다릅니다. `getWorkflowLog` 작업은 워크플로우 로그의 정보를 반환합니다.

충분적으로 워크플로우 로그 정보를 가져오려면 `getNextLogSegment` 작업을 사용합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	워크플로우 이름입니다.
제한 시간	필수	이 작업을 수행하는 동안 클라이언트 요청이 웹 서비스 heap과 연결 상태를 유지하는 시간입니다. 제한 시간 내에 작업이 완료되지 않으면 작업이 실패합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## initializeDlServerConnection

더 이상 사용되지 않은 작업입니다. 통합 서비스에 대한 연결을 초기화하지 않아도 됩니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
DIServerName	필수	워크플로우가 포함된 폴더의 이름입니다.
LoginHandle	선택 사항	세션 ID와 동일합니다.
DIServerDomain	선택 사항	통합 서비스의 도메인 이름입니다.

## Login

**Login** 작업은 데이터 통합 웹 서비스와 메타데이터 웹 서비스에 포함되어 있습니다.

리포지토리에 로그인합니다. **Login** 작업은 지정한 리포지토리에 대해 사용자 이름 및 암호를 인증합니다. 클라이언트 응용 프로그램은 다른 작업을 호출하기 전에 이 작업을 호출해야 합니다. **Login** 작업을 호출한 후 웹 서비스 클라이언트 응용 프로그램은 모든 일괄 웹 서비스 작업을 호출할 수 있습니다.

**Login** 작업에는 리포지토리 이름, 사용자 이름 및 암호가 필요합니다. 이 작업은 암호화된 세션 ID를 반환합니다. 리포지토리는 웹 서비스 협과 동일한 도메인에 있어야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
RepositoryName	필수	로그인할 리포지토리의 이름입니다.
UserName	필수	리포지토리에 로그인하는 데 사용하는 사용자 이름입니다.
암호	필수	리포지토리에 로그인하는 데 사용하는 사용자 계정의 암호입니다.
RepositoryDomainName	선택 사항	리포지토리 서비스의 도메인 이름입니다.
UserNameSpace	조건	리포지토리에 로그인하는 데 사용하는 사용자 계정의 보안 도메인입니다. PowerCenter 도메인에 2개 이상의 보안 도메인이 있는 경우 필수입니다.

## 세션 만료 기간

세션 ID는 일정 시간 동안 작업을 수행하지 않으면 만료됩니다. 웹 서비스 협의 **SessionExpiryPeriod** 고급 속성을 구성하여 세션이 만료되기 전에 작업을 수행하지 않아도 되는 시간을 지정합니다. **Administrator** 도구에서 웹 서비스 협 고급 속성을 구성할 수 있습니다.

웹 서비스 협은 세션 시작 시 사용자 이름 및 암호를 캐시합니다. 웹 서비스 협은 세션이 만료될 때까지 캐시된 사용자 이름 및 암호를 로그인 인증에 사용합니다. 사용자 계정에 대한 변경 내용은 세션이 만료될 때까지 인증에 영향을 주지 않습니다. 사용자 계정을 삭제하거나 비활성화하는 경우 세션이 만료될 때까지 사용자 계정이 삭제되거나 비활성화되지 않습니다. 사용자 계정이 잠기는 경우 잠긴 사용자는 세션이 만료될 때까지 요청을 보낼 수 있습니다.

**참고:** **SessionExpiryPeriod** 값이 큰 경우 사용자 계정 변경 내용이 적용되려면 일정 시간이 소요됩니다. 보안 위반을 방지하려면 사용자 계정이 세션에서 사용 중일 때 해당 계정을 삭제하거나 수정하지 않아야 합니다.



## Logout

Logout 작업은 데이터 통합 웹 서비스와 메타데이터 웹 서비스에 포함되어 있습니다.

리포지토리에서 로그아웃합니다. Logout 작업은 리포지토리와 통합 서비스에서 연결을 끊습니다. 클라이언트 응용 프로그램 실행 종료 시 이 작업을 호출하여 리소스를 해제하고 리포지토리 서비스 및 웹 서비스 힙 프로세스에서 발생할 수 있는 메모리 누수를 예방합니다.

Logout 작업은 SessionID 매개 변수와 함께 호출합니다. 세션 ID는 리포지토리에 로그인하면 생성됩니다.

## monitorDIServer

통합 서비스의 상태, 활성 및 예약 워크플로우의 세부 정보와 워크플로우 내의 태스크 및 링크에 대한 세부 정보를 검색합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
MonitorMode	필수	모니터링할 워크플로우의 모드입니다. <ul style="list-style-type: none"><li>- RUNNING. 활성 워크플로우의 상태 세부 정보를 반환합니다. 활성 워크플로우에는 실행 중인 워크플로우, 일시 중단된 워크플로우 및 일시 중단 중인 워크플로우가 포함됩니다.</li><li>- SCHEDULED. 예약된 워크플로우의 상태 세부 정보를 반환합니다.</li><li>- ALL. 모든 예약된 워크플로우 및 활성 워크플로우의 정보를 반환합니다.</li></ul>
ServiceName	필수	모니터링할 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## pingDIServer

통합 서비스가 실행 중인지 확인합니다. 반환 값은 ALIVE 또는 FAIL입니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
제한 시간	필수	이 작업을 수행하는 동안 클라이언트 요청이 웹 서비스 힙과 연결 상태를 유지하는 시간입니다. 제한 시간 내에 작업이 완료되지 않으면 작업이 실패합니다.
ServiceName	필수	ping을 수행할 통합 서비스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## recoverWorkflow

일시 중단된 워크플로우를 복구합니다. 통합 서비스에서 일시 중단되고 실패한 모든 **worklet** 및 일시 중단되고 실패한 명령, 전자 메일 및 세션 태스크의 워크플로우를 복구합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	복구할 워크플로우의 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. <ul style="list-style-type: none"><li>- NORMAL. 복구하지 않고 세션을 다시 시작합니다.</li><li>- RECOVERY. 세션을 복구합니다.</li></ul>
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.

매개 변수 이름	필수/선택	설명
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## resumeWorkflow

더 이상 사용되지 않은 작업입니다. `recoverWorkflow` 작업을 대신 사용합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	다시 시작할 워크플로우의 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.

매개 변수 이름	필수/선택	설명
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## scheduleWorkflow

워크플로우를 예약합니다. 요청 시 실행되고 있지 않은 모든 워크플로우를 예약할 수 있습니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	예약할 워크플로우의 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	예약할 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
RequestMode	선택 사항	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 worklet 내에 있는 경우 <WorkletName>을 입력합니다.<TaskName>. 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우를 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.

매개 변수 이름	필수/선택	설명
AttributeValue	선택 사항	워크플로우를 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## startSessionLogFetch

더 이상 사용되지 않은 작업입니다. 이 작업은 세션 로그의 정보를 증분적으로 가져오기 시작합니다.

**startSessionLogFetch** 작업은 **getNextLogSegment** 작업과 함께 사용하기 위한 로그 핸들을 생성합니다. **startSessionLogFetch** 작업을 호출한 후 로그의 끝에 도달할 때까지 **startSessionLogFetch**에서 생성한 로그 핸들을 사용하여 **getNextLogSegment** 작업을 호출합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	세션이 포함된 워크플로우의 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.

## startTask

워크플로우 내에서 특정 태스크를 시작합니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	태스크가 포함된 워크플로우의 이름입니다.

매개 변수 이름	필수/선택	설명
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 이름입니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## startWorkflow

워크플로우를 시작합니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	실행할 워크플로우의 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	실행할 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우를 시작하는 이유를 설명합니다.

매개 변수 이름	필수/선택	설명
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## startWorkflowEx

워크플로우를 시작합니다. **startWorkflowEx** 작업은 워크플로우의 실행 인스턴스 ID를 반환합니다. **startWorkflow** 작업 대신 **startWorkflowEx** 작업을 사용하여 작업에서 시작한 워크플로우의 실행 ID를 가져옵니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	실행할 워크플로우의 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. <ul style="list-style-type: none"> <li>- NORMAL. 복구하지 않고 세션을 다시 시작합니다.</li> <li>- RECOVERY. 세션을 복구합니다.</li> </ul>

매개 변수 이름	필수/선택	설명
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunInstanceName	선택 사항	실행할 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## startWorkflowFromTask

태스크에서 워크플로우를 시작합니다. 태스크에서 워크플로우를 시작하면 통합 서비스가 선택한 태스크부터 워크플로우의 끝까지 워크플로우를 실행합니다.

시작할 태스크에 대해 태스크 인스턴스 경로를 지정해야 합니다. 태스크 인스턴스 경로는 워크플로우 내부에서 태스크 인스턴스를 식별합니다. 워크플로우 내의 태스크는 해당 태스크 이름으로만 식별됩니다. *worklet* 내의 태스크는 마침표로 구분된 해당 *worklet* 및 태스크 이름으로 식별됩니다. *WorkletName.TaskName*. 예를 들어 워크플로우에 다른 *worklet* B를 포함하는 *worklet* A가 있습니다. 태스크 C는 *worklet* B 내의 태스크입니다. 태스크 C의 태스크 인스턴스 경로는 A.B.C입니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.



다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	실행할 워크플로우의 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	시작할 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우를 시작하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## startWorkflowLogFetch

더 이상 사용되지 않은 작업입니다. 이 작업은 워크플로우 로그의 정보를 증분적으로 가져옵니다.

`startWorkflowLogFetch` 작업은 `getNextLogSegment` 작업과 함께 사용하기 위한 로그 핸들을 생성합니다. `startWorkflowLogFetch` 작업을 호출한 후 로그의 끝에 도달할 때까지 `startWorkflowLogFetch`에서 생성한 로그 핸들을 사용하여 `getNextLogSegment` 작업을 호출합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	워크플로우 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.

## stopTask

통합 서비스에서 실행 중인 태스크를 중지합니다. 태스크, 워크플로우 또는 **worklet**은 언제든지 중지하거나 중단할 수 있습니다. 워크플로우의 태스크를 중지하면 통합 서비스에서 해당 태스크와 태스크 경로의 다른 모든 태스크 처리를 중지합니다.

`isAbort` 매개 변수를 **True**로 설정하여 실행 중인 태스크를 중단할 수도 있습니다. 중단할 태스크에 대한 태스크 인스턴스 경로를 지정합니다. 일반적으로 통합 서비스에서 태스크를 중지하지 못하는 경우에만 태스크를 중단합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	태스크가 포함된 워크플로우의 이름입니다.
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <b>worklet</b> 내에 있는 경우 <code>&lt;WorkletName&gt;</code> 을 입력합니다. <code>&lt;TaskName&gt;</code> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 이름입니다.

매개 변수 이름	필수/선택	설명
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
RequestMode	선택 사항	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 태스크를 중단하려면 이 매개 변수를 True로 설정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## 관련 항목:

- [“startWorkflowFromTask” 페이지 48](#)

## stopWorkflow

실행 중인 워크플로우를 중지합니다. 워크플로우를 중지하면 통합 서비스에서 워크플로우에서 현재 실행 중인 모든 태스크를 중지하려고 시도합니다. 워크플로우에 **worklet**이 포함되어 있으면 통합 서비스에서 **worklet**에서 현재 실행 중인 모든 태스크도 중지하려고 시도합니다.

워크플로우 중지 이외에도 **isAbort** 매개 변수를 **True**로 설정하여 실행 중인 워크플로우를 중단할 수 있습니다. 일반적으로 통합 서비스에서 워크플로우를 중지하지 못하는 경우에만 워크플로우를 중단합니다.

이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	중지할 워크플로우의 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	중지할 워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	중지할 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.

매개 변수 이름	필수/선택	설명
RequestMode	선택 사항	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	워크플로우를 중단할지 여부를 나타냅니다. 워크플로우를 중단하려면 이 매개 변수를 <i>True</i> 로 설정합니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## unscheduleWorkflow

워크플로우의 예약을 취소합니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	예약을 취소할 워크플로우의 이름입니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	예약을 취소할 워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	예약을 취소할 워크플로우 실행 인스턴스의 이름입니다.

매개 변수 이름	필수/선택	설명
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
RequestMode	선택 사항	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## waitTillTaskComplete

통합 서비스에서 실행 중인 태스크가 완료될 때까지 대기합니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	태스크가 포함된 워크플로우의 이름입니다.

매개 변수 이름	필수/선택	설명
TaskInstancePath	필수	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 <i>worklet</i> 내에 있는 경우 <i>&lt;WorkletName&gt;</i> 을 입력합니다. <i>&lt;TaskName&gt;</i> . 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	태스크가 포함된 워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## waitTillWorkflowComplete

다른 워크플로우가 완료된 후에만 작업 또는 워크플로우를 실행하려면 이 작업을 사용하여 다른 워크플로우가 완료되는 시기에 대한 알림을 받습니다. 이 작업을 호출하려면 먼저 리포지토리에 로그인해야 합니다.

다음 테이블에는 이 작업에 필요한 매개 변수가 설명되어 있습니다.

매개 변수 이름	필수/선택	설명
SessionID	필수	로그인 후 생성된 세션 ID입니다.
FolderName	필수	워크플로우가 포함된 폴더의 이름입니다.
WorkflowName	필수	워크플로우 이름입니다.
RequestMode	필수	다음과 같은 세션 태스크에 대한 복구 전략을 나타냅니다. - NORMAL. 복구하지 않고 세션을 다시 시작합니다. - RECOVERY. 세션을 복구합니다.
ServiceName	필수	워크플로우를 실행하는 통합 서비스의 이름입니다.
WorkflowRunId	선택 사항	워크플로우 실행 인스턴스의 ID입니다.
WorkflowRunInstanceName	선택 사항	워크플로우 실행 인스턴스의 이름입니다.
Reason	선택 사항	워크플로우 또는 태스크를 시작하는 이유를 설명합니다.
ParameterFileName	선택 사항	워크플로우 실행 시 사용할 매개 변수 파일의 이름입니다.
TaskInstancePath	선택 사항	태스크의 위치를 지정하는 경로입니다. 태스크가 워크플로우 내에 있는 경우 태스크 이름만 입력합니다. 태스크가 worklet 내에 있는 경우 <WorkletName>을 입력합니다.<TaskName>. 태스크 인스턴스 경로를 정규화된 문자열로 입력합니다.
IsAbort	선택 사항	태스크를 중단할지 여부를 나타냅니다. 이 작업에는 적용되지 않습니다.
OSUser	선택 사항	워크플로우에 할당된 운영 체제 프로필을 지정합니다.
DomainName	선택 사항	통합 서비스의 도메인 이름입니다.
AttributeName	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 이름입니다.
AttributeValue	선택 사항	워크플로우 또는 태스크를 시작하거나 예약하는 데 사용하는 특성 이름 및 값 쌍의 특성 값입니다.
키	선택 사항	워크플로우 또는 태스크를 시작하는 데 사용하는 키입니다.
MustUse	선택 사항	워크플로우 또는 태스크를 시작하는 데 키를 사용해야 하는지 여부를 나타냅니다.
ParameterScope	선택 사항	매개 변수 배열 정의의 매개 변수 범위입니다.
ParameterName	선택 사항	매개 변수 배열 정의의 매개 변수 이름입니다.
ParameterValue	선택 사항	매개 변수 배열 정의의 매개 변수 값입니다.

## 제 5 장

# 클라이언트 응용 프로그램 기록

이 장에 포함된 항목:

- [클라이언트 응용 프로그램 기록 개요, 56](#)
- [일괄 웹 서비스용 클라이언트 응용 프로그램, 57](#)
- [일괄 웹 서비스용 Java 클라이언트 응용 프로그램, 59](#)
- [일괄 웹 서비스용 C# 클라이언트 응용 프로그램, 61](#)
- [실시간 웹 서비스용 클라이언트 응용 프로그램, 64](#)
- [실시간 웹 서비스용 Java 클라이언트 응용 프로그램, 65](#)
- [매개 변수 배열 사용, 67](#)
- [클라이언트 요청에 보안 추가, 69](#)

## 클라이언트 응용 프로그램 기록 개요

이 장에서는 **PowerCenter** 웹 서비스 공급자가 제공하는 웹 서비스를 사용하기 위해 클라이언트 응용 프로그램을 기록하는 방법에 대한 개요를 제공합니다. 클라이언트 응용 프로그램을 작성하는 단계의 일반 설명은 **Java** 및 **.NET** 프레임워크를 사용하여 클라이언트 응용 프로그램을 작성하는 방법의 예제 뒤에 제공됩니다.

**PowerCenter** 웹 서비스용 클라이언트 응용 프로그램을 작성하려면 웹 서비스 **WSDL**과 웹 서비스 툴킷이 필요합니다. 웹 서비스 툴킷은 웹 서비스 **WSDL**에서 클라이언트 측 프록시 클래스를 생성하여 클라이언트 응용 프로그램을 쉽게 작성할 수 있게 해줍니다. **Microsoft .NET** 및 **Apache Axis** 웹 서비스 툴킷을 사용하여 **PowerCenter** 웹 서비스용 클라이언트 응용 프로그램을 기록할 수 있습니다.

클라이언트 응용 프로그램을 작성하여 **PowerCenter** 일괄 또는 실시간 웹 서비스를 실행할 수 있습니다. 응용 프로그램 개발은 동일한 기본 단계를 따릅니다.

**참고:** 웹 서비스 협은 청크된 메시지를 처리할 수 있습니다. 청크된 전송 인코딩을 클라이언트 요청에서 활성화하려면 다음 헤더를 **SOAP** 메시지에 추가합니다.

```
TRANSFER_ENCODING=chunked
```



# 일괄 웹 서비스용 클라이언트 응용 프로그램

PowerCenter 웹 서비스 공급자에서 사용 가능한 일괄 웹 서비스에 액세스하는 클라이언트 응용 프로그램 개발에는 다음 요소가 포함됩니다.

- 클라이언트 프록시 클래스
- 초기화
- 세션 유지 관리
- 작업 호출
- 리소스 정리
- 오류 처리
- 프록시 개체

## 클라이언트 프록시 클래스 생성

웹 서비스 협에서 제공되는 일괄 웹 서비스 작업을 사용하려면 웹 서비스 툴킷을 사용하여 웹 서비스 WSDL에 필요한 클라이언트 프록시 클래스를 생성해야 합니다.

클라이언트 프록시를 생성하려면 다음 단계를 수행합니다.

1. 개발하려는 언어와 플랫폼에 적합한 웹 서비스 툴킷을 선택합니다.
2. 웹 서비스 협 콘솔에서 메타데이터 웹 서비스 및 데이터 통합 웹 서비스용 WSDL 파일을 다운로드합니다. 기본적으로 웹 서비스 협 콘솔에서 WSDL 파일을 다운로드할 때 끝점 URL이 웹 서비스 협 호스트 이름과 포트 번호로 설정됩니다. 프록시 클래스를 생성하기 전에 WSDL 파일에 올바른 끝점 URL이 포함되어 있는지 확인합니다.
3. 웹 서비스 툴킷을 사용하여 WSDL 파일에서 클라이언트 측 프록시 클래스를 생성합니다. 프록시 클래스 생성에 대한 자세한 내용은 웹 서비스 툴킷 설명서를 참조하십시오. 각 툴킷은 고유한 방법으로 클라이언트 프록시 클래스를 생성합니다.

## 초기화

클라이언트 응용 프로그램은 메타데이터 및 데이터 통합 웹 서비스 작업을 호출하기 전에 초기화 단계를 수행합니다.

초기화를 수행하려면 다음 단계를 수행합니다.

1. 메타데이터 API에 대해 프록시 클래스를 인스턴스화합니다.  
예제에서 메타데이터 API 프록시 개체의 이름은 *MWSPProxy*입니다. 이 섹션에서는 *MWSPProxy*라는 이름을 사용하여 메타데이터 API 프록시 개체를 참조합니다.
2. 데이터 통합 API에 대해 프록시 클래스를 인스턴스화합니다.  
이 예제에서 데이터 통합 API 프록시 개체의 이름은 *DIWSPProxy*입니다. 이 섹션에서는 *DIWSPProxy*라는 이름을 사용하여 데이터 통합 API 프록시 개체를 참조합니다.
3. *MWSPProxy* 개체를 사용하여 로그인 작업을 호출합니다.  
로그인 작업은 리포지토리 이름, 사용자 이름 및 암호가 필요하며 세션 ID를 반환합니다. 이 작업 호출은 *MWSPProxy* 개체를 리포지토리 이름과 사용자 이름 쌍에 연결합니다. *MWSPProxy* 개체를 사용하여 일괄 웹 서비스 작업에 대해 생성된 모든 후속 요청은 이러한 리포지토리 이름과 사용자 이름을 사용합니다.

## 세션 유지 관리

웹 서비스 협은 리소스를 캐시하는 세션 유지 관리가 필요합니다. SOAP 메시지의 SOAP 헤더는 세션 유지 관리를 쉽게 수행할 수 있도록 세션 정보를 전달합니다.

세션 유지 관리를 설정하고 수행하려면 다음 단계를 수행합니다.

1. 로그인 작업 호출의 응답에서 루트 요소 이름 **Context**와 네임스페이스 **http://www.informatica.com/wsh**가 있는 헤더를 추출합니다. 이 SOAP 헤더에는 웹 서비스 협에서 보낸 세션 ID가 포함되어 있습니다.
2. 로그인 작업 호출 후에 MWSPProxy 개체에서 SOAP 헤더를 설정합니다. 이렇게 하면 MWSPProxy 개체를 사용하는 모든 후속 요청에 대해 SOAP 헤더의 세션 ID를 보냅니다.
3. DIWSPProxy 개체를 사용하여 생성된 모든 후속 요청에 대해 동일한 세션 ID를 보내도록 DIWSPProxy 개체에서 동일한 세션 ID로 SOAP 헤더를 설정합니다.

## 작업 호출

이제 MWSPProxy 및 DIWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업과 데이터 통합 웹 서비스 작업을 호출할 수 있습니다. MWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업을 호출합니다. DIWSPProxy 개체를 사용하여 데이터 통합 웹 서비스 작업을 호출합니다.

## 리소스 정리

웹 서비스 협은 세션 만료를 구현하여 성능 및 리소스 정리에 적용합니다. 로그아웃 작업은 클라이언트 응용 프로그램에서 보유하고 있는 웹 서비스 협 리소스를 해제하고 정리 작업을 수행합니다.

리소스를 해제하려면 MWSPProxy 개체를 사용하여 로그아웃 작업을 호출합니다.

리포지토리에 로그인하지만 로그아웃 작업을 호출하지 않으면 웹 서비스 협에서 세션 만료 기간이 지난 후에 리소스 정리를 수행합니다.

## 오류 처리

SOAP 응답의 SOAP fault 요소에는 웹 서비스 호출 동안 발생하는 오류가 포함됩니다.

일괄 웹 서비스 작업을 호출하는 동안 클라이언트 응용 프로그램은 적절한 오류 처리 구성표를 구현하여 SOAP 결함을 검색해야 합니다. 이 구성표는 툴킷에 따라 다릅니다.

웹 서비스 툴킷은 fault 요소의 faultcode 및 faultstring 필드를 가져올 수 있도록 예외 처리 구성표를 제공합니다. 그러나 detail 요소 필드를 구문 분석하여 오류 코드와 확장 세부 정보를 가져오는 XML 파서가 필요할 수 있습니다.

## 프록시 개체

로그인 작업 호출은 사용자가 제공하는 리포지토리 및 사용자 이름에 대한 세션을 작성합니다. 로그인 작업 호출에서 가져오는 세션 ID는 이 세션을 식별합니다. 세션 ID는 메타데이터 프록시 개체에 해당합니다. 메타데이터 프록시 개체는 세션 ID가 올바르게 올바른 개체로 간주됩니다. 로그아웃 작업을 호출하면 세션 ID는 해당 메타데이터 및 데이터 통합 프록시 개체와 함께 올바르게 않게 됩니다.

# 일괄 웹 서비스용 Java 클라이언트 응용 프로그램

이 섹션에서는 Axis 웹 서비스 툴킷을 사용하여 Java로 클라이언트 응용 프로그램을 기록하는 단계를 강조 표시합니다.

**참고:** 다음 섹션의 샘플 코드 조각은 웹 서비스 협과 함께 제공되는 일괄 웹 서비스 샘플 프로그램에서 발췌한 것입니다. 다음 디렉터리에서 PowerCenter 웹 서비스용 샘플 프로그램을 볼 수 있습니다.

```
<PowerCenterInstallationDir>/server/samples/WebServices/samples/BatchWebServices/axis
```

## Axis에서 클라이언트 프록시 클래스 생성

Axis 웹 서비스 툴킷을 사용하여 Java로 클라이언트 프록시 클래스를 생성할 수 있습니다. 툴킷을 사용하려면 axis.jar 파일을 CLASSPATH 환경 변수에 포함시켜야 합니다.

Java로 클라이언트 프록시 클래스를 생성하려면 다음 단계를 수행합니다.

1. 웹 서비스 협 콘솔에서 메타데이터 웹 서비스 및 데이터 통합 웹 서비스 WSDL 파일을 다운로드합니다.  
WSDL의 끝점 URL에 웹 서비스 협에 대한 올바른 호스트 이름과 포트 번호가 있는지 확인합니다. 끝점 URL이 올바르지 않은 경우 WSDL의 definitions\service\port 계층에 있는 address 요소를 업데이트합니다.
2. 다음 명령을 사용하여 클라이언트 프록시 클래스를 생성합니다.

```
java org.apache.axis.wsdl.WSDL2Java --NStoPkg  
http://www.informatica.com/wsh=ProxyClasses -W <WSDLFile>
```

-W 옵션은 래핑된 문서 리터럴 서비스에 대한 지원을 해제합니다.

예를 들어 이름이 Metadata.wsdl 및 DataIntegration.wsdl인 WSDL 파일의 경우 다음 명령을 실행합니다.

```
java org.apache.axis.wsdl.WSDL2Java --NStoPkg  
http://www.informatica.com/wsh=ProxyClasses -W Metadata.wsdl  
java org.apache.axis.wsdl.WSDL2Java --NStoPkg  
http://www.informatica.com/wsh=ProxyClasses -W DataIntegration.wsdl
```

이 명령은 ProxyClasses 패키지의 클라이언트 프록시 클래스를 생성합니다. 명령은 다음과 같은 프록시 클래스를 생성합니다.

- **MetadataInterface.java.** 메타데이터 웹 서비스용 인터페이스가 포함되어 있습니다.
- **DataIntegrationInterface.java.** 데이터 통합 웹 서비스용 인터페이스가 포함되어 있습니다.

## Axis에서 초기화

클라이언트 응용 프로그램은 메타데이터 웹 서비스 및 데이터 통합 웹 서비스를 호출하기 전에 초기화 단계를 수행해야 합니다.

초기화를 수행하려면 다음 단계를 수행합니다.

1. 다음과 같이 서비스 로케이터 클래스를 인스턴스화하여 MetadataService 및 DataIntegrationService 개체를 작성합니다.

```
MetadataService mdService = new MetadataServiceLocator();  
DataIntegrationService diService = new DataIntegrationServiceLocator();
```

2. 1단계에서 작성한 MetadataService 개체에서 MetadataInterface 개체(MWSPProxy)를 가져옵니다.

Metadata.wsdl의 메타데이터 서비스 끝점 URL에 올바른 URL이 있는 경우 MWSPProxy 개체를 가져옵니다.

```
MWSPProxy=mdService.getMetadata();
```

그렇지 않으면 MWSPProxy 개체를 가져옵니다.

```
MWSPProxy=mdService.getMetadata(new java.net.URL(MWS_URL));
```

MWS\_URL은 메타데이터 웹 서비스에 대한 끝점 URL을 포함하는 변수입니다.

MWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업을 호출합니다.

3. 1단계에서 작성한 `DataIntegrationService` 개체에서 `DataIntegrationInterface` 개체(`DIWSPProxy`)를 가져옵니다.

`DataIntegration.wsdl`의 서비스 끝점 URL에 올바른 URL이 있는 경우 `DIWSPProxy` 개체를 가져옵니다.

```
DIWSPProxy=diService.getDataIntegration();
```

그렇지 않으면 `DIWSPProxy` 개체를 가져옵니다.

```
DIWSPProxy=diService.getDataIntegration(new java.net.URL(DIWS_URL));
```

`DIWS_URL`은 데이터 통합 웹 서비스에 대한 끝점 URL을 포함하는 변수입니다.

`DIWSPProxy` 개체를 사용하여 데이터 통합 웹 서비스 작업을 호출합니다.

4. `MWSPProxy` 개체로 로그인 작업을 호출하여 클라이언트 응용 프로그램 사용자 계정에 대한 세션 ID를 작성합니다. 로그인 작업은 `LoginRequest` 개체에 래핑되어 있는 도메인, 리포지토리, 사용자 이름 및 암호를 가져오고 세션 ID를 반환합니다.

```
LoginRequest loginReq = new LoginRequest();
loginReq.setRepositoryDomainName(REPO_DOMAIN_NAME);
loginReq.setRepositoryName(REPO_NAME);
loginReq.setUserName(USER_NAME);
loginReq.setPassword(PASSWORD);
String sessionID = MWSPProxy.login(loginReq);
```

`REPO_DOMAIN_NAME`은 `PowerCenter` 도메인 이름을 포함하는 문자열이고 `REPO_NAME`은 도메인의 리포지토리 이름을 포함하는 문자열이며, `USER_NAME`은 리포지토리에 올바른 사용자 이름을 포함하는 문자열이고 `PASSWORD`는 리포지토리에 로그인할 사용자의 암호를 포함하는 문자열입니다.

5. `MWSPProxy` 및 `DIWSPProxy` 개체를 세션 ID의 리포지토리 및 사용자 이름에 연결합니다. `MWSPProxy` 또는 `DIWSPProxy` 개체를 사용하여 일괄 웹 서비스에 대해 생성된 모든 후속 요청은 세션 ID의 리포지토리 및 사용자 이름을 사용합니다.

```
((org.apache.axis.client.Stub)MWSPProxy).setHeader(createSessionHeader(sessionID));
((org.apache.axis.client.Stub)DIWSPProxy).setHeader(createSessionHeader(sessionID));
```

## Axis에서 세션 유지 관리

웹 서비스 협은 리소스를 캐시하는 세션 유지 관리가 필요합니다. SOAP 메시지의 SOAP 헤더는 세션 유지 관리를 쉽게 수행할 수 있도록 세션 정보를 전달합니다.

세션 유지 관리를 수행하려면 다음 단계를 수행합니다.

1. `MWSPProxy` 개체를 사용하는 로그인 작업 호출의 응답에서 루트 요소 이름 `Context`와 네임스페이스 `http://www.informatica.com/wsh`가 있는 SOAP 헤더를 추출합니다. 이 SOAP 헤더에는 웹 서비스 협에서 보낸 세션 ID가 포함되어 있습니다.

```
/** Create session ID in the Soap message header */
public static SOAPHeaderElement createSessionHeader(String sessID) throws SOAPException
{
    String WSSE_NS = "http://www.informatica.com/wsh";
    String WSSE_PREFIX = "infa";
    Name hdrname = SOAPFactory.newInstance().createName("Context", WSSE_PREFIX, WSSE_NS);
    SOAPHeaderElement header = new SOAPHeaderElement(hdrname);
    SOAPElement token = header.addChildElement("SessionId", "", "");
    token.addTextNode(sessID);
    return header;
}
```

2. `MWSPProxy` 개체를 사용하는 모든 후속 요청에 대해 SOAP 헤더의 이 세션 ID를 보냅니다. 로그인 작업 호출 후에 `MWSPProxy` 개체에 SOAP 헤더를 한 번 설정합니다.

```
((org.apache.axis.client.Stub) MWSPProxy).setHeader(createSessionHeader(sessionID));
```

3. 동일한 SOAP 헤더가 있는 `DIWSPProxy` 개체에 SOAP 헤더를 설정합니다.

```
((org.apache.axis.client.Stub) DIWSPProxy).setHeader(createSessionHeader(sessionID));
```

## Axis에서 작업 호출 생성

이제 MWSPProxy 및 DIWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업과 데이터 통합 웹 서비스 작업을 호출할 수 있습니다.

예를 들어 getAllDIServers 작업을 호출하여 통합 서비스 목록을 가져올 수 있습니다.

```
DIServerInfoArray servers = MWSPProxy.getAllDIServers(null);
if (servers.getDIServerInfo() != null) {
    for(int i=0; i < servers.getDIServerInfo().length ; i++) {
        System.out.println("("+(i+1)+") "+servers.getDIServerInfo(i).getName());
    }
}
```

pingDIServer 작업을 호출하여 통합 서비스의 상태를 확인할 수 있습니다.

```
DIServiceInfo diInfo = new DIServiceInfo();
diInfo.setDomainName(DI_DOMAIN_NAME);
diInfo.setServiceName(SERVICE_NAME);
PingDIServerRequest pingReq = new PingDIServerRequest();
pingReq.setDIServiceInfo(diInfo);
pingReq.setTimeout(100);
EPingState eps = DIWSPProxy.pingDIServer(pingReq);
```

DI\_DOMAIN\_NAME은 통합 서비스가 포함되어 있는 도메인의 이름을 포함하는 변수입니다. SERVICE\_NAME은 통합 서비스 이름을 포함하는 변수입니다.

## Axis에서 정리

정리 작업은 클라이언트 응용 프로그램에서 보유하고 있는 웹 서비스 힙 리소스를 해제합니다. 리소스를 정리하고 해제하려면 MWSPProxy 개체를 사용하여 로그아웃 작업을 호출합니다.

```
MWSPProxy.logout(null);
```

## Axis에서 오류 처리

try 블록에 코드를 배치하고 FaultDetails 개체를 캐치하여 Axis에서 클라이언트 응용 프로그램 오류 처리를 구현할 수 있습니다. FaultDetails 클래스는 클라이언트 프록시의 일부로 생성됩니다.

try 블록에서 다음 코드를 사용하여 FaultDetails 개체를 캐치할 수 있습니다.

```
try {
    // Code for steps explained above.
} catch(FaultDetails fault) {
    // Display fault code
    System.out.println("fault code : " + fault.getFaultCode());
    // Display fault string
    System.out.println("fault string : " + fault.getFaultString());
    // Display error code
    System.out.println("error code is : " + fault.getErrorCode());
    // Display extended details
    System.out.println("extended detail is : " + fault.getExtendedDetails());
}
```

# 일괄 웹 서비스용 C# 클라이언트 응용 프로그램

이 섹션에서는 .NET 웹 서비스 툴킷을 사용하여 C#으로 클라이언트 응용 프로그램을 기록하는 단계를 강조 표시합니다.

**참고:** 다음 섹션의 샘플 코드 조각은 일괄 웹 서비스 샘플 프로그램에서 발췌한 것입니다. 다음 디렉터리에서 샘플 프로그램을 볼 수 있습니다.

```
<PowerCenterInstallationDir>\server\samples\WebServices\samples\BatchWebServices\dotnet\csharp
```

## .NET에서 클라이언트 프록시 클래스 생성

Microsoft .NET 웹 서비스 툴킷을 사용하여 C#으로 웹 서비스 헵에 대한 클라이언트 프록시 클래스를 작성할 수 있습니다.

C#으로 클라이언트 프록시를 생성하려면 다음 단계를 수행합니다.

1. 웹 서비스 헵 콘솔에서 메타데이터 웹 서비스 및 데이터 통합 웹 서비스 WSDL 파일을 다운로드합니다. WSDL의 끝점 URL에 웹 서비스 헵에 대한 올바른 호스트 이름과 포트 번호가 있는지 확인합니다. 끝점 URL이 올바르지 않은 경우 WSDL의 definitions\service\port 계층에 있는 address 요소를 업데이트합니다.
2. 다음 명령을 사용하여 클라이언트 프록시 클래스를 생성합니다.

```
wsdl <WSDLFile>
```

예를 들어 이름이 Metadata.wsdl 및 DataIntegration.wsdl인 WSDL 파일의 경우 다음 명령을 실행합니다.

```
wsdl Metadata.wsdl  
wsdl DataIntegration.wsdl
```

명령은 다음과 같은 프록시 클래스를 생성합니다.

- **MetadataService.cs.** 메타데이터 웹 서비스용 인터페이스가 포함되어 있습니다.
- **DataIntegrationService.cs.** 데이터 통합 웹 서비스용 인터페이스가 포함되어 있습니다.

## .NET에서 초기화

클라이언트 응용 프로그램은 메타데이터 웹 서비스 및 데이터 통합 웹 서비스를 호출하기 전에 초기화 단계를 수행해야 합니다.

초기화를 수행하려면 다음 단계를 수행합니다.

1. MetadaService 클래스 개체(MWSPProxy)를 인스턴스화합니다.

```
MWSPProxy= new MetadaService();
```

Metadata.wsdl의 메타데이터 서비스 끝점 URL에 올바른 URL이 없는 경우 다음 코드를 사용하여 URL을 설정할 수 있습니다.

```
MWSPProxy.Url = MWS_URL;
```

MWS\_URL은 메타데이터 웹 서비스에 대한 끝점 URL을 포함하는 변수입니다.

MWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업을 호출합니다.

2. DataIntegrationService 클래스 개체(DIWSProxy)를 인스턴스화합니다.

```
DIWSProxy= new DataIntegrationService ();
```

DataIntegration.wsdl의 데이터 통합 서비스 끝점 URL에 올바른 URL이 없는 경우 다음 코드를 사용하여 URL을 설정할 수 있습니다.

```
DIWSProxy.Url = DIWS_URL;
```

DIWS\_URL은 데이터 통합 웹 서비스 끝점 URL을 포함하는 문자열입니다.

DIWSProxy 개체를 사용하여 데이터 통합 웹 서비스 작업을 호출합니다.

3. MWSPProxy 개체로 로그인 작업을 호출하여 클라이언트 응용 프로그램 사용자 계정에 대한 세션 ID를 작성합니다. 로그인 작업은 LoginRequest 개체에 래핑되어 있는 도메인, 리포지토리, 사용자 이름 및 암호를 가져오고 세션 ID를 반환합니다.

```
LoginRequest loginReq = new LoginRequest();
loginReq.RepositoryDomainName = REPO_DOMAIN_NAME;
loginReq.RepositoryName = REPO_NAME;
loginReq.UserName = USER_NAME;
loginReq.Password = PASSWORD;
String sessID = MWSPProxy.Login(loginReq);
```

REPO\_DOMAIN\_NAME은 PowerCenter 도메인 이름을 포함하는 문자열이고 REPO\_NAME은 도메인의 리포지토리 이름을 포함하는 문자열이며, USER\_NAME은 리포지토리에 올바른 사용자 이름을 포함하는 문자열이고 PASSWORD는 리포지토리에 로그인할 사용자의 암호를 포함하는 문자열입니다.

4. MWSPProxy 및 DIWSPProxy 개체를 세션 ID의 리포지토리 및 사용자 이름에 연결합니다. MWSPProxy 또는 DIWSPProxy 개체를 사용하여 일괄 웹 서비스에 대해 생성된 모든 후속 요청은 세션 ID의 리포지토리 및 사용자 이름을 사용합니다.

```
MWSPProxy.Context.SessionId = sessID;
DIWSPProxy.Context.SessionId = sessID;
```

## .NET에서 세션 유지 관리

웹 서비스 협은 리소스를 캐시하는 세션 유지 관리가 필요합니다. SOAP 메시지의 SOAP 헤더는 세션 유지 관리를 쉽게 수행할 수 있도록 세션 정보를 전달합니다.

추가적인 단계는 수행하지 않아도 됩니다. .NET 클라이언트 프록시 클래스는 세션 유지 관리를 처리합니다.

## .NET에서 작업 호출 생성

이제 MWSPProxy 및 DIWSPProxy 개체를 사용하여 메타데이터 웹 서비스 작업과 데이터 통합 웹 서비스 작업을 호출할 수 있습니다.

예를 들어 getAllDIServers 작업을 호출하여 통합 서비스 목록을 가져올 수 있습니다.

```
DIServerInfo[] servers = MWSPProxy.GetAllDIServers(null);
if (servers != null) {
    for(int i=0; i < servers.Length ; i++) {
        Console.WriteLine("("+(i+1)+") "+servers[i].Name);
    }
}
```

pingDIServer 작업을 호출하여 통합 서비스의 상태를 확인할 수 있습니다.

```
PingDIServerRequest pingReq = new PingDIServerRequest();
pingReq.Timeout = (PING_TIME_OUT);
DIServiceInfo diInfo1 = new DIServiceInfo();
diInfo1.DomainName = DI_DOMAIN_NAME;
diInfo1.ServiceName = DI_SERVICE_NAME1;
pingReq.DIServiceInfo = diInfo1;
EPingState pingResult = DIWSPProxy1.pingDIServer(pingReq);
```

DI\_DOMAIN\_NAME은 통합 서비스가 포함되어 있는 도메인의 이름을 포함하는 변수입니다.

DI\_SERVICE\_NAME은 통합 서비스 이름을 포함하는 변수입니다.

## .NET에서 오류 처리

try 블록에 코드를 배치하고 SOAP 예외 개체를 캐치하여 .NET에서 클라이언트 응용 프로그램 오류 처리를 구현할 수 있습니다. SOAP 예외 클래스는 .NET 프레임워크 SDK의 일부입니다.



try 블록에서 다음 코드를 사용하여 SOAP 예외 개체를 캐치할 수 있습니다.

```
try {
    //Code for steps explained above.
}
catch(SoapException fault) {
    // Display fault code
    Console.WriteLine("fault code is : " + fault.Code);
    // Display fault string
    Console.WriteLine("fault string is : " + fault.Message);
    // Parsing detail element
    XmlNode detail = fault.Detail;
    XmlElement WSHFaultDetails = detail["WSHFaultDetails", "http://www.informatica.com/PowerCenter"];
    XmlElement ErrorCode= WSHFaultDetails ["ErrorCode"];
    XmlElement ExtendedDetails= WSHFaultDetails ["ExtendedDetails"];
    // Display error code
    Console.WriteLine ("error code is : " + ErrorCode.InnerText);
    // Display extended details
    Console.WriteLine ("extended detail is : " + ExtendedDetails.InnerText);
}
```

## 실시간 웹 서비스용 클라이언트 응용 프로그램

실시간 웹 서비스용 클라이언트 응용 프로그램에는 다음 요소가 포함되어 있습니다.

- 웹 서비스 워크플로우
- 클라이언트 프록시 클래스
- 초기화
- 작업 호출
- 오류 처리

### 웹 서비스 워크플로우

실시간 웹 서비스 클라이언트 응용 프로그램을 작성하여 웹 서비스 워크플로우를 실행합니다. 클라이언트 응용 프로그램을 작성하기 전에 **PowerCenter**에서 매핑과 워크플로우를 작성합니다. 워크플로우의 다음 옵션을 활성화하여 클라이언트 응용 프로그램이 워크플로우를 실행할 수 있게 합니다.

- **웹 서비스.** 워크플로우를 웹 서비스 워크플로우로 변환하는 웹 서비스 옵션을 활성화합니다.
- **실행 가능.** 클라이언트 응용 프로그램이 웹 서비스 워크플로우를 실행할 수 있게 하는 실행 가능 옵션을 활성화합니다.
- **표시.** 웹 서비스 힙 콘솔에서 웹 서비스 힙이 웹 서비스용 **WSDL**을 게시할 수 있도록 표시 옵션을 활성화합니다.

### 클라이언트 프록시 클래스 생성

**PowerCenter**에서 작성하는 실시간 웹 서비스를 사용하려면 액세스하려는 웹 서비스의 **WSDL**에서 클라이언트 프록시 클래스를 생성해야 합니다.

클라이언트 프록시를 생성하려면 다음 단계를 수행합니다.

1. 개발하려는 언어와 플랫폼에 적합한 웹 서비스 툴킷을 선택합니다.
2. 웹 서비스 힙 콘솔에서 실시간 웹 서비스용 **WSDL**을 다운로드합니다.



3. 웹 서비스 툴킷을 사용하여 WSDL에서 클라이언트 측 프록시 클래스를 생성합니다. 프록시 클래스 생성에 대한 자세한 내용은 웹 서비스 툴킷 설명서를 참조하십시오. 각 툴킷은 고유한 방법으로 클라이언트 프록시 클래스를 생성합니다.

## 초기화

클라이언트 응용 프로그램이 웹 서비스 작업을 호출하려면 클라이언트 프록시 클래스의 웹 서비스 개체를 인스턴스화하고 웹 서비스용 포트를 가져와야 합니다.

## 작업 호출

웹 서비스 작업을 호출하려면 클라이언트 응용 프로그램이 요청 개체를 작성하여 포트 작업에 전달해야 합니다. 웹 서비스가 응답을 보내오면 클라이언트 응용 프로그램이 필요에 따라 응답을 처리해야 합니다.

## 오류 처리

실시간 웹 서비스 클라이언트 응용 프로그램에서 오류를 처리하는 방식은 일괄 웹 서비스 클라이언트 응용 프로그램과 동일합니다. SOAP 응답의 SOAP fault 요소에는 웹 서비스 호출 동안 발생하는 오류가 포함됩니다. 클라이언트 응용 프로그램은 적절한 오류 처리 구성표를 구현하여 SOAP 결함을 검색해야 합니다.

# 실시간 웹 서비스용 Java 클라이언트 응용 프로그램

이 섹션에서는 Axis 웹 서비스 툴킷을 사용하여 PowerCenter 실시간 웹 서비스를 호출하는 Java 클라이언트 응용 프로그램 작성 방법에 대한 지침을 제공합니다. Axis 웹 서비스 툴킷 사용 방법에 대한 자세한 내용은 Apache 웹 사이트에서 해당 설명서를 참조하십시오.

<http://ws.apache.org/axis/java/user-guide.html>

PowerCenter 웹 서비스 워크플로우를 호출하는 클라이언트 응용 프로그램을 작성하기 전에 웹 서비스 워크플로우를 작성하고 웹 서비스용 WSDL을 생성해야 합니다. 그런 다음 웹 서비스 WSDL을 기반으로 클라이언트 응용 프로그램을 작성합니다.

PowerCenter 웹 서비스를 작성하고 WSDL을 생성하려면 다음 단계를 수행합니다.

1. 웹 서비스 워크플로우에 대한 매핑을 작성합니다. 매핑을 작성하면 웹 서비스 클라이언트에서 메시지를 받고, 데이터를 변환하고, 응답을 웹 서비스 클라이언트로 다시 보내거나 PowerCenter에서 지원하는 모든 대상에 기록할 수 있습니다.
2. 워크플로우를 작성하고 웹 서비스로 활성화합니다. 매핑을 실행할 워크플로우를 작성하고 워크플로우 속성에서 웹 서비스 옵션을 활성화합니다. PowerCenter 외부에 있는 클라이언트 응용 프로그램이 워크플로우를 실행할 수 있도록 실행 가능 옵션을 선택합니다.
3. 웹 서비스 워크플로우에 대한 WSDL을 찾아서 다운로드합니다. 웹 서비스 워크플로우를 작성하면 PowerCenter에서 웹 서비스용 WSDL을 생성합니다. 웹 서비스를 표시 가능하도록 구성하는 경우 웹 서비스와 연결된 웹 서비스 힙 콘솔에서 WSDL을 볼 수 있습니다.

웹 서비스를 작성한 후 웹 서비스 워크플로우를 실행하는 클라이언트 응용 프로그램을 개발할 수 있습니다.

## 관련 항목:

- [“웹 서비스 워크플로우 작성 및 구성” 페이지 96](#)
- [“웹 서비스 섹션 ” 페이지 24](#)

## 실시간 웹 서비스용 클라이언트 응용 프로그램 작성

실시간 웹 서비스를 호출하는 클라이언트 응용 프로그램을 작성하려면 다음 단계를 수행합니다.

1. 웹 서비스에 대한 클라이언트 프록시 클래스를 생성합니다.  
프록시 클래스를 작성한 후, 웹 서비스를 호출하는 **Java** 응용 프로그램을 작성합니다. **Java** 응용 프로그램 내에서 다음 단계를 수행합니다.
2. 웹 서비스 개체를 초기화합니다.
3. 요청 개체를 작성합니다.
4. 요청 개체를 포트 작업에 전달하고 응답을 처리합니다.

**참고:** 다음 섹션의 샘플 코드 조각은 다중 행 조회를 위한 실시간 웹 서비스 샘플 프로그램에서 발췌한 것입니다. 다음 디렉터리에서 예제를 볼 수 있습니다.

```
<PowerCenterInstallationDir>/server/samples/WebServices/samples/RealTimeWebServices/  
UnprotectedWebServices/axis/CustomLookup_MULTIPLEROW
```

### 1단계. Axis에서 클라이언트 프록시 클래스 생성

Axis 웹 서비스 툴킷을 사용하여 웹 서비스 WSDL에 대한 **Java** 클라이언트 프록시 클래스를 생성할 수 있습니다. 구체적으로 WSDL2Java 도구를 실행하여 **Java** 프록시 클래스 파일을 생성할 수 있습니다.

WSDL의 끝점 URL에 웹 서비스에 대한 올바른 호스트 이름과 포트 번호가 있는지 확인합니다. 끝점 URL이 올바르지 않은 경우 WSDL의 \definitions\service\port 계층에 있는 address 요소를 업데이트합니다.

다음 명령을 사용하여 클라이언트 프록시 클래스를 생성합니다.

```
java org.apache.axis.wsdl.WSDL2Java -W <WSDLFile>
```

예를 들어 이름이 SampleWS.wsdl인 WSDL의 경우 다음 명령을 실행합니다.

```
java org.apache.axis.wsdl.WSDL2Java -W SampleWS.wsdl
```

-W 옵션은 래핑된 문서 리터럴 서비스에 대한 지원을 해제합니다.

WSDL2Java는 WSDL에 정의된 각 데이터 유형에 대해 클래스를 생성합니다. 기본적으로 WSDL2Java는 WSDL의 네임스페이스를 기반으로 패키지 이름을 생성합니다. 일반적으로 네임스페이스가 *http://x.y.com* 또는 *urn:x.y.com* 형식인 경우 해당 패키지는 *com.y.x*가 됩니다.

### 2단계. 웹 서비스 개체 초기화

웹 서비스 작업을 호출하려면 클라이언트 프록시 클래스에 웹 서비스 개체를 작성하고 웹 서비스용 포트를 가져와야 합니다.

웹 서비스 개체를 작성하려면 서비스 로케이터 클래스를 인스턴스화합니다. 샘플 프로그램에서 다음 코드는 서비스 로케이터를 인스턴스화합니다.

```
CustomerLookup_MULTIPLEROW service = new CustomerLookup_MULTIPLEROWLocator();
```

웹 서비스용 포트를 가져오려면 포트 유형에 대해 작성된 프록시 클래스를 사용합니다. 샘플 프로그램에서 다음 코드는 웹 서비스용 포트를 가져옵니다.

```
CustomerLookup_MULTIPLEROWPort port =  
    service.getCustomerLookup_MULTIPLEROWPort(new java.net.URL(END_POINT_URL));
```

END\_POINT\_URL 변수에는 WSDL의 URL이 포함되어 있습니다.

### 3단계. 요청 개체 작성

요청 개체를 작성하고 웹 서비스에 전달할 모든 필수 매개 변수를 작성해야 합니다. 샘플 클라이언트 응용 프로그램에서 다음 코드는 조회 요청 개체를 작성합니다.

```
CustomerLookupRequest request = new CustomerLookupRequest();
request.setCustomerID_in(CustomerID);
```

### 4단계. 요청 보내기 및 응답 처리

요청 개체를 작성한 후 해당 개체를 포트 작업에 전달합니다. 웹 서비스에서 응답을 다시 보냅니다. 요구 사항에 맞게 응답을 처리할 수 있습니다.

샘플 클라이언트 응용 프로그램에서 다음 코드는 요청 개체를 포트에 전달하고 응답을 표시합니다.

```
CustomerLookupResponse[] response = port.customerLookup_MULTIPLEROWOperation(requestOperation);
System.out.println();
if (response[0].getCustomerID_out() == 0)
{
    System.out.println("Customer(s) with the ID as " + CustomerID + " does not exist!!!");
}
else
{
    System.out.println("***** Customer(s) that matches with the Customer ID is/are ...");
    for (int i = 0; i < response.length; i++)
    {
        System.out.println("***** Customer ID: " + response[i].getCustomerID_out());
        System.out.println("***** Customer Name: " + response[i].getCustomerName_out());
        System.out.println("***** Customer Age: " + response[i].getCustomerAge_out());
        System.out.println("***** Customer Gender: " +
            response[i].getCustomerGender_out());
        System.out.println("***** Customer Address: " +
            response[i].getCustomerAddress_out());
        if (i < response.length - 1) System.out.println ();
    }
}
```

## 매개 변수 배열 사용

PowerCenter에서 매개 변수는 세션 사이에 변경할 수 있는 값(예: 데이터베이스 연결이나 소스 또는 대상 파일)을 나타냅니다. 워크플로우 또는 세션과 연결된 매개 변수를 작성하여 워크플로우 또는 세션을 실행할 때마다 유연성을 제공할 수 있습니다.

웹 서비스 클라이언트 응용 프로그램의 경우 매개 변수 파일 또는 매개 변수 배열에서 워크플로우 또는 세션과 연결된 매개 변수에 값을 정의할 수 있습니다. 매개 변수 파일의 매개 변수를 사용하려면 클라이언트 응용 프로그램에서 매개 변수 파일 이름을 지정합니다. 매개 변수 파일은 통합 서비스에서 액세스할 수 있어야 합니다. 매개 변수 배열을 사용하려면 클라이언트 응용 프로그램에서 매개 변수 배열의 요소에 매개 변수 값을 제공합니다.

예를 들어 워크플로우 또는 태스크를 시작하는 요청은 워크플로우 또는 태스크와 연결된 매개 변수를 매개 변수 파일 이름 또는 매개 변수 배열의 매개 변수 및 값 목록을 통해 지정할 수 있습니다.

### 매개 변수 배열 정의

SOAP 요청의 매개 변수 정의는 매개 변수의 범위, 이름 및 값으로 구성되어 있습니다. 통합 서비스는 워크플로우 또는 태스크를 실행할 때 매개 변수 파일의 매개 변수를 사용하는 방식과 동일하게 배열의 매개 변수를 사용합니다.

WSDL에는 매개 변수 배열 요소에 대한 다음 정의가 포함되어 있습니다.

```
<complexType name="Parameter">
  <sequence>
    <element name="Scope" type="xsd:string" />
    <element name="Name" type="xsd:string" />
    <element name="Value" type="xsd:string" />
  </sequence>
</complexType>

<complexType name="ParameterArray">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0" name="Parameters"
      nillable="true" type="impl:Parameter" />
  </sequence>
</complexType>
```

예를 들어 매개 변수 파일에 다음 매개 변수가 있습니다.

```
[s_m_A]
$a=1
$b=2
$c=3
[WSH_Folder.s_m_B]
$d=4
```

매개 변수 배열의 동일한 매개 변수를 사용하여 웹 서비스에서 **StartWorkflow** 작업으로의 호출에 대한 SOAP 요청에는 다음 요소가 포함될 수 있습니다.

```
<StartWorkflow>
...
  <Parameters>
    <Parameter>
      <Scope>s_m_A</Scope>
      <Name>$a</Name>
      <Value>1</Value>
    </Parameter>
    <Parameter>
      <Scope>s_m_A</Scope>
      <Name>$b</Name>
      <Value>2</Value>
    </Parameter>
    <Parameter>
      <Scope>s_m_A</Scope>
      <Name>$c</Name>
      <Value>3</Value>
    </Parameter>
    <Parameter>
      <Scope>WSH_Folder.s_m_B</Scope>
      <Name>$d</Name>
      <Value>4</Value>
    </Parameter>
  </Parameters>
</StartWorkflow>
```

**WorkflowRequest** 및 **TaskRequest** 유형에는 **ParameterArray** 요소가 포함됩니다. 매개 변수 배열에는 매개 변수 수를 자유롭게 지정할 수 있습니다.

Axis의 웹 서비스 클라이언트 응용 프로그램에서 발췌한 다음 샘플 코드는 **WorkflowRequest**에서 매개 변수 배열을 작성하는 방법을 보여줍니다.

```
Parameter[] parameters = new Parameter[4];

Parameter param1 = new Parameter();
Param1.setScope("s_m_A");
Param1.setName("$a");
Param1.setValue("1");
Parameters[0] = param1;

Parameter param2 = new Parameter();
Param2.setScope("s_m_A");
```

```

Param2.setName("$b");
Param2.setValue("2");
Parameters[1] = param2;

Parameter param3 = new Parameter();
Param3.setScope("s_m_A");
Param3.setName("$c");
Param3.setValue("3");
Parameters[2] = param3;

Parameter param4 = new Parameter();
Param4.setScope("WSH_Folder.s_m_B");
Param4.setName("$d");
Param4.setValue("4");
Parameters[3] = param4;

WorkflowRequest wfReq = new WorkflowRequest();
wfReq.setParameters(parameters);

```

다음 작업에서 매개 변수 배열을 사용할 수 있습니다.

- startWorkflow
- startWorkflowFromTask
- recoverWorkflow
- startTask

## 매개 변수 배열 사용에 대한 규칙 및 지침

웹 서비스 요청에서 매개 변수 배열을 사용할 때에는 다음 규칙과 지침을 사용합니다.

- **매개 변수 파일 또는 매개 변수 배열을 사용합니다.** 웹 서비스 작업 호출을 생성할 때 매개 변수 파일 이름과 매개 변수 배열을 SOAP 요청에 지정하지 않습니다. 매개 변수 파일과 매개 변수 배열을 SOAP 요청에 둘 다 지정하는 경우 웹 서비스 험은 해당 요청이 매개 변수 목록과 매개 변수 파일을 지정한다고 알려주는 결함 메시지를 경고로 반환합니다.
- **매개 변수 파일과 매개 변수 배열이 정의된 경우 통합 서비스는 SOAP 요청의 매개 변수 배열에 있는 매개 변수의 값을 사용합니다.** 통합 서비스는 다음 조건에 해당되는 경우 매개 변수 배열에 정의된 매개 변수의 값을 사용합니다.
  - 매개 변수 배열을 웹 서비스 요청에 지정하여 워크플로우를 시작합니다.
  - 워크플로우는 워크플로우 속성에 정의되어 있는 연결된 매개 변수 파일을 가지고 있습니다.

## 클라이언트 요청에 보안 추가

웹 서비스 험은 웹 서비스에 대해 다음 유형의 보안을 사용합니다.

- **사용자 자격 증명.** 요청에 대한 자격 증명을 가져오려면 실행할 웹 서비스를 포함하는 PowerCenter 리포지토리에 클라이언트가 로그인해야 합니다. 로그인하면 웹 서비스 클라이언트가 SOAP 요청에 포함시켜야 하는 세션 ID가 생성됩니다.
 

웹 서비스 험은 일괄 웹 서비스에 대해 이 보안 옵션을 사용합니다. 일괄 웹 서비스를 호출하는 클라이언트 응용 프로그램은 기타 작업을 호출하기 전에 리포지토리에 로그인해야 합니다.
- **사용자 이름 토큰.** 사용자 이름 토큰을 사용하고 OASIS 웹 서비스 보안 표준을 기반으로 하는 웹 서비스 보안입니다. 이 보안에는 SOAP 메시지에 대해 콘텐츠 무결성과 보안을 보장하는 SOAP 확장 집합이 포함됩니다.

사용자 이름 토큰은 보안 웹 서비스에 대한 기본 보안 옵션입니다. 기본적으로 웹 서비스 협에서 보안 웹 서비스에 대해 생성한 WSDL에는 UsernameToken 요소가 있는 보안 헤더가 포함됩니다.

OASIS 웹 서비스 보안 표준에 대한 자세한 내용은 OASIS 웹 사이트에서 웹 서비스 보안 사양을 읽어 보십시오.

[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ws-sx](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-sx)

**참고:** 클라이언트 응용 프로그램이 웹 서비스를 실행하기 위한 로그인 요청을 보내는 경우 응답을 받은 후에는 명시적 로그아웃 요청을 보냅니다. 해당하는 로그아웃 요청이 없는 로그인 요청은 리포지토리 서비스 및 웹 서비스 협 프로세스에서 메모리 누수를 일으킬 수 있습니다.

## SOAP 요청의 UsernameToken

웹 서비스 협에서 생성한 WSDL을 기반으로 클라이언트 응용 프로그램을 작성하는 경우 기본적으로 요청 개체의 헤더에 UsernameToken 요소가 포함됩니다.

SOAP 요청의 UsernameToken 요소는 다음과 같은 암호 보안 중 하나를 가지고 있을 수 있습니다.

- **일반 텍스트 암호.** 일반 텍스트로 되어 있는 암호를 포함합니다.
- **해시된 암호.** MD5 또는 SHA-1 해시 함수를 사용하여 해시된 암호화된 암호를 포함합니다.
- **요약 암호.** 임시 값과 타임스탬프로 해시된 암호화된 암호를 포함합니다.

UsernameToken의 Password 요소에 사용자 암호를 포함시킵니다.

암호 요소에는 사용된 암호 보안 유형을 표시하는 유형 특성이 있습니다. Type 특성이 생략되면 암호 유형의 기본값은 *PasswordText*입니다.

**참고:** Informatica 도메인이 Kerberos 네트워크 인증을 사용하는 경우 해시된 암호 또는 요약 암호를 SOAP 요청에 사용할 수 없습니다.

## 일반 텍스트 암호

UsernameToken 요소에는 다음 하위 요소가 포함됩니다.

- **Username 요소.** PowerCenter 원시 보안 도메인 또는 모든 LDAP 보안 도메인에서 찾을 수 있는 사용자 이름을 포함합니다. 기본 보안 도메인은 원시 보안 도메인입니다. 사용자 이름이 원시 보안 도메인에 속하는 경우 Username 요소에는 보안 도메인의 이름이 필요하지 않습니다. 사용자 이름이 LDAP 보안 도메인에 속하는 경우 사용자 이름은 보안 도메인의 이름과 슬래시(/) 뒤에 와야 합니다.

Username 요소 예제를 보여주는 다음 표에는 사용자 계정의 보안 도메인을 표시하는 데 사용하는 형식이 포함되어 있습니다.

Username 요소의 값	보안 도메인
<UsernameToken> <사용자 이름>Native/Administrator</Username> <암호>Administrator</Password> </UsernameToken>	원시
<UsernameToken> <사용자 이름>/Administrator</Username> <암호>Administrator</Password> </UsernameToken>	원시

Username 요소의 값	보안 도메인
<pre>&lt;UsernameToken&gt;   &lt;사용자 이름&gt;Administrator&lt;/Username&gt;   &lt;암호&gt;Administrator&lt;/Password&gt; &lt;/UsernameToken&gt;</pre>	원시
<pre>&lt;UsernameToken&gt;   &lt;사용자 이름&gt;LDAPAdm/Administrator&lt;/Username&gt;   &lt;암호&gt;Administrator&lt;/Password&gt; &lt;/UsernameToken&gt;</pre>	이름이 <i>LDAPAdm</i> 인 LDAP 보안 도메인입니다.

- **Password 요소.** 일반 텍스트로 되어 있는 암호를 포함합니다. Password 요소의 Type 특성은 생략하거나 *PasswordText*로 설정할 수 있습니다.

## 해시된 암호

UsernameToken 요소에는 다음 하위 요소가 포함됩니다.

- **Username 요소.** PowerCenter 원시 보안 도메인에서 찾을 수 있는 사용자 이름을 포함합니다.
- **Password 요소.** 해시된 암호를 포함합니다. 암호를 MD5 또는 SHA-1 해시 함수로 해시하고 Base64로 인코딩해야 합니다. Password 요소의 Type 특성은 생략하거나 "PasswordText"로 설정할 수 있습니다.

다음 코드는 해시된 암호를 사용하는 요청에 대한 보안 헤더의 예제를 보여줍니다.

```
<soapenv:Header>
  <!-- UsernameTokens -->
  <inf:Security>
    <UsernameToken>
      <Username>Native/Administrator</Username>
      <Password>Ntm58Cxf7SB0QAz30lsTq1nv-D7</Password>
    </UsernameToken>
  </inf:Security>
</soapenv:Header>
```

## 타사 도구를 사용하여 해시된 암호 작성

타사 도구(예: OpenSSL 및 Java MessageDigest 클래스)를 사용하여 해시된 암호를 작성할 수 있습니다.

### UNIX의 OpenSSL

UNIX 시스템에서 OpenSSL을 사용하여 해시된 암호를 작성하려면 메시지 요약 명령 *dgst*와 함께 OpenSSL을 실행합니다.

다음 예제는 MD5 해시 함수를 사용하고 Base64로 인코딩된 암호 문자열 Administrator에 대해 해시된 암호를 작성하는 방법을 보여줍니다.

```
echo -n "Administrator" | openssl dgst -md5 -binary | openssl base64
```

다음 예제는 SHA-1 해시 함수를 사용하고 Base64로 인코딩된 암호 문자열 Administrator에 대해 해시된 암호를 작성하는 방법을 보여줍니다.

```
echo -n "Administrator" | openssl dgst -sha1 -binary | openssl base64
```

**참고:** 이러한 예제에서 *echo* 명령은 줄 바꿈 문자를 문자열에 추가합니다. 명령에서 *-n* 옵션은 줄 바꿈 문자를 제거합니다.

## Windows의 OpenSSL

Windows 시스템에서 OpenSSL을 사용하여 해시된 암호를 작성하려면 메시지 요약 명령 **dgst**와 함께 OpenSSL을 실행합니다.

다음 예제는 **SHA-1** 해시 함수를 사용하고 **Base64**로 인코딩된 해시된 암호를 작성하는 방법을 보여줍니다.

```
openssl dgst -sha1 -binary -out <output file name> <input file name>
openssl enc -base64 -in <output file name>
```

입력 파일에는 해시하려는 암호 문자열이 포함됩니다. OpenSSL이 해시된 암호를 출력 파일에 기록합니다.

## Java MessageDigest

다음 예제는 **MD5** 해시 함수를 사용하고 **Base64**로 인코딩된 해시된 암호를 **Java MessageDigest** 클래스를 사용하여 작성하는 방법을 보여줍니다.

```
public static String md5hash(String password) throws Exception{
    MessageDigest digest = java.security.MessageDigest.getInstance("MD5");
    digest.reset();
    digest.update(password.getBytes());
    byte[] hash = digest.digest();
    return new String(org.apache.commons.codec.binary.Base64.encodeBase64(hash));
}
```

다음 예제는 **SHA-1** 해시 함수를 사용하고 **Base64**로 인코딩된 해시된 암호를 **Java MessageDigest** 클래스를 사용하여 작성하는 방법을 보여줍니다.

```
public static String sha1hash(String password) throws Exception{
    MessageDigest digest = java.security.MessageDigest.getInstance("SHA-1");
    digest.reset();
    digest.update(password.getBytes());
    byte[] hash = digest.digest();
    return new String(org.apache.commons.codec.binary.Base64.encodeBase64(hash));
}
```

## 요약 암호

**UsernameToken** 요소에는 다음 하위 요소가 포함됩니다.

- **Username 요소.** PowerCenter 원시 보안 도메인에서 찾을 수 있는 사용자 이름을 포함합니다.
- **Password 요소.** 요약 암호를 포함합니다. 암호는 **Nonce** 요소의 임시 값 및 **Created** 요소의 타임스탬프와 연결된 암호를 해시하여 생성된 값입니다. 암호를 **SHA-1** 해시 함수로 해시하고 **Base64**로 인코딩해야 합니다.

요약 암호 보안의 경우 **Password** 요소의 **Type** 특성을 **PasswordDigest**로 설정해야 합니다.

- **Nonce 요소.** 한 번만 사용할 수 있는 무작위 값인 임시 값을 포함합니다.
- **Created 요소.** 요청이 작성된 시간을 나타내는 타임스탬프 값을 포함합니다. 타임스탬프는 **UTC** 형식을 사용합니다. yyyy-MM-dd'T'HH:mm:ss.SSS'Z'. 예: 2008-08-11T18:06:32.425Z.

SOAP 요청에 포함시키는 임시 값은 한 번만 사용할 수 있습니다. 기본적으로 **Created** 요소의 값으로 표시되는 요청 작성 시간으로부터 300초(5분) 동안 유효합니다. 클라이언트 응용 프로그램은 임시 값이 유효한 시간 내에 요청을 전송해야 합니다. 예를 들어 **Created** 값이 오전 10시에 요청이 작성되었음을 표시합니다. 요청은 오전 10시에서 오전 10시 5분까지 유효합니다. 클라이언트 응용 프로그램이 오전 10시 이전 또는 오전 10시 5분 이후에 웹 서비스 협에 요청을 보낼 경우, 요청 및 임시 값이 유효하지 않게 되고 요청이 실패합니다.

요약 암호는 표준 OASIS 암호 요약 알고리즘을 사용합니다.

```
Password_Digest = Base64 ( SHA-1 ( nonce + created + password ) )
```

임의 도구를 사용하여 임시 값, 타임스탬프 및 요약 암호를 생성할 수 있습니다.



다음 코드는 요약 암호를 사용하는 요청에 대한 보안 헤더의 예제를 보여줍니다.

```
<soapenv:Header>
  <!-- UsernameTokens -->
  <inf:Security>
    <UsernameToken>
      <Username>Administrator</Username>
      <Password Type="PasswordDigest"> Xty5lCAf5SV00AY30tsYq7nv/DI=</Password>
      <Nonce>KjsaeiuDFKJEwkr4332rL=</Nonce>
      <Created>2008-08-12T01:11:47.013Z</Created>
    </UsernameToken>
  </inf:Security>
</soapenv:Header>
```

## 타사 도구를 사용하여 요약 암호 작성

타사 도구(예: Java MessageDigest 클래스)를 사용하여 요약 암호를 작성할 수 있습니다.

다음 예제는 타임스탬프와 임시 값이 있고 Base64로 인코딩된 요약 암호를 Java MessageDigest 클래스를 사용하여 작성하는 방법을 보여줍니다.

```
public static String oasisDigest(String password, String nonce) throws Exception{
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
    String created = sdf.format(new Date());
    System.out.println("Created : " + created);
    System.out.println("Nonce : " + new
String(org.apache.commons.codec.binary.Base64.encodeBase64(nonce.getBytes())));
    String toDigest = nonce + created + password;
    MessageDigest digest = java.security.MessageDigest.getInstance("SHA-1");
    digest.reset();
    digest.update(toDigest.getBytes());
    byte[] hash = digest.digest();
    return new String(org.apache.commons.codec.binary.Base64.encodeBase64(hash));
}
```

웹 서비스 테스트 도구(예: soapUI)를 사용하여 클라이언트 응용 프로그램을 테스트하는 경우 이 도구를 사용하여 클라이언트 요청에 대한 요약 암호를 생성할 수 있습니다.

## 제 6 장

# 웹 서비스 소스 및 대상 작업

이 장에 포함된 항목:

- [웹 서비스 소스 및 대상 개요, 74](#)
- [웹 서비스 소스 및 대상 이해, 74](#)
- [웹 서비스 소스 또는 대상 정의 가져오기, 77](#)
- [소스 또는 대상 정의 작성, 80](#)

## 웹 서비스 소스 및 대상 개요

웹 서비스 워크플로우에서 사용할 매핑을 작성할 때 매핑에 대한 소스 및 대상은 웹 서비스 입력 및 출력 메시지를 정의해야 합니다. 웹 서비스 소스는 웹 서비스 작업의 입력 메시지를 정의하며 웹 서비스 SOAP 요청에 대한 메타데이터를 나타냅니다. 웹 서비스 대상은 웹 서비스 작업의 출력 메시지를 정의하며 웹 서비스 SOAP 응답에 대한 메타데이터를 나타냅니다.

WSDL(Web Services Description Language)은 웹 서비스의 입력 및 출력 메시지를 설명합니다. 작성하려는 웹 서비스 워크플로우에 대한 WSDL이 있는 경우 WSDL에서 소스 및 대상 정의를 가져올 수 있습니다. WSDL이 없는 경우 관계형 또는 플랫폼 파일 소스나 대상의 열을 기반으로 입력 및 출력 메시지를 작성할 수 있습니다. 또는 정의한 열을 기반으로 입력 및 출력 메시지를 작성할 수 있습니다.

웹 서비스 소스 및 대상 정의는 디자이너를 사용하여 작성합니다. 다음 방법으로 웹 서비스 소스 및 대상 정의를 작성할 수 있습니다.

- **WSDL에서 정의를 가져옵니다.** WSDL에 정의된 작업에 대한 소스 또는 대상 정의를 가져옵니다. 소스 정의를 가져오면 디자이너가 입력 메시지의 정의를 가져옵니다. 대상 정의를 가져오면 디자이너가 출력 및 결함 메시지의 정의를 가져옵니다.
- **관계형 또는 플랫폼 파일 소스 및 대상에서 정의를 작성합니다.** 폴더에 정의된 관계형 또는 플랫폼 파일 소스 및 대상 정의를 기반으로 웹 서비스 소스 및 대상 정의를 작성할 수 있습니다. 수동으로 열을 정의하고 데이터 유형 및 열 크기를 지정할 수도 있습니다. 관계형 또는 플랫폼 파일 소스 및 대상에서 웹 서비스 소스 및 대상 정의를 작성하거나 수동으로 열을 정의하는 경우 WSDL이 필요하지 않습니다.

## 웹 서비스 소스 및 대상 이해

XML 소스 및 대상과 유사하게 웹 서비스 소스 및 대상 정의는 XML 보기로 구성됩니다. XML 보기는 입력 및 출력 메시지에 정의된 요소와 특성을 나타내는 열 그룹입니다.

소스 및 대상 정의를 가져오면 디자이너가 WSDL에 정의된 입력 또는 출력 메시지의 요소를 기반으로 XML 보기를 생성합니다. 입력 또는 출력 메시지에 포함되는 MIME 첨부 파일을 위한 보기도 생성합니다.

웹 서비스 소스 및 대상 정의를 작성하면 디자이너가 관계형 또는 플랫폼 파일 소스나 대상에 정의된 열이나 사용자가 수동으로 정의한 열을 기반으로 XML 보기를 작성합니다.

## XML 보기 및 그룹

웹 서비스 소스 및 대상 정의에 포함될 수 있는 보기는 다음과 같습니다.

- **엔벨로프.** SOAP envelope 및 body 요소에 해당하는 XML 보기입니다. 엔벨로프 보기는 입력 또는 출력 메시지에 대한 포트와 기본 키가 포함된 기본 보기입니다.

본문 메시지 부분이 단순한 경우 디자이너는 엔벨로프 보기를 생성합니다.

본문 메시지 부분이 복잡한 경우 디자이너는 추가 본문 보기를 생성할 수 있습니다.

- **요소.** 입력 또는 출력 메시지에 다중 발생 요소가 포함된 경우 작성되는 보기입니다. 디자이너는 입력 또는 출력 메시지에 있는 각각의 다중 발생 요소에 대해 요소 보기를 생성합니다. 요소 보기는 엔벨로프 보기와 *n:1* 관계를 가집니다.

- **유형.** 입력 또는 출력 메시지에 복합 유형의 정의가 포함된 경우 작성되는 보기입니다. 디자이너는 입력 또는 출력 메시지에 있는 각각의 복합 유형 요소에 대해 유형 보기를 생성합니다. 유형 보기는 엔벨로프 보기와 *n:1* 관계를 가집니다.

디자이너는 항목 관계 모드로 가져온 웹 서비스 소스 및 대상 정의에 대해 유형 보기를 생성합니다.

- **헤더.** SOAP header 요소에 해당하는 XML 보기입니다. 헤더 메시지 부분이 복잡한 경우 디자이너는 헤더 보기를 개별 요소 및 유형 보기로 분할할 수 있습니다.

- **결합.** 작업의 출력 메시지에 대해 결합 메시지가 정의된 경우 작성되는 보기입니다. 디자이너는 작업에 대해 정의된 각 결합 메시지에 대해 결합 보기를 생성합니다. 결합 보기는 엔벨로프 보기와 *n:1* 관계를 가집니다. 결합 보기는 웹 서비스 대상 정의에만 포함됩니다.

디자이너는 정규화된 계층 관계 모드 또는 항목 관계 모드로 가져온 웹 서비스 대상 정의에 대해 결합 보기를 생성합니다.

- **첨부 파일.** MIME 첨부 파일이 포함된 WSDL에 대해 생성되는 첨부 파일 보기입니다. 첨부 파일 보기는 엔벨로프 보기와 *n:1* 관계를 가집니다.

디자이너는 MIME 첨부 파일에 대한 요소 정의가 포함된 WSDL을 기반으로 웹 서비스 소스 또는 대상 정의에 대해 첨부 파일 보기를 생성합니다.

### 관련 항목:

- [“WSDL 첨부 파일” 페이지 93](#)

## 소스 정의

디자이너는 입력 메시지의 정의를 기반으로 웹 서비스 소스 정의에 대해 XML 보기를 생성합니다.

다음 테이블에는 디자이너가 웹 서비스 소스 정의에 대해 생성할 수 있는 XML 보기가 표시되어 있습니다.

가져오기 모드	엔벨로프	요소	유형	결합	첨부 파일
항목 관계	예	예	예	아니요	예
정규화된 계층 관계	예	예	아니요	아니요	예

## 대상 정의

WSDL의 작업에 대해 대상 정의를 작성하면 디자이너가 해당 작업과 연결된 모든 결합 메시지와 출력 메시지를 가져옵니다. 작업 내의 함수가 다양한 결합을 발생시키는 경우 디자이너는 대상 정의에 여러 개의 결합 보기를 작성합니다. 결합 메시지는 요청을 처리하는 중에 발생하는 오류를 나타냅니다.

다음 테이블에는 디자이너가 웹 서비스 대상 정의에 대해 생성할 수 있는 XML 보기가 표시되어 있습니다.

가져오기 모드	엔벨로프	요소	유형	결합	Attachment
항목 관계	예	예	예	예	예
정규화된 계층 관계	예	예	아니요	예	예

**참고:** 결합 메시지에 대해 개별 대상 정의를 작성하려면 PowerCenter Designer에서 웹 서비스 옵션을 구성합니다.

## 웹 서비스 소스 및 대상을 가져오거나 작성하기 위한 규칙 및 지침

웹 서비스 소스 및 대상을 가져오거나 작성할 때에는 다음 규칙 및 지침을 사용합니다.

- **복합 관계가 있는 요소에 대해 WSDL을 사용합니다.** 복합 요소 관계가 있는 웹 서비스 소스 또는 대상을 작성하려면 먼저 요소 계층을 정의하는 WSDL을 작성한 후 WSDL에서 소스 또는 대상을 가져옵니다. WSDL을 사용하여 다중 발생 요소 또는 복합 유형 요소가 포함된 웹 서비스 소스 또는 대상을 작성합니다.
- **수동으로 단순 웹 서비스 소스 또는 대상 정의를 정의합니다.** 단순 열 집합이 있고 중첩 요소가 없는 웹 서비스 소스 또는 대상을 작성하려면 수동으로 정의를 작성하거나 관계형 또는 플랫폼 파일 소스나 대상 정의를 사용합니다. 웹 서비스 소스 또는 대상 정의의 모든 열이 여러 번 발생한다고 지정할 수 있습니다.
- **동일한 프로세스를 사용하여 소스 및 대상 정의를 작성합니다.** 소스 정의를 작성할 때 관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 대상 정의를 동시에 작성합니다. 웹 서비스 매핑의 소스 및 대상을 동시에 작성하려면 웹 서비스 정의 작성 창에서 소스 작성 및 대상 작성 옵션이 선택되어 있는지 확인합니다. 예를 들어 소스 분석기에서 소스 > 웹 서비스 공급자 > 웹 서비스 정의 작성을 순서대로 선택합니다. 웹 서비스 정의 작성 창에서 대상 작성 옵션을 선택합니다.
- **동일한 방법을 사용하여 요청-응답 매핑에 대한 소스 및 대상 정의를 작성합니다.** 요청-응답 웹 서비스 매핑을 작성하는 경우 동일한 방법을 사용하여 소스 및 대상 정의를 작성합니다. 예를 들어 WSDL에서 소스 정의를 가져오는 경우 WSDL의 동일한 작업에서 대상 정의를 가져옵니다. 열을 정의하거나 관계형 또는 플랫폼 파일 소스 및 대상을 사용하여 소스 정의를 작성하는 경우 동일한 방법을 사용하여 대상 정의를 작성합니다.
- **WSDL을 사용하여 결합 보기가 있는 대상을 작성합니다.** 대상 정의에 특정 데이터 오류를 표시하기 위한 결합 보기가 있게 하려면 WSDL을 사용하여 웹 서비스 대상 정의를 작성합니다. 플랫폼 파일 또는 관계형 소스나 대상을 기반으로 대상 정의를 작성하는 경우 해당 정의에 결합 보기를 정의할 수 없습니다. 플랫폼 파일 또는 관계형 소스나 대상을 기반으로 웹 서비스 대상 정의를 정의하는 경우 웹 서비스 협이 시스템 결합만을 위한 결합 메시지를 생성할 수 있습니다.
- **WSDL의 입력 및 출력 메시지는 동일한 인코딩 스타일을 가지고 있어야 합니다.** WSDL에서 웹 서비스 소스 및 대상을 가져오는 경우 입력 및 출력 메시지의 인코딩 스타일은 동일해야 합니다. 입력 메시지가 RPC/SOAP 인코딩 스타일을 사용하는 경우 출력 메시지도 RPC/SOAP 인코딩 스타일을 사용해야 합니다. 입력 메시지가 문서/리터럴 스타일을 사용하는 경우 출력 메시지도 문서/리터럴 스타일을 사용해야 합니다.  
관계형 또는 플랫폼 파일 소스나 대상을 기반으로 웹 서비스 소스나 대상 정의를 작성하거나 수동으로 작성하는 경우 디자이너는 입력 및 출력 메시지에 문서/리터럴 인코딩 스타일을 사용합니다.
- **WSDL의 요소는 표준 W3C XML 스키마를 참조할 수 없습니다.** 표준 W3C XML 스키마를 참조하는 요소가 포함된 WSDL에서는 웹 서비스 소스 및 대상 정의를 가져올 수 없습니다.

- **항목 관계 모드를 사용하여 빈 complexType 요소가 있는 WSDL을 가져옵니다.** 웹 서비스 요청의 입력 메시지에 값을 포함하지 않는 complexType 요소가 WSDL에 포함되어 있는 경우 항목 관계 모드를 사용하여 WSDL에서 소스 및 대상 정의를 가져옵니다. 정규화된 계층 모드를 사용하여 WSDL에서 소스 및 대상 정의를 가져오는 경우 사용자가 빈 complexType 요소가 있는 요청을 보내면 웹 서비스에서 결함 응답을 생성합니다.
- **올바른 XML 구문이 있는 WSDL에서 소스 및 대상을 가져옵니다.** 올바르지 않은 WSDL에서 가져오는 경우 디자인어는 WSDL 정의를 웹 서비스 마법사에 제대로 표시할 수 없습니다. 일부 경우에 디자인어는 오류 메시지를 생성하지 않지만 WSDL을 부분적으로 구문 분석하고 구문 분석된 서비스 및 작업만 표시합니다. WSDL에서 정의를 가져오고 웹 서비스 마법사가 올바른 WSDL 정의를 표시하지 않는 경우 XML 파일로 WSDL을 열어서 구문이 올바른지 확인합니다.
- **올바른 구문을 사용하여 2차원 배열을 정의합니다.** WSDL의 complexType 요소를 2차원 문자열 배열로 정의하는 경우 다음 구문을 사용합니다.  

```
wsdl:arrayType="xsd:string[][]"
```

서로 다른 구문을 사용하여 정의된 2차원 배열이 있는 WSDL에서 웹 서비스 소스 및 대상 정의를 가져올 수 없습니다.
- **다수의 XML 보기를 생성하는 WSDL에서 웹 서비스 소스 및 대상을 가져올 수 없습니다.** WSDL에서 생성할 수 있는 XML 보기 수는 400개로 제한됩니다. 400개가 넘는 XML 보기가 있는 웹 서비스 소스 또는 대상을 작성하려면 WSDL 작업 공간에서 수동으로 그룹을 작성합니다.

#### 관련 항목:

- [“SOAP 결함 처리” 페이지 20](#)

## 웹 서비스 소스 또는 대상 정의 가져오기

작성하려는 웹 서비스의 입력 및 출력 메시지를 정의하는 WSDL이 있는 경우 WSDL에서 소스 및 대상 정의를 가져올 수 있습니다. 웹 서비스 마법사를 사용하여 WSDL에서 웹 서비스 소스 또는 대상 정의를 가져올 수 있습니다. 동일한 프로세스를 사용하여 소스 및 대상 정의를 가져올 수 있습니다.

### 가져오기 모드

WSDL에서 가져올 때 사용하는 다음 모드를 기반으로 웹 서비스 소스 또는 대상 정의에 대한 XML 보기를 작성할 수 있습니다.

- **항목 관계.** WSDL에서 가져온 소스 또는 대상 정의에 대한 기본 가져오기 모드입니다. 이 가져오기 모드를 사용하여 하나의 큰 계층 대신 보기 간의 관계를 작성합니다. 항목 관계가 있는 웹 서비스 소스 또는 대상을 작성하면 디자인어가 다중 발생 요소 및 복합 유형에 대해 개별 보기를 생성합니다. 디자인어에는 파생된 모든 복합 유형에 대한 보기가 포함됩니다.
- **정규화된 계층 관계.** 정규화된 계층 보기에서 각 요소 또는 특성은 한 번 나타납니다. 일대다 관계는 보기와 관련된 키가 있는 개별 XML 보기로 됩니다.
- **XML 보기를 생성하지 마십시오.** XML 보기를 정의하지 않고 소스 또는 대상 정의를 작성하려면 이 가져오기 모드를 사용합니다. WSDL 작업 공간을 사용하여 XML 보기 및 포트를 추가할 수 있습니다.

## 관련 항목:

- [“XML 보기를 작성하지 않고 WSDL에서 가져오기” 페이지 79](#)

## 메시지 ID

준비된 매핑의 웹 서비스 소스 및 대상 정의를 사용하려면 메시지 ID를 웹 소스 및 대상 정의에 포함시켜야 합니다. 웹 서비스 헵은 메시지 ID를 기본 키로 사용하여 웹 서비스에 대한 요청 및 응답을 바인딩합니다. 예를 들어 첫 번째 세션은 웹 서비스 소스에서 읽고 관계형 대상에 기록합니다. 두 번째 세션은 관계형 대상을 소스로 사용하고 웹 서비스 대상에 기록합니다. 웹 서비스 헵은 메시지 ID를 사용하여 첫 번째 세션의 요청에 있는 입력 메시지를 두 번째 세션의 응답에 있는 출력 메시지에 연결합니다.

그리드에서 웹 서비스 세션을 실행하는 경우 메시지 ID를 웹 소스 및 대상 정의에도 포함시켜야 합니다. 그리드에서 웹 서비스 세션을 실행하는 경우 통합 서비스가 그리드의 노드에서 다중 DTM 프로세스로 세션 스레드를 분산시킵니다. 통합 서비스는 이 메시지 ID를 사용하여 노드 전체에서 웹 서비스 입력 및 출력 메시지를 연결합니다.

웹 서비스 소스 또는 대상 정의를 작성할 때 메시지 포트를 추가하면 디자이너가 메시지 ID와 클라이언트 포트를 엔벨로프 보기에 추가합니다.

다음 테이블에는 엔벨로프 보기에 추가되는 메시지 ID와 클라이언트 포트가 설명되어 있습니다.

포트 이름	설명
MessageID	웹 서비스 헵이 요청을 받으면 메시지 ID를 생성합니다. 웹 서비스 헵은 이 ID를 사용하여 수신 요청과 전송 응답을 연관시킵니다.
ClientIP	웹 서비스 클라이언트의 TCP/IP 주소입니다.

## 고급 옵션

WSDL에서 웹 서비스 소스 또는 대상을 가져올 때 정의되지 않은 길이가 있는 필드의 길이를 지정하고 XML 열에 대한 이름 지정 규칙을 지정할 수 있습니다.

다음 테이블에는 WSDL에서 웹 서비스 소스 또는 대상을 가져올 때 설정할 수 있는 고급 옵션이 표시되어 있습니다.

옵션	설명
모든 무한 길이 재정의	문자열과 같이 정의되지 않은 길이가 있는 필드에 기본 길이를 지정할 수 있습니다. 기본적으로 이 옵션은 선택되어 있습니다.
XML 열에 대한 이름 생성	<p>이 옵션을 선택하여 스키마의 요소 또는 특성 이름이나 시퀀스 번호로 XML 열의 이름을 지정할 수 있습니다. 이름을 사용하는 경우 다음 옵션 중에서 선택합니다.</p> <ul style="list-style-type: none"> <li>- XML 열이 특성을 참조하는 경우, 요소 이름을 해당 특성의 접두사로 지정합니다. PowerCenter는 XML 열의 이름에 다음 형식을 사용합니다. <i>NameOfElement_NameOfAttribute</i></li> <li>- 모든 XML 열의 XML 보기 이름에 접두사를 지정합니다. PowerCenter는 XML 열의 이름에 다음 형식을 사용합니다. <i>NameOfView_NameOfElement</i></li> <li>- 모든 외래 키 열의 XML 보기 이름에 접두사를 지정합니다. PowerCenter는 생성된 외래 키 열의 이름에 다음 형식을 사용합니다. <i>FK_NameOfView_NameOfParentView_NameOfPKColumn</i></li> </ul> <p>열 이름의 최대 길이는 80자입니다. PowerCenter에서는 80자를 초과하는 열 이름이 잘립니다. 열 이름이 고유하지 않은 경우 PowerCenter는 숫자 접미사를 추가하여 이름을 고유하게 유지합니다.</p>
문자열에 매핑된 anyType 요소의 기본 길이	<p>anyType 유형의 요소에 대해 작성된 문자열 포트의 기본 길이입니다. anyType 유형의 요소에 대해 문자열 유형의 포트를 작성할 수 있습니다. 기본적으로 문자열 길이는 여기에 설정하는 값입니다.</p> <p>문자열 길이를 변경하려면 WSDL 작업 공간에서 웹 서비스 소스 또는 대상 정의를 편집합니다. 기본값은 10,000입니다.</p>

## XML 보기를 작성하지 않고 WSDL에서 가져오기

WSDL에서 소스 또는 대상 정의를 가져오고 수동으로 XML 보기 및 포트를 정의하려는 경우 빈 소스 또는 대상 정의를 작성할 수 있습니다.

예를 들어 입력 메시지에 10개의 요소를 정의하는 WSDL이 있지만 2개의 요소만 소스 정의에 포함시키려고 할 수 있습니다. 이 경우 빈 소스 정의를 작성하고 수동으로 2개의 요소를 정의할 수 있습니다. 대상 정의는 영향을 받지 않습니다. 대상 정의를 가져오고 XML 보기를 작성할 수 있습니다.

XML 보기를 작성하지 않고 WSDL에서 소스 또는 대상 정의를 가져오려면 가져오기 프로세스의 2단계에서 'XML 보기를 생성하지 마십시오.' 옵션을 선택합니다. 빈 소스 또는 대상 정의를 작성한 후 WSDL 작업 공간을 사용하여 XML 보기 및 포트를 정의하고 보기 간의 관계를 정의합니다. 소스 또는 대상 정의의 제목을 마우스 오른쪽 단추로 클릭하고 WSDL 작업 공간을 선택합니다.

### 관련 항목:

- [“WSDL 작업 공간에서 정의 편집” 페이지 85](#)

## WSDL에서 웹 서비스 소스 또는 대상 정의 가져오기

동일한 단계에 따라 WSDL에서 웹 서비스 소스 또는 대상 정의를 가져옵니다. 소스 및 대상 정의는 WSDL에서 서로 다른 요소를 나타내므로 디자이너에서 작성한 소스 정의는 대상 정의와 다릅니다.

로컬로 액세스하거나 URL을 통해 액세스할 수 있는 WSDL에서 웹 서비스 소스 또는 대상을 가져올 수 있습니다. RPC/SOAP 인코딩 또는 문서/리터럴 인코딩 스타일의 WSDL에서 정의를 가져올 수 있습니다.

**참고:** 디자이너는 웹 서비스 대상 정의를 가져올 때 작업 및 대상 유형(예: 출력 또는 대상)을 기반으로 정의의 이름을 지정합니다. 가져온 후에 정의의 이름을 변경하는 경우 메타데이터 확장 탭에서 대상 유형을 확인할 수 있습니다.



WSDL에서 웹 서비스 소스 또는 대상 정의를 가져오려면 다음을 수행합니다.

1. 소스 정의를 가져오려면 소스 분석기에서 소스 > 웹 서비스 공급자 > WSDL에서 가져오기를 순서대로 클릭합니다. 대상 정의를 가져오려면 대상 디자이너에서 대상 > 웹 서비스 공급자 > WSDL에서 가져오기를 순서대로 클릭합니다.
2. 고급 옵션을 클릭합니다.  
XML 보기 작성 및 이름 지정 옵션 창이 나타납니다.
3. 정의되지 않은 길이가 있는 필드에 기본 길이를 지정하고 XML 열의 이름을 생성하는 방법을 선택합니다.
4. 로컬 파일 또는 URL에서 가져오려면 선택합니다.  
URL에서 가져오는 경우 URL을 입력하거나 주소 목록에서 URL을 선택하고 열기를 클릭합니다.  
로컬 파일에서 가져오는 경우 로컬 폴더에서 WSDL 파일을 선택하고 열기를 클릭합니다.
5. 소스 또는 대상 정의를 작성하려는 WSDL에 정의된 작업을 선택합니다.  
**참고:** 오류가 포함된 WSDL에서 가져오는 경우 WSDL에 정의된 서비스, 바인딩, 포트 또는 작업 목록이 웹 서비스 마법사(1단계) 창에 올바르게 표시될 수 없습니다. 이 창에는 비어 있거나 부분적인 WSDL 정의 트리가 표시됩니다. 예를 들어 WSDL에 유형 정의 오류가 포함되어 있는 경우 비어 있는 WSDL 정의 트리가 창에 표시됩니다.
6. 다음을 클릭합니다.  
웹 서비스 정의 작성 옵션 대화 상자가 나타납니다.
7. 가져오기 모드를 선택합니다.  
가져오기 모드는 생성할 XML 보기의 유형을 결정합니다. 항목 관계 또는 계층 관계로 XML 보기를 생성할 수 있습니다. 기본 가져오기 모드는 항목 관계입니다.
8. 정의된 보기 또는 포트 없이 소스 또는 대상 정의를 작성하려면 'XML 보기를 생성하지 마십시오.' 옵션을 선택합니다.  
XML 보기를 생성하지 않는 경우 디자이너는 빈 소스 또는 대상 정의를 작성합니다. 소스 또는 대상 정의에는 보기 또는 포트가 포함되지 않습니다. 이 경우 WSDL 작업 공간을 사용하여 소스 또는 대상 정의에 대한 보기와 포트를 수동으로 추가해야 합니다.
9. 소스 또는 대상 정의에 메시지 및 헤더 포트를 추가할지 여부를 선택합니다.  
자세한 내용은 [“메시지 ID” 페이지 78](#)을 참조하십시오.
10. 동일한 가져오기 프로세스를 사용하여 소스 및 대상 정의를 생성하려면 소스 작성 및 대상 작성 옵션을 둘 다 선택합니다.  
디자이너는 선택한 옵션에 따라 소스 및 대상 정의를 작성합니다.
11. 마침을 클릭합니다.  
웹 서비스 소스 또는 대상 정의가 작업 공간에 나타납니다.

## 소스 또는 대상 정의 작성

웹 서비스 소스 또는 대상 정의를 가져올 WSDL이 없는 경우 관계형 또는 플랫폼 파일 소스나 대상에서 정의를 작성할 수 있습니다. 수동으로 소스 또는 대상에 대한 포트를 정의하고 데이터 유형 및 발생을 지정할 수도 있습니다.

다른 소스 및 대상이나 수동으로 정의된 열에서 웹 서비스 소스 및 대상 정의를 작성하는 경우 디자이너는 항목 관계 모드에서 보기를 작성합니다. 계층 관계 모드에서는 다른 소스 또는 대상에서 웹 서비스 소스 또는 대상을 작성할 수 없습니다.



다음과 같은 관계형 데이터베이스에서 작성된 관계형 소스 및 대상에서 웹 서비스 소스 및 대상 정의를 작성할 수 있습니다.

- Oracle
- DB2
- Informix
- Teradata
- Microsoft SQL Server
- Sybase

관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 소스나 대상을 작성하는 경우 디자이너는 소스 및 대상의 바로 가기를 포함하여 폴더에서 사용 가능한 소스 및 대상을 나열합니다.

동일한 프로세스를 사용하여 소스 및 대상 정의를 작성합니다. 관계형 또는 플랫폼 파일 소스나 대상을 기반으로 웹 서비스 소스나 대상을 작성하는 경우 디자이너 작업 공간에서 소스나 대상 정의의 열을 편집할 수 있습니다.

WSDL에서 웹 서비스 소스 또는 대상 정의를 가져오는 경우 다음 옵션을 선택할 수 있습니다.

- 다중 발생 요소
- 메시지 포트

#### 관련 항목:

- [“디자이너 작업 공간에서 정의 편집” 페이지 83](#)

## 다중 발생 요소

관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 소스 및 대상 정의를 작성하는 경우 소스나 대상 정의의 열이 여러 번 발생하는지 여부를 지정해야 합니다. 다중 발생 요소 옵션을 선택하면 열이 그룹으로 여러 번 발생함을 나타냅니다. 목록의 열은 배열을 나타냅니다.

이 옵션을 사용하여 작성된 웹 서비스 소스 또는 대상을 편집하는 경우 소스 또는 대상 정의의 열에 대한 다중 발생 속성은 변경할 수 없습니다.

## 메시지 포트

웹 소스 및 대상 정의에 메시지 ID를 포함시킬 수 있습니다. 웹 서비스 헵은 메시지 ID를 기본 키로 사용하여 웹 서비스에 대한 요청 및 응답을 바인딩합니다.

**참고:** 디자이너 작업 공간에서 웹 서비스 소스 또는 대상 정의를 편집하는 경우 메시지 및 클라이언트 포트는 웹 서비스 정의 탭에 나타나지 않습니다. 메시지 및 클라이언트 포트는 수정할 수 없습니다.

#### 관련 항목:

- [“메시지 ID” 페이지 78](#)
- [“웹 서비스 정의 탭” 페이지 84](#)

## 관계형 또는 플랫폼 파일 소스나 대상에서 소스 또는 대상 작성

다음 방법 중 하나로 웹 서비스 소스 또는 대상을 작성하려면 이 프로시저를 사용합니다.

- 플랫폼 파일 소스 또는 대상에서 작성
- 관계형 소스 또는 대상에서 작성

- 수동으로 열을 추가하고 열의 이름, 데이터 유형 및 전체 자릿수를 지정하여 작성

관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 소스나 대상을 작성하려면 다음을 수행합니다.

1. 웹 서비스 소스를 작성하려면 소스 분석기에서 소스 > 웹 서비스 공급자 > 웹 서비스 정의 작성을 순서대로 클릭합니다. 웹 서비스 대상을 작성하려면 대상 디자이너에서 대상 > 웹 서비스 공급자 > 웹 서비스 정의 작성을 순서대로 클릭합니다.
2. 소스 및 대상 정의를 사용하려는 웹 서비스 매핑에 대한 이름을 입력합니다.  
디자이너는 웹 서비스 정의 이름을 소스 및 대상 정의의 이름으로 사용합니다. 또한 *\_input* 접미사를 소스 정의 이름에 추가하거나 *\_output* 접미사를 대상 정의 이름에 추가합니다.  
웹 서비스 소스 또는 대상 정의에 포함시키려는 열을 알고 있는 경우 소스 또는 대상 정의에 직접 추가할 수 있습니다. 관계형 또는 플랫폼 파일 소스나 대상이 폴더에 있는 경우 관계형 또는 플랫폼 파일 열을 기반으로 소스나 대상 정의를 작성할 수 있습니다.
3. 목록에 열을 추가하려면 추가 단추를 클릭하고 열 이름, 데이터 유형 및 전체 자릿수를 지정합니다.  
다중 발생 요소가 있는 웹 서비스 소스를 작성하거나 웹 서비스 대상을 작성하는 경우 웹 서비스 힙은 Null 이 아님 속성을 무시합니다.
4. 관계형 또는 플랫폼 파일 소스나 대상 정의의 열을 기반으로 웹 서비스 소스나 대상을 작성하려면 소스/대상에서 가져오기 단추를 클릭하고 소스나 대상 정의를 선택합니다.  
디자이너가 선택된 소스 또는 대상에서 찾은 열을 나열합니다.
5. 확인을 클릭합니다.
6. 필요한 경우 열 이름과 문자열 열의 전체 자릿수를 편집합니다.  
열을 정의에 추가하고 열의 데이터 유형 및 속성을 정의할 수 있습니다. 사용하지 않으려는 열은 삭제할 수 있습니다.  
소스 또는 대상 정의에 정의하는 열이 한 번 발생하는지 아니면 여러 번 발생하는지를 지정할 수 있습니다.
7. 열이 여러 번 발생하는 경우 다중 발생 요소를 선택합니다.  
이 옵션은 열이 그룹으로 두 번 이상 발생함을 나타냅니다. 이 옵션을 선택하면 디자이너가 모든 열을 포함하는 요소 보기를 생성합니다.
8. 소스 정의와 함께 대상 정의도 같이 작성하려는 경우 대상 작성을 클릭하고 3 단계부터 7 단계까지 반복하여 대상 정의에 포트를 추가합니다.  
대상 정의와 함께 소스 정의도 같이 작성하려는 경우 소스 작성을 클릭하고 3 단계부터 7 단계까지 반복하여 소스 정의에 포트를 추가합니다.
9. 메시지 및 클라이언트 포트를 소스 또는 대상 정의에 추가하려면 메시지 포트 추가를 클릭합니다.  
디자이너가 메시지 및 클라이언트 포트를 소스 또는 대상 정의의 엔벨로프 보기에 추가합니다. 동일한 프로세스를 사용하여 소스 및 대상 정의를 작성하는 경우 디자이너는 메시지 및 클라이언트 포트를 소스 및 대상 정의의 엔벨로프 보기에 추가합니다.
10. 확인을 클릭합니다.

디자이너가 웹 서비스 소스 또는 대상 정의를 작성합니다.

소스 및 대상 정의의 XML 보기를 검토하여 보기 및 포트가 웹 서비스 매핑 요구 사항과 일치하는지 확인합니다. 열을 추가, 삭제 또는 수정하려면 디자이너 작업 공간에서 소스 또는 대상 정의를 편집합니다.

## 관련 항목:

- [“다중 발생 요소” 페이지 81](#)
- [“메시지 ID” 페이지 78](#)
- [“디자이너 작업 공간에서 정의 편집” 페이지 83](#)

## 제 7 장

# 웹 서비스 소스 및 대상 편집

이 장에 포함된 항목:

- [웹 서비스 소스 및 대상 편집 개요, 83](#)
- [디자이너 작업 공간에서 정의 편집, 83](#)
- [WSDL 작업 공간에서 정의 편집, 85](#)

## 웹 서비스 소스 및 대상 편집 개요

작성 방법을 기반으로 웹 서비스 소스 또는 대상 정의를 편집할 수 있습니다.

- **WSDL에서 가져온 소스 또는 대상 정의.** WSDL에서 웹 서비스 소스 또는 대상 정의를 가져오는 경우 WSDL 작업 공간에서 소스 또는 대상 정의를 편집할 수 있습니다. WSDL 작업 공간을 사용하여 소스 또는 대상 정의의 보기를 추가, 수정 또는 삭제할 수 있습니다.

디자이너 작업 공간에서 정의를 보고 일부 속성을 편집할 수 있습니다.

- **관계형 또는 플랫 파일 소스 및 대상에서 작성된 소스 또는 대상 정의.** 관계형 또는 플랫 파일 대상을 기반으로 웹 서비스 소스 또는 대상 정의를 작성하는 경우 디자이너 작업 공간에서 열을 편집할 수 있습니다.

WSDL 작업 공간에서 소스 또는 대상 정의를 볼 수 있습니다. WSDL 작업 공간에서는 소스 또는 대상 정의를 편집할 수 없습니다.

**참고:** PowerCenter 버전 9.0.1 이상에서 작성된 웹 서비스 소스 및 대상을 편집할 수 있습니다. 웹 서비스 워크플로우를 실행할 때 오류가 발생하지 않게 하려면 이전 버전에서 업그레이드된 웹 서비스 소스 및 대상을 편집하지 않아야 합니다. 이전 버전의 PowerCenter에서 작성된 웹 서비스 소스 또는 대상을 업데이트하려면 PowerCenter 버전 9.0.1 이상에서 소스 또는 대상을 다시 작성해야 합니다.

## 디자이너 작업 공간에서 정의 편집

디자이너 작업 공간에서 언제든지 웹 서비스 소스 또는 대상 정의에 대한 비즈니스 설명서에 설명을 추가하거나 링크를 지정할 수 있습니다. 소스 또는 대상 정의를 수동으로 작성하거나 관계형 또는 플랫 파일 대상을 기반으로 작성하는 경우 소스 또는 대상 정의의 열 목록을 수정할 수 있습니다. 열을 변경하면 변경 내용이 XML 보기에 즉시 반영됩니다.

소스 또는 대상 정의의 속성을 보거나 편집하려면 소스 분석기의 소스 정의 또는 대상 디자이너의 대상 정의를 두 번 클릭합니다. 또는 소스나 대상 정의의 제목을 마우스 오른쪽 단추로 클릭하고 편집을 선택할 수 있습니다.

다음 탭에서 웹 서비스 소스 및 대상 정의를 보거나 편집할 수 있습니다.

- **테이블.** 테이블 탭에서 소유자 이름과 설명을 제공하고 정의 이름을 변경할 수 있습니다. 테이블 유형은 변경할 수 없습니다.
- **열.** 열 탭에서 문자열 데이터 유형의 전체 자릿수를 편집할 수 있습니다. 또한 비즈니스 이름과 열 설명을 추가할 수 있습니다.
- **특성.** 특성 탭에서 소스 또는 대상 정의의 각 열에 대한 특성 값을 볼 수 있습니다.
- **메타데이터 확장.** 메타데이터 확장 탭에서 웹 서비스 도메인 메타데이터 확장을 볼 수 있습니다. 또한 사용자 정의 메타데이터 도메인의 메타데이터 확장을 추가할 수 있습니다.
- **웹 서비스 정의.** 이 탭은 관계형 또는 플랫폼 파일 대상에서 작성된 소스 또는 대상 정의를 편집하는 경우에 표시됩니다. 소스 또는 대상 정의의 열을 추가, 편집 또는 삭제할 수 있습니다. 변경 내용은 열 탭에 즉시 표시됩니다.

## 테이블 탭

테이블 탭에는 소스 또는 대상 정의에 대한 테이블 정보가 표시됩니다. 소스 또는 대상 정의의 이름을 변경할 수 있습니다. 소스 또는 대상 정의의 소유자와 설명도 수정할 수 있습니다.

## 열 탭

열 탭에는 웹 서비스 소스 또는 대상 정의의 XML 보기가 표시됩니다. 문자열 및 이진 데이터 유형의 전체 자릿수를 편집하고 비즈니스 이름과 열 설명을 추가할 수 있습니다.

문자열 데이터 유형의 기본 전체 자릿수는 WSDL 가져오기 프로세스 동안 무한 길이 데이터의 전체 자릿수가 설정되는 값입니다. WSDL에서 소스 또는 대상 정의를 가져올 때 문자열 데이터 유형의 전체 자릿수를 설정할 수 있습니다. 개별 열의 전체 자릿수는 정의를 편집할 때 설정할 수 있습니다.

**참고:** 매핑 디자이너는 500MB를 초과하는 전체 열 길이가 있는 소스 및 대상 웹 서비스 정의를 사용하는 매핑을 무효화합니다.

## 특성 탭

특성 탭은 웹 서비스 소스 또는 대상 정의의 각 필드에 대한 XPath 및 XMLDataType 값을 표시하는 읽기 전용 탭입니다. 정의에 첨부 파일 그룹이 있는 경우 특성 탭은 데이터 필드에 MIME 유형을 표시합니다.

## 메타데이터 확장 탭

메타데이터 확장 탭에서 메타데이터 확장을 작성할 수 있습니다. 또한 웹 서비스 공급자 도메인의 공급업체 정의 확장을 볼 수 있습니다. 이러한 메타데이터 확장은 메시지 유형을 식별하며 메시지 유형은 입력, 출력 또는 결합일 수 있습니다.

메타데이터 확장에 대한 자세한 내용은 *PowerCenter 리포지토리 가이드*를 참조하십시오.

## 웹 서비스 정의 탭

웹 서비스 정의 탭은 수동으로 정의되거나 관계형 또는 플랫폼 파일 소스나 대상을 기반으로 하는 웹 서비스 소스나 대상 정의에 대해 표시됩니다. 소스 또는 대상 정의의 열을 추가하거나 삭제할 수 있습니다. 열의 이름과 데이터 유형을 변경하고 특정 데이터 유형의 전체 자릿수와 배율을 수정할 수 있습니다. 또한 열이 한 번 발생하는지 아니면 여러 번 발생하는지를 지정할 수 있습니다.

웹 서비스 정의 탭에서 열을 변경하면 해당 변경 내용이 열 탭에 반영됩니다.

# WSDL 작업 공간에서 정의 편집

WSDL에서 소스 또는 대상 정의를 가져오고 XML 보기를 작성하는 경우 WSDL 작업 공간에서 XML 보기, 포트 및 관계를 편집할 수 있습니다. WSDL에서 소스 또는 대상 정의를 가져오지만 XML 보기를 생성하지 않는 경우 WSDL 작업 공간을 사용하여 보기를 작성하고, 구성 요소를 수정하고, 열을 추가하고, 작업 공간의 보기 관계를 유지 관리할 수 있습니다. 소스 또는 대상 정의를 업데이트하면 디자이너가 소스 또는 대상을 포함하는 모든 매핑에 변경 내용을 전달합니다.

WSDL 작업 공간에서 소스 또는 대상 정의를 보거나 편집하려면 소스 분석기의 소스 정의 또는 대상 디자이너의 대상 정의에 대한 제목을 마우스 오른쪽 단추로 클릭합니다. 그런 다음 WSDL 작업 공간을 선택합니다.

WSDL 작업 공간은 XML 편집기에 해당합니다. XML 편집기를 사용하는 방식과 동일하게 WSDL 작업 공간을 사용합니다. 그러나 WSDL 작업 공간은 웹 서비스 소스 및 대상 정의와 관련된 보기의 변경 내용에 대해 유효성 검사를 수행합니다. 또한 WSDL 작업 공간에서는 XML 작업 공간에서 허용되는 일부 태스크를 수행할 수 없습니다.

WSDL 작업 공간에서 수행할 수 없는 태스크는 다음과 같습니다.

- 웹 서비스 소스 또는 대상 정의의 XML 보기를 추가하거나 편집할 때 열을 피벗합니다.
- XML 보기에서 요소 또는 특성을 필터링할 XPath 쿼리 조건자를 작성합니다.
- XML 데이터를 미리 봅니다.
- FileName 열을 XML 보기에 추가합니다.
- 참조 포트를 추가합니다.
- 항목 관계를 다시 작성합니다.
- 열 창에서 XML 보기 옵션을 설정합니다.

## WSDL 작업 공간에 대한 규칙 및 지침

WSDL 작업 공간에서 XML 보기를 편집하거나 웹 서비스 소스 또는 대상 정의에 XML 보기를 추가할 때에는 다음 규칙과 지침을 사용합니다.

- 웹 서비스 매핑에 대한 소스 및 대상 정의에는 웹 서비스 요청, 응답 및 결함 메시지를 위한 SOAP:envelope에 해당하는 엔벨로프 보기가 있어야 합니다.
- 소스 정의는 입력 메시지에 대한 보기를 정의해야 합니다. 출력 또는 결함 메시지에 대한 보기는 정의할 수 없습니다.
- 소스 또는 대상 정의에 대한 루트 그룹의 이름과 루트 그룹의 기본 키는 다음과 같은 이름 지정 규칙을 사용해야 합니다. 여기서 <NameString>은 임의의 영숫자 문자열일 수 있습니다.
  - 루트 그룹의 이름은 Message 또는 X\_<NameString>\_Envelope여야 합니다.
  - 루트 그룹의 기본 키는 PK\_Message 또는 PK\_<NameString>\_Envelope여야 합니다.
  - 루트 그룹과 해당 기본 키의 <NameString>은 동일해야 합니다.
- 대상 정의는 출력 또는 결함 메시지에 대한 보기를 정의해야 합니다. 입력 메시지에 대한 보기는 정의할 수 없습니다.
- anyType 유형 또는 모든 유형의 요소를 정의할 수 있습니다. anyType 유형의 요소에 대해 문자열 포트를 작성하거나 complexType 유형의 요소에 매핑할 수 있습니다.
- 대상 정의의 엔벨로프 보기에는 보기 루트와 엔벨로프 노드로 사용되는 보기 행이 있어야 합니다.
- 소스 또는 대상 정의에서 soap:Body 및 soap:Header 요소의 유형 정의는 변경할 수 없습니다.

- 기본 네임스페이스를 설정하고 소스 또는 대상 정의의 보기에 정의된 네임스페이스에 대한 접두사를 변경할 수 있습니다. 네임스페이스는 변경할 수 없습니다. 다음 문자열은 네임스페이스 접두사로 사용할 수 없습니다.
  - mime
  - wsdl
  - soap
  - soapenc
  - http

## 제 8 장

# 웹 서비스 매핑 작업

이 장에 포함된 항목:

- [웹 서비스 매핑 작업 개요, 87](#)
- [웹 서비스 매핑 유형, 88](#)
- [WSDL에서 매핑 생성, 89](#)
- [관계형 또는 플랫폼 파일 소스나 대상에서 매핑 생성, 90](#)
- [변환 또는 맵렛에서 매핑 생성, 90](#)
- [웹 서비스 매핑에서 대상 인스턴스 편집, 92](#)
- [첨부 파일, 93](#)

## 웹 서비스 매핑 작업 개요

웹 서비스 소스 및 대상 정의를 작성한 후 통합 서비스에서 웹 서비스 요청의 받은 데이터를 처리하고 웹 서비스 응답을 보내는 방법을 결정하는 매핑을 작성합니다. 웹 서비스 매핑은 SOAP 요청으로 입력 메시지를 받고, 데이터를 변환하고, SOAP 응답으로 출력 메시지를 보냅니다.

다른 PowerCenter 매핑을 작성하는 방법과 동일하게 매핑 디자이너에서 웹 서비스 매핑을 작성할 수 있습니다. 웹 서비스 소스 및 대상 정의와 변환을 매핑에 추가합니다.

웹 서비스 소스 정의, 소스 한정자 변환 및 웹 서비스 대상 정의를 포함하는 매핑을 생성할 수도 있습니다. PowerCenter Designer는 웹 서비스 매핑을 생성하는 몇 가지 방법을 제공합니다.

다음 방법으로 웹 서비스 매핑을 생성할 수 있습니다.

- **WSDL에서 소스 및 대상 정의 가져오기.** WSDL에서 웹 서비스 소스 또는 대상을 작성하는 방법과 동일하게 WSDL에서 매핑을 작성할 수 있습니다.
- **관계형 또는 플랫폼 파일 소스나 대상 정의에서 생성.** 관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 소스나 대상을 작성하는 방법과 동일하게 관계형 또는 플랫폼 파일 소스나 대상에서 매핑을 작성할 수 있습니다.
- **변환 또는 맵렛에서 생성.** 단일 입력 및 단일 출력을 사용하여 재사용 가능 변환 또는 맵렛에서 매핑을 작성할 수 있습니다.

매핑을 생성한 후 웹 서비스 매핑을 완료하는 데 필요한 추가 변환, 링크 및 다른 모든 매핑 개체를 추가할 수 있습니다.

## 웹 서비스 매핑 유형

매핑을 작성하여 웹 서비스 클라이언트에서 메시지를 받고, 데이터를 변환하고, 웹 서비스 클라이언트에 응답을 보내거나 PowerCenter에서 지원하는 모든 대상에 응답을 기록할 수 있습니다. 소스 및 대상 정의에 따라 통합 서비스는 SOAP 요청의 일부로 첨부 파일을 받고 보낼 수 있습니다.

플랫 파일 또는 XML 소스 및 대상이 포함된 매핑을 작성하고 웹 서비스 워크플로우에서 사용할 수도 있습니다. 이렇게 하면 파일에서 읽지 않고 첨부 파일로 SOAP 호출을 통해 메시지 데이터를 받을 수 있습니다.

작성하는 매핑은 실행하려는 웹 서비스의 유형에 따라 다릅니다.

- **요청-응답 웹 서비스.** 요청-응답 웹 서비스는 웹 서비스 클라이언트에서 들어오는 요청을 받고, 데이터를 변환하고, 웹 서비스 클라이언트에 응답을 보냅니다. 요청-응답 웹 서비스는 웹 서비스 소스와 웹 서비스 대상을 둘 다 사용합니다.

하나의 매핑 또는 여러 개의 매핑을 작성하여 요청-응답 웹 서비스를 처리할 수 있습니다.

- **하나의 매핑.** 웹 서비스 소스 및 웹 서비스 대상 정의를 둘 다 포함하는 하나의 매핑을 작성합니다. 통합 서비스는 단일 세션에서 들어오는 요청을 받고, 데이터를 변환하고, 응답을 보냅니다.

- **여러 개의 매핑.** 웹 서비스 클라이언트에 응답을 보내기 전에 데이터를 준비하는 여러 개의 매핑을 작성합니다. 각 매핑에 대한 세션이 포함된 워크플로우를 작성할 수 있습니다.

- **단방향 웹 서비스.** 웹 서비스 클라이언트에서 업데이트 및 알림을 받지만 응답을 보낼 필요가 없는 경우 단방향 매핑을 작성할 수 있습니다. 단방향 매핑은 소스에 대한 웹 서비스 클라이언트를 사용합니다. 통합 서비스가 데이터를 대상에 로드하며, 대개 웹 서비스 요청을 통해 실시간 이벤트에서 트리거됩니다.

매핑에 포함시키는 웹 서비스 소스 및 대상 정의는 작성하는 매핑 유형에 따라 다릅니다.

다음 테이블에는 매핑 유형을 기반으로 사용하는 웹 서비스 소스 및 대상 정의가 설명되어 있습니다.

매핑 유형	웹 서비스 소스	웹 서비스 대상
요청-응답	웹 서비스 소스 정의의 인스턴스 하나가 있어야 합니다.	웹 서비스 대상 정의의 인스턴스 하나가 있어야 합니다. 대상 정의에 결합 보기 여러 개가 있을 수 있습니다.
단방향	웹 서비스 소스 정의의 인스턴스 하나가 있어야 합니다.	웹 서비스 대상 정의가 포함되지 않습니다.

관련 항목:

- [“첨부 파일” 페이지 93](#)

## 요청-응답 매핑

요청-응답 매핑은 웹 서비스 소스와 웹 서비스 대상을 사용합니다.

요청-응답 매핑을 작성하는 경우 동일한 방법으로 작성된 소스 및 대상 정의를 사용합니다. WSDL에서 소스 정의를 가져오는 경우 WSDL의 동일한 작업에서 대상 정의를 가져옵니다. 열을 정의하거나 관계형 또는 플랫 파일 소스 및 대상을 사용하여 소스 정의를 작성하는 경우 동일한 방법을 사용하여 대상 정의를 작성합니다.

웹 서비스 소스 및 대상 정의가 동일한 방법을 사용하여 작성되게 하려면 하나의 프로세스에서 소스 및 대상 정의를 작성합니다.

**참고:** WSDL의 동일한 작업에서 소스 및 대상 정의를 가져오지 않거나 동일한 방법을 사용하여 작성하지 않으면 예기치 않은 결과가 발생할 수 있습니다.



SQL 변환을 사용하여 데이터베이스를 업데이트하거나 요청-응답 매핑의 다중 데이터베이스 행을 검색할 수 있습니다. SQL 변환은 다중 데이터베이스 행을 대상에 반환할 수 있습니다. 처리 중에 데이터베이스 오류가 발생하면 SQL 변환은 데이터베이스에서 오류를 수신하고 오류 텍스트를 대상에 보냅니다.

SQL 변환을 사용하여 다중 행을 가져오는 웹 서비스 예제는 PowerCenter와 함께 제공되는 실시간 웹 서비스 예제를 참조하십시오. 기본적으로 실시간 웹 서비스 샘플 프로그램은 다음 디렉터리에 설치되어 있습니다.

`/<PowerCenterInstallDir>/server/samples/WebServices/samples/RealTimeWebServices`

#### 관련 항목:

- [“소스 또는 대상 정의 작성” 페이지 80](#)

## 준비된 매핑

요청-응답 세션을 실행할 때 데이터를 먼저 준비해야 하는 경우 여러 개의 매핑을 작성하여 데이터를 처리할 수 있습니다.

예를 들어 처리해야 하는 메시지 데이터를 받습니다. WebSphere MQ를 통해 외부 시스템에 대한 비동기 호출을 생성해야 합니다. 다음 매핑을 작성합니다.

1. 웹 서비스 소스 정의가 포함된 요청 매핑을 작성합니다. 이 매핑은 플랫폼 파일 대상과 WebSphere MQ 대상에 기록합니다. 모든 메시지 데이터를 두 대상 모두에 기록합니다.  
외부 응용 프로그램은 WebSphere MQ 대상에서 메시지를 받고, 메시지를 처리하고, 다른 WebSphere MQ 대기열에 메시지를 보냅니다.
2. 웹 서비스 대상 정의가 포함된 응답 매핑을 작성합니다. 이 매핑은 첫 번째 매핑의 플랫폼 파일 대상을 소스로 사용합니다. 또한 처리된 데이터가 포함된 WebSphere MQ 대기열을 소스로 사용합니다.

웹 서비스 험은 메시지 ID를 사용하여 준비된 매핑의 요청 및 응답을 연결합니다. 준비된 매핑의 웹 서비스 소스 및 대상 정의를 사용하려면 메시지 ID를 웹 소스 및 대상 정의에 포함시켜야 합니다.

#### 관련 항목:

- [“메시지 ID” 페이지 78](#)

## WSDL에서 매핑 생성

로컬로 액세스하거나 URL을 통해 액세스할 수 있는 WSDL에서 웹 서비스 소스 또는 대상을 가져와서 웹 서비스 매핑을 생성할 수 있습니다.

WSDL에서 소스 및 대상 정의를 가져와서 웹 서비스 매핑을 생성하면 디자이너는 사용자가 선택하는 작업의 입력 메시지에서 소스 정의를 작성합니다. 디자이너는 사용자가 선택하는 작업의 출력 메시지에서 대상 정의를 작성합니다.

WSDL에서 생성된 웹 서비스 매핑에는 다음 개체가 포함됩니다.

- 웹 서비스 소스 정의
- 소스 한정자
- 웹 서비스 대상 정의

디자이너는 대상 인스턴스를 통해 소스 인스턴스로부터 포트를 연결합니다. 매핑을 완료하려면 작성하려는 웹 서비스에 필요한 변환과 다른 매핑 구성 요소를 추가합니다.

WSDL에서 매핑을 생성하려면 다음을 수행합니다.

1. PowerCenter Designer에서 매핑 디자이너를 엽니다.
2. 매핑 > 웹 서비스 매핑 작성 > WSDL에서 가져오기를 클릭합니다.

WSDL에서 소스 및 대상을 가져와서 웹 서비스 매핑을 생성하는 프로시저는 WSDL에서 웹 서비스 소스 또는 대상 정의를 작성하는 프로시저와 동일합니다. 자세한 내용은 [“웹 서비스 소스 또는 대상 정의 가져오기” 페이지 77](#)을 참조하십시오.

3. 리포지토리에 매핑을 저장합니다.

## 관계형 또는 플랫폼 파일 소스나 대상에서 매핑 생성

관계형 또는 플랫폼 파일 소스나 대상을 기반으로 웹 서비스 매핑을 생성할 수 있습니다. 관계형 또는 플랫폼 파일 소스나 대상을 사용하여 웹 서비스 소스 및 대상 정의에서 열을 정의합니다.

관계형 또는 플랫폼 파일 소스나 대상에서 매핑을 생성하는 경우 생성되는 웹 서비스 매핑에는 다음 개체가 포함됩니다.

- 웹 서비스 소스 정의
- 소스 한정자
- 웹 서비스 대상 정의

디자이너는 대상 인스턴스를 통해 소스 인스턴스로부터 포트를 연결합니다. 매핑을 완료하려면 작성하려는 웹 서비스에 필요한 변환과 다른 매핑 구성 요소를 추가합니다.

**참고:** 관계형 또는 플랫폼 파일 소스나 대상에서 매핑을 생성할 때에는 동일한 프로세스에서 웹 서비스 소스 및 대상을 작성합니다. 서로 다른 시간에 작성된 웹 서비스 소스 및 대상이 있는 매핑을 포함하는 워크플로우를 실행하면 워크플로우가 실패할 수 있습니다.

관계형 또는 플랫폼 파일 소스나 대상에서 매핑을 생성하려면 다음을 수행합니다.

1. PowerCenter Designer에서 매핑 디자이너를 엽니다.
2. 매핑 > 웹 서비스 매핑 작성 > 소스/대상 정의 사용을 클릭합니다.

관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 매핑을 생성하는 프로시저는 관계형 또는 플랫폼 파일 소스나 대상에서 웹 서비스 소스 및 대상 정의를 작성하는 프로시저와 동일합니다. 자세한 내용은 [“소스 또는 대상 정의 작성” 페이지 80](#)을 참조하십시오.

3. 리포지토리에 매핑을 저장합니다.

## 변환 또는 맵렛에서 매핑 생성

재사용 가능 변환 또는 맵렛에서 매핑을 생성할 수 있습니다. 디자이너는 변환 또는 맵렛의 포트를 사용하여 웹 서비스 소스 및 대상 정의를 생성합니다.

**참고:** 변환 또는 맵렛에서 매핑을 생성할 때에는 동일한 프로세스에서 웹 서비스 소스 및 대상을 작성합니다. 서로 다른 시간에 작성된 웹 서비스 소스 및 대상이 있는 매핑을 포함하는 워크플로우를 실행하면 워크플로우가 실패할 수 있습니다.

## 재사용 가능 변환에서 매핑 생성

다음 테이블에는 웹 서비스 매핑을 생성할 수 있는 변환 유형이 설명되어 있습니다.

변환	유형	그룹
식	수동	단일
HTTP	수동	하나의 입력 및 하나의 출력
Java	활성 또는 수동	하나의 입력 및 하나의 출력
조회	수동	단일
SQL	활성 또는 수동	하나의 입력 및 하나의 출력
저장 프로시저	수동	단일

웹 서비스 매핑을 생성하는 데 사용하는 변환은 재사용 가능 변환이어야 합니다. 변환을 기반으로 웹 서비스 매핑을 생성하는 경우 디자이너는 폴더에서 사용 가능한 재사용 가능 변환과 재사용 가능 변환의 바로 가기를 나열합니다.

변환에서 웹 서비스 매핑을 생성하는 경우 디자이너는 변환의 포트를 사용하여 소스 및 대상 정의에 대한 열을 정의합니다. 그런 다음 변환 입력 포트를 반영하는 XML 보기가 있는 소스와 변환 출력 포트를 반영하는 XML 보기가 있는 대상 정의가 포함된 매핑을 작성합니다.

변환에서 생성된 웹 서비스 매핑에는 다음 개체가 포함됩니다.

- 웹 서비스 소스 정의
- 소스 한정자
- 매핑을 생성하는 데 사용된 변환
- 웹 서비스 대상 정의

디자이너는 대상 인스턴스를 통해 소스 인스턴스로부터 포트를 연결합니다.

## 맵렛에서 매핑 생성

다음 유형의 맵렛에서 웹 서비스 매핑을 생성할 수 있습니다.

- 하나의 입력 변환과 하나의 출력 변환이 포함된 맵렛
- 활성 변환이 없는 맵렛

맵렛에서 웹 서비스 매핑을 생성하는 경우 디자이너는 프로세스에 허용되는 맵렛과 맵렛의 바로 가기를 나열합니다.

맵렛에서 웹 서비스 매핑을 생성하는 경우 디자이너는 맵렛의 포트를 사용하여 소스 및 대상 정의에 대한 열을 정의합니다. 그런 다음 맵렛 입력 포트를 반영하는 XML 보기가 있는 소스와 맵렛 출력 포트를 반영하는 XML 보기가 있는 대상 정의가 포함된 매핑을 작성합니다.

맵렛에서 생성된 웹 서비스 매핑에는 다음 개체가 포함됩니다.

- 웹 서비스 소스 정의
- 소스 한정자
- 매핑을 생성하는 데 사용된 맵렛
- 웹 서비스 대상 정의

디자이너는 대상 인스턴스를 통해 소스 인스턴스로부터 포트를 연결합니다.

## 재사용 가능 변환 또는 맵렛에서 매핑 생성

재사용 가능 변환 또는 맵렛에서 웹 서비스 매핑을 생성할 때에는 동일한 프로시저를 사용합니다.

재사용 가능 변환 또는 맵렛에서 웹 서비스 매핑을 생성하려면 다음을 수행합니다.

1. 매핑 디자이너에서 매핑 > 웹 서비스 매핑 작성 > 변환/맵렛 정의 사용을 클릭합니다.
2. 웹 서비스 매핑에 사용하려는 변환 또는 맵렛을 선택합니다.  
디자이너가 입력 포트 목록, 데이터 유형, 전체 자릿수 및 배율을 표시합니다.  
매핑의 소스 및 대상 정의에 있는 열이 한 번 발생하는지 아니면 여러 번 발생하는지를 지정할 수 있습니다.
3. 열이 여러 번 발생하는 경우 소스 및 대상이 다중 발생 개체임을 선택합니다.  
이 옵션은 소스 및 대상의 열이 배열임을 나타냅니다. 그룹으로 지정된 컬럼은 여러 번 발생합니다.
4. 메시지 및 클라이언트 포트를 소스 또는 대상 정의에 추가하려면 메시지 포트 추가를 클릭합니다.  
디자이너가 메시지 및 클라이언트 포트를 소스 및 대상 정의의 엔벨로프 보기에 추가합니다.
5. 확인을 클릭합니다.  
디자이너가 웹 서비스 매핑을 작성하고 매핑이 작성되었음을 나타내는 메시지를 표시합니다. 디자이너는 변환 또는 맵렛 이름을 *m\_* 접두사가 붙는 소스 및 대상 정의의 이름으로 사용합니다. 또한 *\_input* 접미사를 소스 정의 이름에 추가하고 *\_output* 접미사를 대상 정의 이름에 추가합니다.

## 웹 서비스 매핑에서 대상 인스턴스 편집

웹 서비스 매핑을 생성한 후 매핑의 대상 인스턴스를 편집할 수 있습니다. 매핑 디자이너에서 대상 인스턴스를 편집할 때 대상 디자이너에서 사용할 수 없는 속성을 편집할 수 있습니다.

웹 서비스 매핑의 대상 정의를 편집하려면 매핑 디자이너에서 대상 정의 인스턴스를 두 번 클릭합니다.

속성 탭에서 편집할 수 있는 변환 특성은 다음과 같습니다.

- 로드 범위
- 부분 로드 복구

### 로드 범위

로드 범위 특성은 대상에 대한 로드 범위를 지정합니다. 웹 서비스 대상 정의의 로드 범위는 변환의 변환 범위와 유사합니다.

다음 값으로 로드 범위를 설정할 수 있습니다.

- 트랜잭션. 로드 범위를 트랜잭션으로 설정하면 통합 서비스에서 트랜잭션의 모든 데이터를 수신할 때 응답을 생성합니다. 대상의 모든 그룹은 동일한 트랜잭션 생성기에서 데이터를 수신해야 합니다.
- 모든 입력. 로드 범위를 모든 입력으로 설정하면 통합 서비스에서 모든 들어오는 데이터를 수신한 후에 응답을 생성합니다. 대상의 여러 그룹은 서로 다른 트랜잭션 생성기에서 데이터를 수신할 수 있습니다. 통합 서비스는 로드 범위가 모든 입력인 경우 커밋을 무시합니다.

기본값은 트랜잭션입니다. 변환 범위에 대한 자세한 내용은 *PowerCenter 고급 워크플로우 가이드*를 참조하십시오.

## 부분 로드 복구

부분 로드 복구 특성은 복구 중에 대상이 이전 부분 로드를 처리하는 방법을 지정합니다.

웹 서비스 대상의 경우 기본값인 없음을 사용합니다. 웹 서비스에는 복구를 지정할 수 없습니다.

## 첨부 파일

다음 방식으로 첨부 파일을 사용하도록 PowerCenter 웹 서비스 워크플로우를 구성할 수 있습니다.

- SOAP 메시지의 첨부 파일로 플랫폼 파일 또는 XML 소스나 대상 사용
- MIME 첨부 파일과 함께 WSDL 사용

## 플랫폼 파일 또는 XML 소스 및 대상 첨부 파일

SOAP 메시지 요청 또는 응답의 첨부 파일로 데이터를 받고 보낼 수 있습니다. 소스 또는 대상은 플랫폼 파일 또는 XML 문서일 수 있습니다. 예를 들어 웹 서비스 응용 프로그램의 메시지가 포함된 플랫폼 파일에 FTP를 사용하여 주기적으로 액세스합니다. FTP를 사용하는 대신 웹 서비스 워크플로우를 작성하여 SOAP 요청의 첨부 파일로 플랫폼 파일에서 데이터를 받을 수 있습니다.

SOAP 메시지 요청의 첨부 파일로 데이터를 받으려면 매핑의 플랫폼 파일 또는 XML 소스 정의를 사용합니다. 웹 서비스의 소스로 플랫폼 파일을 사용하려면 플랫폼 파일용 웹 서비스 공급자 관독기를 사용하도록 관독기를 구성합니다. 매핑을 실행하는 웹 서비스 세션을 편집합니다. 세션 속성에서 매핑 탭을 클릭하고 소스를 선택합니다. 관독기를 플랫폼 파일 관독기에서 플랫폼 파일용 웹 서비스 공급자 관독기로 변경합니다.

SOAP 메시지 응답의 첨부 파일로 데이터를 보내려면 매핑의 플랫폼 파일 또는 XML 대상 정의를 사용합니다. 웹 서비스의 대상으로 플랫폼 파일을 사용하려면 플랫폼 파일용 웹 서비스 공급자 기록기를 사용하도록 기록기를 구성합니다. 매핑을 실행하는 웹 서비스 세션을 편집합니다. 세션 속성에서 매핑 탭을 클릭하고 대상을 선택합니다. 기록기를 플랫폼 파일 기록기에서 플랫폼 파일용 웹 서비스 공급자 기록기로 변경합니다.

## WSDL 첨부 파일

소스 및 대상 정의에 따라 SOAP 요청의 일부로 첨부 파일을 받고 보낼 수 있습니다. 첨부 파일은 텍스트 파일(예: XML 문서)이어야 합니다. 이진 문서(예: JPEG, GIF 또는 PDF 파일)는 첨부할 수 없습니다. 예를 들어 Oracle 데이터베이스에서 XML 문서를 추출하여 응답 메시지의 첨부 파일로 웹 서비스 클라이언트에 전달할 수 있습니다.

이진 파일을 소스로 사용하려면 웹 서비스 소스에 전달하기 전에 파일을 hexbinary 또는 base64binary로 변환합니다. hexbinary 또는 base64binary 파일은 텍스트 파일로 처리됩니다. 이와 비슷하게 웹 서비스 대상에서 생성한 응답 텍스트 파일을 이진 파일로 변환할 수 있습니다.

다음 테이블에는 웹 서비스 정의의 첨부 파일 그룹 포트가 설명되어 있습니다.

포트 이름	설명
FK_Att_Name	루트 그룹의 PK_Message를 가리키는 생성된 외래 키입니다.
Att_Data_Name	첨부 파일을 포함합니다. 특성 탭에서 첨부 파일에 대한 MIME 유형을 볼 수 있습니다.
Att_Index_Name	메시지의 각 첨부 파일에 대한 고유 식별자입니다.
Att_Type_Name	첨부 파일의 유형입니다.

## MIME 첨부 파일과 함께 WSDL 사용에 대한 규칙 및 지침

첨부 파일에 대한 작업을 수행할 때에는 다음 규칙과 지침을 사용합니다.

- 요청 또는 응답은 하나의 첨부 파일을 포함할 수 있습니다.
- 첨부 파일은 텍스트 파일이어야 하며 UTF-8 코드 페이지 또는 UTF-8 코드 페이지의 하위 집합인 코드 페이지를 사용해야 합니다.
- 요청 또는 응답을 통해 첨부 파일을 전달하려면 첨부 파일 그룹의 모든 포트를 연결해야 합니다.
- 매핑의 정의에 첨부 파일 그룹이 포함되어 있지만 첨부 파일을 보내거나 받지 않으려는 경우 그룹에 있는 어떠한 포트도 연결하지 않아야 합니다.
- 다른 소스에서 메시지를 받고 각 메시지에 첨부 파일이 포함되어 있는 경우 시퀀스 생성기 변환을 사용하여 응답에 포함시켜 보내는 각 첨부 파일에 대해 고유 인덱스를 생성합니다.
- 첨부 파일을 보내거나 받으려면 MIME 첨부 파일을 지원하는 툴킷을 사용하여 클라이언트 응용 프로그램을 작성합니다.

## 제 9 장

# 웹 서비스 워크플로우에 대한 작업

이 장에 포함된 항목:

- [웹 서비스 워크플로우에 대한 작업 개요, 95](#)
- [웹 서비스 워크플로우 작성 및 구성, 96](#)
- [웹 서비스 공급자 판독기 및 기록기 구성, 99](#)
- [웹 서비스 세션을 위한 파티션 구성, 101](#)
- [웹 서비스 워크플로우 문제 해결, 101](#)

## 웹 서비스 워크플로우에 대한 작업 개요

웹 서비스 워크플로우는 워크플로우 관리자를 사용하여 작성합니다. 웹 서비스 워크플로우를 작성하려면 워크플로우에 대해 웹 서비스 옵션을 활성화한 후 웹 서비스 속성을 구성합니다.

웹 서비스 워크플로우에서 세션을 작성하는 경우 작성되는 세션을 웹 서비스 세션이라고 합니다. 다음 유형의 매핑을 웹 서비스 세션에 포함시킬 수 있습니다.

- 웹 서비스 매핑
- 플랫 파일 매핑
- XML 매핑

웹 서비스 세션은 웹 서비스 공급자 판독기 및 기록기를 사용합니다. 웹 서비스 매핑에 웹 서비스 소스와 대상이 포함되어 있으면 기본적으로 세션에서 웹 서비스 공급자 판독기 및 기록기를 사용합니다. 웹 서비스 매핑에 플랫 파일, XML 소스 또는 대상이 포함되어 있으면 판독기 및 기록기 유형을 웹 서비스 공급자 판독기 또는 기록기로 변경해야 합니다.

웹 서비스 세션에 XML, 플랫 파일 소스 또는 대상이 포함되어 있으면 클라이언트 응용 프로그램이 요청을 SOAP 메시지에 대한 MIME 첨부 파일로 웹 서비스 협에 보냅니다. 첨부 파일을 보내거나 받으려면 MIME 첨부 파일을 지원하는 툴킷을 사용하여 클라이언트 응용 프로그램을 작성해야 합니다.

웹 서비스 협이 웹 서비스 워크플로우를 실행하기 위한 SOAP 메시지 요청을 받으면 통합 서비스에 요청을 전달합니다. 통합 서비스는 웹 서비스 요청을 실행한 후 웹 서비스 협에 응답을 전달합니다. 웹 서비스 협은 SOAP 메시지 응답을 생성하고 웹 서비스 클라이언트에 응답을 다시 전달합니다.

웹 서비스 소스 및 대상 정의가 포함된 세션에서 다중 파티션을 설정할 수 있습니다. 통합 서비스는 세션의 소스, 대상 및 파티션 수에 따라 웹 서비스 협에 대한 연결을 작성합니다.

**참고:** 웹 서비스 워크플로우를 실행하려면 Administrator 도구에서 웹 서비스 협을 작성하고 구성한 후 실행하려는 웹 서비스 워크플로우가 포함된 리포지토리와 연결해야 합니다.

# 웹 서비스 워크플로우 작성 및 구성

웹 서비스 워크플로우를 작성하려면 웹 서비스 매핑을 처리하도록 워크플로우를 구성하고 워크플로우 속성에서 웹 서비스 옵션을 활성화합니다. 웹 서비스 클라이언트가 워크플로우를 실행할 수 있도록 웹 서비스를 구성할 수 있습니다.

웹 서비스 워크플로우를 작성하고 구성하려면 다음 단계를 수행합니다.

- 웹 서비스 워크플로우를 작성합니다.
- 웹 서비스를 구성합니다.

## 웹 서비스 워크플로우 작성

웹 서비스 워크플로우를 작성하려면 워크플로우에 대해 웹 서비스 옵션을 활성화합니다. 그런 다음 웹 서비스를 구성하고 웹 서비스 세션을 워크플로우에 추가합니다. 웹 서비스 세션은 웹 서비스 매핑을 기반으로 합니다.

대부분의 경우 웹 서비스 워크플로우에는 입력 메시지를 위한 웹 서비스 소스 하나와 출력 메시지를 위한 웹 서비스 대상 하나가 포함되어 있습니다. 세션은 대상의 다중 결합 보기에 기록할 수 있습니다. 단방향 웹 서비스는 응답을 보내지 않으므로 웹 서비스 대상이 필요하지 않습니다.

웹 서비스 워크플로우를 작성할 때 통합 서비스를 지정해야 합니다. 통합 서비스 찾아보기 단추를 사용하여 가능한 통합 서비스 목록에서 선택합니다.

웹 서비스 워크플로우를 작성한 후, 웹 서비스 매핑을 실행하기 위한 세션을 추가할 수 있습니다. 세션을 작성하고 임의의 워크플로우에 추가하는 방식과 동일하게 세션을 작성하고 웹 서비스 워크플로우에 추가합니다.

**참고:** 워크플로우 마법사를 사용하여 웹 서비스 워크플로우를 작성하지 않아야 합니다. 워크플로우 마법사를 사용하는 경우 웹 서비스 옵션을 선택할 수 없습니다.

웹 서비스 워크플로우를 작성하는 방법은 다음과 같습니다.

1. 워크플로우 관리자에서 워크플로우 디자인어를 열고 워크플로우 > 작성을 클릭합니다.
2. 워크플로우 이름을 입력합니다.
3. 워크플로우를 실행할 통합 서비스를 선택하려면 통합 서비스 찾아보기 단추를 클릭하고 목록에서 선택합니다.
4. 웹 서비스 옵션을 활성화하고 서비스 구성을 클릭하여 웹 서비스 워크플로우를 구성합니다.  
웹 서비스 옵션을 활성화하면 기본적으로 동시 실행 구성 옵션이 활성화됩니다. 웹 서비스 워크플로우 구성 속성에는 웹 서비스의 동시 실행을 위한 설정이 포함되어 있습니다.
5. 필요에 따라 웹 서비스 워크플로우 속성을 구성합니다.
6. 확인을 클릭합니다.

## 웹 서비스 워크플로우 구성

웹 서비스 워크플로우를 구성할 때 웹 서비스 워크플로우를 실행할 웹 서비스 협을 할당하고 웹 서비스 실행 및 액세스를 위한 옵션을 구성할 수 있습니다.



다음 표에는 웹 서비스에 대해 구성할 수 있는 속성이 설명되어 있습니다.

속성	설명
서비스 이름	웹 서비스의 이름입니다. 사용자가 워크플로우를 체크 인하면 웹 서비스 협이 이 이름을 게시하고 해당 서비스가 표시됩니다. 기본 이름은 리포지토리 이름, 폴더 이름 및 워크플로우 이름으로 연결되어 있습니다. 이 이름은 고유해야 합니다.
제한 시간(초)	요청 제한 시간이 초과되기 전에 웹 서비스 협이 요청을 처리하고 SOAP 응답을 생성하는 데 사용할 수 있는 최대 시간입니다. 웹 서비스 협이 제한 시간 내에 응답을 생성할 수 없는 경우 웹 서비스 클라이언트에 결함 메시지를 보내고 연결을 끊습니다. 기본값은 60초입니다. 0으로 설정하면 제한 시간이 비활성화됩니다.
서비스 시간 임계값(밀리초)	다음 요청을 처리하기 위해 다른 인스턴스를 시작하기 전에 웹 서비스 협이 요청을 처리하는 데 사용할 수 있는 최대 시간입니다. 서비스 시간은 웹 서비스 협이 SOAP 요청을 받는 시간부터 시작하여 SOAP 응답을 생성하는 시간까지입니다. 웹 서비스 협에서 요청을 처리하는 데 걸리는 평균 시간이 서비스 시간을 초과하는 경우 웹 서비스 협은 웹 서비스의 새 인스턴스를 시작하여 새 요청을 처리합니다. 예를 들어 서비스 시간이 1000밀리초로 설정되어 있습니다. 웹 서비스 협이 1000밀리초 내에 요청을 처리할 수 없는 경우 웹 서비스의 다른 인스턴스를 시작하여 다음 SOAP 요청을 처리합니다. 기본값은 1000입니다. 참고: 성능 저하를 방지하려면 서비스 시간 임계값을 100밀리초 미만으로 설정하지 않아야 합니다.
웹 서비스 협	워크플로우를 실행할 웹 서비스 협 서비스입니다. 찾아보기 단추를 클릭하여 웹 서비스 워크플로우를 실행할 웹 서비스 협 서비스를 하나 이상 선택합니다. 기본적으로 웹 서비스 워크플로우는 리포지토리와 연결된 모든 웹 서비스 협 서비스에서 실행할 수 있습니다. <b>참고:</b> 워크플로우를 수동으로 시작할 계획인 경우 워크플로우를 실행할 웹 서비스 협을 선택합니다. 모든 협에서 실행은 선택하지 않습니다. 웹 서비스 워크플로우를 시작하기 전에 웹 서비스 협이 활성화되었는지 확인합니다.
협당 최대 실행 횟수	웹 서비스 협에서 시작할 수 있는 최대 웹 서비스 인스턴스 수입니다. 인스턴스가 동적 또는 수동으로 시작되었는지에 상관없이 웹 서비스 협에서 실행되는 웹 서비스 워크플로우의 모든 인스턴스가 해당 수에 포함됩니다. 최대값에 도달하면 웹 서비스 협이 다른 웹 서비스 인스턴스를 시작할 수 없습니다.
보호됨	웹 서비스를 실행하기 전에 인증이 필요합니다. 웹 서비스 협은 사용자 이름 토큰을 기반으로 요청을 인증합니다. 서비스를 보호하거나 공용으로 사용하도록 선택할 수 있습니다. 워크플로우를 실행할 수 있는 모든 PowerCenter 사용자는 워크플로우 관리자, <i>pmcmd</i> 또는 LMAPI를 사용하여 보호된 웹 서비스 워크플로우를 실행할 수 있습니다. 웹 서비스가 보호되지 않은 경우 모든 웹 서비스 클라이언트는 인증 과정 없이 서비스를 시작할 수 있습니다. 자세한 내용은 <a href="#">“클라이언트 요청에 보안 추가” 페이지 69</a> 을 참조하십시오.

속성	설명
표시	웹 서비스 협 콘솔에 웹 서비스를 표시합니다. 서비스를 표시하는 경우 웹 서비스 협은 웹 서비스 협 콘솔에 웹 서비스와 WSDL을 게시합니다. 웹 서비스 협 콘솔에서 웹 서비스를 테스트하고 WSDL을 보거나 다운로드할 수 있습니다. 서비스가 표시되지 않으면 웹 서비스 협은 웹 서비스 WSDL을 게시하지 않습니다.
실행 가능	웹 서비스 클라이언트가 웹 서비스 협에 요청을 보내서 워크플로우를 시작할 수 있게 지원됩니다. 웹 서비스 워크플로우가 실행 가능한 경우 웹 서비스 클라이언트 요청을 통해 워크플로우를 시작하거나 워크플로우가 실행 중인 동안 웹 서비스를 실행할 수 있습니다. 웹 서비스 클라이언트가 워크플로우를 시작할 수 있게 하려면 요청 시 실행할 워크플로우를 예약합니다. 웹 서비스 워크플로우가 실행 가능하지 않은 경우 웹 서비스 클라이언트는 워크플로우가 실행 중인 동안 웹 서비스를 호출할 수 있지만 워크플로우를 시작할 수 없습니다. 비활성화된 경우 워크플로우 관리자, LMAPi 또는 <i>pmcmd</i> 를 통해 워크플로우를 시작할 수 있습니다.

## 웹 서비스 워크플로우 동시 실행

웹 서비스 협이 웹 서비스의 속성에 설정된 값과 리소스 가용성을 기반으로 웹 서비스 워크플로우의 새 인스턴스를 시작할 시기를 결정합니다. 또한 웹 서비스 공급자 관독기의 속성에 설정된 값을 기반으로 웹 서비스 워크플로우의 인스턴스를 종료할 시기를 결정합니다.

### 새 인스턴스 시작

웹 서비스 협이 현재 리소스 사용량과 다음과 같은 웹 서비스 워크플로우의 속성을 기반으로 웹 서비스 워크플로우의 다른 인스턴스를 시작할 시기를 결정합니다.

- **서비스 시간 임계값.** 웹 서비스 협에서 웹 서비스를 처리하는 데 걸리는 평균 시간이 서비스 시간 임계값을 초과하는 경우 웹 서비스 협이 웹 서비스의 다른 인스턴스를 시작합니다.
- **협당 최대 실행 횟수.** 인스턴스 수가 협당 최대 실행 횟수에 도달할 때까지 웹 서비스 협이 웹 서비스 인스턴스를 시작합니다. 최대 실행 횟수에 도달하면 웹 서비스 협은 평균 서비스 시간 임계값이 초과하지 않은 경우에도 웹 서비스의 새 인스턴스를 시작하지 않습니다.

### 인스턴스 종료

웹 서비스 협이 현재 리소스 사용량과 다음과 같은 웹 서비스 공급자 관독기의 속성을 기반으로 웹 서비스 워크플로우 인스턴스를 종료합니다.

- **유휴 시간.** 워크플로우 인스턴스에서 유휴 시간 내에 어떤 요청도 받지 않는 경우 웹 서비스 협이 워크플로우 인스턴스를 종료합니다.
- **메시지 수.** 워크플로우 인스턴스에서 수신한 메시지 수가 통합 서비스에서 세션 내에 읽도록 구성된 최대 메시지 수에 도달하면 웹 서비스 협이 워크플로우 인스턴스를 종료합니다.
- **관독기 시간 제한.** 통합 서비스가 웹 서비스 협에서 입력 메시지를 읽을 수 있는 최대 시간에 도달하면 통합 서비스가 웹 서비스 협의 입력 메시지 읽기를 중지합니다. 웹 서비스 협이 워크플로우 인스턴스를 종료합니다.

이러한 속성이 워크플로우에 대해 구성된 임계값에 도달하는 경우 웹 서비스 협이 웹 서비스 워크플로우 인스턴스를 종료합니다.

### 관련 항목:

- [“웹 서비스 워크플로우 구성” 페이지 96](#)
- [“웹 서비스 공급자 관독기 구성” 페이지 99](#)

# 웹 서비스 공급자 판독기 및 기록기 구성

웹 서비스 세션을 구성할 때 세션 판독기 및 기록기를 구성할 수 있습니다. 기본적으로 웹 서비스 소스 및 대상이 있는 웹 서비스 세션은 웹 서비스 공급자 판독기 및 기록기를 사용합니다.

웹 서비스 세션에 플랫폼 파일, XML 소스 또는 대상이 포함되어 있으면 웹 서비스 공급자 판독기 또는 기록기를 사용하도록 세션을 구성해야 합니다. 웹 서비스 협은 SOAP 메시지에 대한 MIME 첨부 파일로 요청 및 응답을 보냅니다.

웹 서비스 세션에 대해 판독기를 구성할 때에는 종료 조건(예: 유휴 시간 및 메시지 수)을 구성합니다.

웹 서비스 세션에 대해 기록기를 구성할 때에는 통합 서비스에서 대상 데이터를 캐시하는 데 사용하는 캐싱 정보를 구성합니다. 대상 데이터의 출력 형식도 구성할 수 있습니다.

워크플로우 관리자를 사용하여 웹 서비스 세션을 구성합니다. 워크플로우 디자이너에서 웹 서비스 워크플로우의 세션을 편집합니다. 웹 서비스 공급자 판독기를 구성하려면 매핑 탭을 클릭하고 소스를 선택합니다. 웹 서비스 공급자 기록기를 구성하려면 대상을 선택합니다.

## 관련 항목:

- [“첨부 파일” 페이지 93](#)

## 웹 서비스 공급자 판독기 구성

웹 서비스 공급자 판독기에 대해 구성하는 속성은 매핑에 사용된 소스 유형에 따라 다릅니다.

다음 표에는 웹 서비스 세션에 대해 구성하는 소스 속성이 설명되어 있습니다.

속성	판독기 유형	설명
유휴 시간	<ul style="list-style-type: none"><li>- 웹 서비스</li><li>- 웹 서비스 공급자 판독기 플랫폼 파일</li><li>- 웹 서비스 공급자 판독기 XML 파일</li></ul>	통합 서비스가 소스에서 읽기 작업을 중지하고 웹 서비스 협이 워크플로우 인스턴스를 종료하기 전에 통합 서비스가 메시지 수신을 대기하는 시간(초)입니다. 이 속성의 조건이 충족되면 세션이 중지됩니다. 기본값은 180입니다.
메시지 수	<ul style="list-style-type: none"><li>- 웹 서비스</li><li>- 웹 서비스 공급자 판독기 플랫폼 파일</li><li>- 웹 서비스 공급자 판독기 XML 파일</li></ul>	웹 서비스 협이 워크플로우 인스턴스를 종료하기 전에 통합 서비스가 읽는 메시지 수입니다. -1 값은 메시지 수에 제한이 없음을 나타냅니다. 세션이 플랫폼 파일 또는 XML 대상을 사용하는 경우 메시지 수를 항상 1로 구성합니다. 자세한 내용은 <a href="#">“XML 및 플랫폼 파일 세션을 위한 판독기 및 기록기 구성” 페이지 101</a> 을 참조하십시오. 이 속성의 조건이 충족되면 세션이 중지됩니다. 기본값은 -1입니다.
판독기 시간 제한	<ul style="list-style-type: none"><li>- 웹 서비스</li><li>- 웹 서비스 공급자 판독기 플랫폼 파일</li><li>- 웹 서비스 공급자 판독기 XML 파일</li></ul>	통합 서비스가 웹 서비스 협에서 소스 메시지를 읽는 시간(초)입니다. 예를 들어 판독기 시간 제한을 10으로 설정하면 통합 서비스가 10초 후에 웹 서비스 협에서 읽기 작업을 중지합니다. 이 속성의 조건이 충족되면 세션이 중지됩니다. 기본값은 0이며 시간에 제한이 없음을 나타냅니다.
빈 콘텐츠를 Null로 처리	웹 서비스 공급자 판독기 XML 파일	빈 문자열을 Null 값으로 처리합니다. 기본적으로 빈 콘텐츠는 Null이 아닙니다.
복구 캐시 폴더	해당 없음	이 속성은 웹 서비스 공급자에서 사용하지 않습니다.

## 웹 서비스 공급자 기록기 구성

웹 서비스 공급자 기록기에 대해 세션 속성을 구성할 때 캐시 크기와 캐시 디렉터리를 구성합니다.

다음 표에는 웹 서비스 세션에 대해 구성하는 대상 속성이 설명되어 있습니다.

속성	기록기 유형	설명
XML 날짜/시간 형식	웹 서비스 공급자 기록기 XML 파일	서비스 대상으로 전달되는 데이터의 날짜/시간 형식입니다. 전체 자릿수는 나노초까지입니다. 다음 날짜/시간 형식에서 선택합니다. - 로컬 시간. 통합 서비스 서버 시간대에 따른 시간입니다. - 시간대가 있는 로컬 시간. 통합 서비스 시간대와 그리니치 표준시 사이의 시간 차이입니다. - UTC. 그리니치 표준시.
Null 콘텐츠 표현	웹 서비스 공급자 기록기 XML 파일	Null 콘텐츠가 대상에서 표현되는 방식을 결정합니다. 다음 옵션 중에서 선택하십시오. - 태그 없음. 태그를 출력하지 않음. - 빈 콘텐츠에 태그 지정. 태그만 출력. 기본값은 태그 없음입니다.
빈 문자열 콘텐츠 표현	웹 서비스 공급자 기록기 XML 파일	빈 문자열이 대상에서 표현되는 방식을 결정합니다. 다음 옵션 중에서 선택하십시오. - 태그 없음. 태그를 출력하지 않음. - 빈 콘텐츠에 태그 지정. 태그만 출력. 기본값은 빈 콘텐츠에 태그 지정입니다.
중복 그룹 행 처리	웹 서비스 공급자 기록기 XML 파일	통합 서비스가 세션 동안 중복 그룹 행을 처리하는 방법을 결정합니다. 다음 옵션 중에서 선택하십시오. - 첫 번째 행. 통합 서비스가 첫 번째 중복 행을 대상에 전달합니다. 통합 서비스는 이 행 뒤에서 처리하는 동일한 기본 키가 있는 행을 거부합니다. - 마지막 행. 통합 서비스가 마지막 중복 행을 대상에 전달합니다. - 오류. 통합 서비스가 첫 번째 행을 대상에 전달합니다. 중복 기본 키가 뒤에 오는 행은 오류 수를 증가시킵니다. 오류 수가 오류 임계값을 초과하면 세션이 실패합니다. 기본값은 오류입니다.
고아 행 처리	웹 서비스 공급자 기록기 XML 파일	통합 서비스가 세션 동안 고아 행을 처리하는 방법을 결정합니다. 다음 옵션 중에서 선택하십시오. - 무시. 통합 서비스가 고아 행을 무시합니다. - 오류. 오류 수가 오류 임계값을 초과하면 세션이 실패합니다.
캐시 크기	- 웹 서비스 - 웹 서비스 공급자 기록기 XML 파일	기록기에서 사용하는 메모리 캐시의 총 크기(바이트)입니다. 대상 인스턴스의 각 그룹에 대한 기본 키 및 외래 키 인덱스 캐시 각각 하나와 모든 그룹에 대한 데이터 캐시 하나가 포함됩니다. 총 캐시 요구 사항은 각 대상 그룹에 대한 데이터 캐시 및 인덱스 캐시 요구 사항의 합계입니다. 기본값은 10,000,000바이트입니다.
캐시 디렉터리	- 웹 서비스 - 웹 서비스 공급자 기록기 XML 파일	대상 캐시 파일의 디렉터리입니다. 기본값은 \$PMCacheDir 서비스 프로세스 변수입니다.

기록기 유형을 웹 서비스 공급자 기록기로 변경할 때에는 다음 규칙과 지침을 사용합니다.

- 플랫폼 파일 대상에 대한 기록기 유형을 변경하면 통합 서비스가 대상 메시지를 캐시하지 않습니다.
- 플랫폼 파일 또는 XML 대상에 대한 기록기 유형을 변경하는 경우 결합 메시지가 아닌 웹 서비스 출력 메시지로 대상을 사용합니다.
- XML 대상에 대한 기록기 유형을 변경하는 경우 XML 기록기 속성도 구성해야 합니다.

## XML 및 플랫폼 파일 세션을 위한 판독기 및 기록기 구성

XML 또는 플랫폼 파일 소스와 대상을 포함하는 매핑을 기반으로 웹 서비스 세션을 작성하려면 판독기 또는 기록기 유형을 웹 서비스 공급자 판독기 또는 기록기로 설정합니다. XML 또는 플랫폼 파일 판독기로 웹 서비스 워크플로우를 실행하기 위해 클라이언트 응용 프로그램은 요청을 SOAP 메시지에 대한 MIME 첨부 파일로 웹 서비스 협에 보냅니다. 웹 서비스 협은 첨부 파일이 있는 SOAP 메시지를 통합 서비스에 전달하고 통합 서비스가 첨부 파일을 처리합니다.

웹 서비스 워크플로우가 XML 또는 플랫폼 파일 기록기로 구성되는 경우 통합 서비스는 응답을 생성하고 웹 서비스 협에 응답을 전달합니다. 웹 서비스 협은 응답을 SOAP 메시지에 대한 MIME 첨부 파일로 웹 서비스 클라이언트에 다시 보냅니다.

플랫폼 파일, XML 소스 또는 대상으로 요청-응답 웹 서비스 세션을 구성할 때에는 다음 규칙과 지침을 사용합니다.

- 판독기 속성에서 메시지 수를 1로 설정합니다.
- 판독기 또는 기록기 유형을 웹 서비스 공급자로 변경하는 경우 세션 하나를 워크플로우에 포함시킵니다.
- 세션 속성에서 판독기 또는 기록기 유형을 웹 서비스 공급자 판독기 또는 기록기로 변경하는 경우 MIME 첨부 파일을 지원하는 토크를 사용하여 클라이언트 응용 프로그램을 작성해야 합니다.

## 웹 서비스 세션을 위한 파티션 구성

웹 서비스 소스 및 대상 정의가 포함된 세션에 다중 파티션을 설정하면 통합 서비스가 세션의 소스, 대상 및 파티션 수에 따라 웹 서비스 협에 대한 연결을 작성합니다. 예를 들어 소스와 대상이 각각 1개씩 있는 세션에 파티션 3개를 구성하면 통합 서비스가 웹 서비스 협에 대한 연결을 6개(소스용으로 3개와 대상용으로 3개) 작성합니다. 파티션을 사용하면 웹 서비스 요청의 동시 실행이 가능합니다.

다중 파티션으로 구성된 세션을 실행할 때 웹 서비스 협은 소스 연결을 사용하여 통합 서비스에 요청을 전달합니다. 통합 서비스는 대상 연결을 사용하여 웹 서비스 협에 응답을 보냅니다. 웹 서비스 협과 통합 서비스는 라운드-로빈 방식으로 소스 및 대상 연결을 사용합니다.

웹 서비스 매핑에 대해 파티션을 구성하는 경우 웹 서비스 소스 및 대상에 대해 통과 파티션을 구성할 수 있습니다.

## 웹 서비스 워크플로우 문제 해결

웹 서비스 세션에 대해 디버거를 실행하면 세션이 실패하고, 세션을 실행하려면 워크플로우 컨텍스트가 필요하다고 알려주는 오류 메시지가 세션 로그에 표시됩니다.

웹 서비스 세션을 디버그하려는 경우 웹 서비스 워크플로우에 대해 디버거를 실행해야 합니다. 워크플로우가 없는 웹 서비스 매핑 또는 재사용 가능 세션에 대해서는 디버거를 실행할 수 없습니다.

소스 WSDL을 업데이트하고 소스 및 대상 정의를 다시 가져왔습니다. 워크플로우가 올바르지만 서비스 WSDL이 업데이트되지 않았습니다.

매핑에 대한 변경 내용은 웹 서비스 협에서 동적으로 반영되지 않습니다. 매핑 변경 내용을 반영하는 WSDL을 생성하려면 워크플로우를 편집하고 저장해야 합니다. 워크플로우를 저장하면 웹 서비스 협이 서비스에 대해 WSDL을 생성합니다.

웹 서비스 워크플로우가 워크플로우 관리자에서 올바르지만 웹 서비스 협을 시작하면 올바르지 않게 됩니다.

웹 서비스 협을 시작하면 워크플로우 관리자의 유효성 검사 규칙뿐만 아니라 고유의 유효성 규칙에 따라 각 웹 서비스 워크플로우의 유효성이 검사됩니다.

웹 서비스 협은 다음 규칙에 따라 웹 서비스 워크플로우의 유효성을 검사합니다.

- 웹 서비스 소스 정의는 매핑에 2개 이상 있을 수 없습니다.
- 웹 서비스 대상 정의는 매핑에 2개 이상 있을 수 없습니다.
- 웹 서비스 대상 정의가 매핑에 없는 경우 웹 서비스 협은 웹 서비스를 단방향 서비스로 처리합니다.
- 리포지토리 서비스는 웹 서비스 협과 연결되어 있어야 합니다.
- 통합 서비스는 워크플로우와 연결되어 있어야 합니다.

워크플로우 관리자의 유효성 검사 탭에서 웹 서비스 협 오류 메시지를 확인하고 오류 메시지가 나타내는 문제를 수정합니다.

웹 서비스 협에서 워크플로우를 가져오려는 중에 서비스 워크플로우에 지정된 통합 서비스가 없으며 서비스 워크플로우가 올바르지 않음을 나타내는 오류 메시지가 표시됩니다.

웹 서비스 워크플로우를 작성할 때 통합 서비스를 할당해야 합니다. 자세한 내용은 [“웹 서비스 워크플로우 구성” 페이지 96](#)을 참조하십시오.

웹 서비스 협에서 인스턴스를 2개 이상 실행하도록 구성된 웹 서비스 워크플로우에 요청을 보냈습니다. 요청을 보낸 후에 웹 서비스 워크플로우를 중지했습니다. 결과적으로 결함 응답을 받았습니다.

웹 서비스 협은 워크플로우 상태를 주기적으로 확인합니다. 워크플로우가 실행 중이지 않음을 등록하기 전에 워크플로우에 요청을 보내는 경우 결함 응답을 생성합니다. 인스턴스를 2개 이상 실행하도록 워크플로우가 구성된 경우 웹 서비스 협은 다른 워크플로우 인스턴스를 시작합니다. 그러나 웹 서비스 협이 요청을 캐시하지 않기 때문에 새 워크플로우 인스턴스에 요청을 다시 보낼 수 없습니다.

버전 지정 리포지토리에서 실시간 웹 서비스 워크플로우를 변경했습니다. 워크플로우를 실행해도 변경 내용이 적용되지 않았습니다.

버전 지정 리포지토리에서 실시간 웹 서비스 워크플로우를 수정하는 경우 변경 내용을 적용하려면 워크플로우를 체크 인해야 합니다.

예를 들어 다른 PowerCenter 통합 서비스와 연결되도록 실시간 웹 서비스 워크플로우를 수정합니다. 변경 내용을 체크 인하면 웹 서비스 협이 새 통합 서비스를 사용하여 워크플로우를 실행합니다. 변경 내용을 체크 인하지 않으면 웹 서비스 협을 다시 시작할 때까지 웹 서비스 협이 새 통합 서비스를 사용하지 않습니다.

## 부록 A

# 웹 서비스 샘플 클라이언트 응용 프로그램

이 부록에 포함된 항목:

- [웹 서비스 샘플 클라이언트 응용 프로그램 개요, 103](#)
- [일괄 웹 서비스 샘플 프로그램 사용, 103](#)
- [일괄 웹 서비스 예제, 105](#)
- [실시간 웹 서비스 샘플 프로그램 사용, 110](#)
- [실시간 웹 서비스 예제, 113](#)

## 웹 서비스 샘플 클라이언트 응용 프로그램 개요

Informatica에서는 PowerCenter 웹 서비스의 사용 방법을 보여주는 샘플 클라이언트 응용 프로그램을 제공합니다. 예제에는 Java 및 C# 언어로 생성된 프로그램이 포함되어 있습니다. Java 샘플 프로그램은 Axis 웹 서비스 툴킷에서 생성한 프록시 클래스를 사용합니다. C# 샘플 프로그램은 wsdl.exe 도구를 사용하여 .NET 플랫폼용으로 생성된 프록시 클래스를 사용합니다. 샘플 프로그램은 PowerCenter 일괄 웹 서비스 및 실시간 웹 서비스와 함께 작동합니다.

웹 서비스 샘플 프로그램은 다음 디렉터리에 설치되어 있습니다.

```
/<PowerCenterInstallDir>/server/samples/WebServices
```

웹 서비스 샘플 프로그램을 실행하기 전에 PowerCenter 도메인에서 웹 서비스 협을 작성하고 활성화합니다. Administrator 도구를 사용하여 웹 서비스 협을 작성, 구성 및 활성화합니다.

## 일괄 웹 서비스 샘플 프로그램 사용

일괄 웹 서비스 샘플 프로그램을 사용하려면 PowerCenter가 설치되고 실행 중이어야 합니다. PowerCenter 도메인에는 리포지토리 서비스와 연결된 웹 서비스 협이 포함되어 있어야 합니다.

일괄 웹 서비스 샘플 프로그램은 다음 디렉터리에 설치되어 있습니다.

```
/<PowerCenterInstallDir>/server/samples/WebServices
```

다음 테이블에는 /WebServices 디렉터리에 있는 파일과 디렉터리가 설명되어 있습니다.

디렉터리	설명
/lib	샘플 프로그램을 실행하는 데 필요한 라이브러리 파일이 포함되어 있습니다.
/ssl	클라이언트 응용 프로그램을 보안 모드(HTTPS)로 실행하기 위한 샘플 키 저장소가 포함되어 있습니다.
/samples/BatchWebServices/axis/ <SampleProgramDirectory>	Java 샘플 프로그램이 포함되어 있습니다. 각 일괄 웹 서비스 샘플 프로그램의 소스 파일은 개별 디렉터리에서 찾을 수 있습니다. 디렉터리 이름은 샘플 프로그램에서 보여주는 일괄 웹 서비스 작업을 나타냅니다. 예를 들어 /multiservers 디렉터리에 있는 샘플 프로그램은 리포지토리 서비스와 연결된 2개 이상의 통합 서비스에 로그인하는 방법을 보여줍니다. 이 디렉터리에는 샘플 프로그램을 컴파일하고 실행하기 위한 배치 파일과 스크립트 파일도 포함되어 있습니다.
/samples/BatchWebServices/axis/proxyclasses	Java 샘플 프로그램을 위한 프록시 클래스가 포함되어 있습니다.
/samples/BatchWebServices/dotnet/csharp/ <SampleProgramDirectory>	C# 샘플 프로그램이 포함되어 있습니다. 각 일괄 웹 서비스 샘플 프로그램의 소스 파일은 개별 디렉터리에서 찾을 수 있습니다. 디렉터리 이름은 샘플 프로그램에서 보여주는 일괄 웹 서비스 작업을 나타냅니다. 예를 들어 /multiservers 디렉터리에 있는 샘플 프로그램은 리포지토리 서비스와 연결된 2개 이상의 통합 서비스에 로그인하는 방법을 보여줍니다. 각 샘플 프로그램 디렉터리에는 샘플 프로그램을 컴파일하기 위한 배치 파일도 포함되어 있습니다.
/samples/BatchWebServices/dotnet/csharp/ proxyclasses	C# 샘플 프로그램을 위한 프록시 클래스가 포함되어 있습니다. 이 디렉터리에는 프록시 클래스를 컴파일하기 위한 배치 파일도 포함되어 있습니다.

## 일괄 웹 서비스 샘플 프로그램 컴파일

일괄 웹 서비스 샘플 프로그램을 컴파일하는 단계는 프로그래밍 언어를 기반으로 결정됩니다.

### 샘플 Java 프로그램 컴파일

샘플 Java 프로그램을 컴파일하려면 샘플 프로그램 디렉터리로 이동하여 컴파일 배치 파일 또는 스크립트 파일을 실행합니다. 컴파일하려는 샘플 프로그램의 이름과 일치하는 배치 파일 또는 스크립트 파일을 실행합니다.

예를 들어 /axis/multithreaded 디렉터리에 있는 Sample1.java 프로그램을 컴파일하려면 해당 디렉터리로 이동하여 CompileSample1.bat(Windows) 또는 CompileSample1.sh(UNIX)를 실행합니다. 컴파일 프로세스는 동일한 디렉터리에서 샘플 프로그램에 대한 .class 파일을 작성합니다.

### 샘플 C# 프로그램 컴파일

샘플 C# 프로그램을 컴파일하려면 다음 단계를 수행합니다.

1. /dotnet/csharp/proxyclasses 디렉터리로 이동하여 compile.bat를 실행합니다.

컴파일 프로세스는 WebServicesHub.dll이라는 이름의 동적 연결 라이브러리를 다음 디렉터리에 작성합니다.

/dotnet/csharp/bin directory.



2. 샘플 프로그램 디렉터리로 이동하여 컴파일하려는 샘플 프로그램의 이름과 일치하는 컴파일 배치 파일을 실행합니다.

컴파일 프로세스는 컴파일된 프로그램 파일의 이름과 **.exe** 확장자가 있는 실행 파일을 작성합니다.

## 일괄 웹 서비스 샘플 프로그램 실행

클라이언트 응용 프로그램을 실행할 때에는 웹 서비스 헷이 실행 중이어야 합니다.

필수 매개 변수와 함께 샘플 프로그램을 실행합니다. 일괄 웹 서비스 샘플 프로그램을 실행하는 단계는 프로그래밍 언어를 기반으로 결정됩니다.

### 샘플 Java 프로그램 실행

샘플 Java 프로그램을 실행하려면 샘플 프로그램 디렉터리로 이동하여 실행하려는 샘플 프로그램의 이름과 일치하는 배치 파일 또는 스크립트 파일을 실행합니다. 예를 들어 `/axis/multithreaded` 디렉터리에 있는 `Sample1.java` 프로그램을 실행하려면 해당 디렉터리로 이동하여 `RunSample1.bat(Windows)` 또는 `RunSample1.sh(UNIX)`를 실행합니다.

### 샘플 C# 프로그램 실행

샘플 C# 프로그램을 실행하려면 샘플 프로그램 디렉터리로 이동하여 실행하려는 샘플 프로그램에 대한 실행 파일을 실행합니다.

## 일괄 웹 서비스 예제

다음 테이블에는 샘플 프로그램이 포함된 디렉터리가 나열되어 있습니다.

플랫폼	디렉터리
Java	<code>/WebServices/samples/BatchWebServices/axis/&lt;SampleProgramDirectory&gt;</code>
C#	<code>/WebServices/samples/BatchWebServices/dotnet/csharp/&lt;SampleProgramDirectory&gt;</code>

동일한 샘플 프로그램 집합이 **Java** 및 **C#** 언어에 대해 제공됩니다. 각 플랫폼은 동일한 디렉터리를 가지고 있으며 각 디렉터리에는 다양한 웹 서비스 사용법을 보여주는 샘플 프로그램이 포함되어 있습니다. 이 섹션에는 **Java** 및 **C#** 샘플 프로그램이 설명되어 있습니다.

## 찾아보기

`/browsing` 디렉터리에 있는 샘플 프로그램은 리포지토리에서 정보를 가져오는 웹 서비스 작업의 사용 방법을 보여줍니다.

### Sample1.java 및 Sample1.cs

이 샘플 프로그램은 리포지토리에 로그인한 후 메타데이터 웹 서비스 작업을 사용하여 리포지토리의 폴더, 워크플로우 및 태스크에 대한 정보와 리포지토리에 등록된 통합 서비스에 대한 정보를 가져옵니다.

**디렉터리:** `/browsing`

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample1.bat 또는 CompileSample1.sh

**Java 샘플을 실행하는 파일:** RunSample1.bat 또는 RunSample1.sh

**C# 샘플을 실행하는 파일:** Sample1.exe

다음 테이블에는 Sample1 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <code>-ns</code> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.

## Sample2.java 및 Sample2.cs

이 샘플 프로그램은 리포지토리에 로그인하고 연결된 통합 서비스에 연결합니다. 샘플 프로그램은 메타데이터 및 데이터 통합 웹 서비스 작업을 사용하여 리포지토리의 폴더에 액세스하고 폴더에서 찾은 첫 번째 워크플로우를 시작하고 중지합니다.

**디렉터리:** /browsing

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample2.bat 또는 CompileSample2.sh

**Java 샘플을 실행하는 파일:** RunSample2.bat 또는 RunSample2.sh

**C# 샘플을 실행하는 파일:** Sample2.exe

다음 테이블에는 Sample2 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <code>-ns</code> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.

매개 변수	설명
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.
통합 서비스 도메인 이름	통합 서비스가 포함된 도메인의 이름입니다.
통합 서비스 이름	통합 서비스 의 이름입니다.

## 데이터 통합

/dataintegration 디렉터리에 있는 샘플 프로그램은 데이터 통합 웹 서비스에서 사용 가능한 워크플로우 및 태스크 작업의 사용 방법을 보여줍니다.

### Sample1.java 및 Sample1.cs

이 샘플 프로그램은 리포지토리에 로그인하고 연결된 통합 서비스에 연결합니다. 샘플 프로그램은 데이터 통합 웹 서비스 작업을 사용하여 통합 서비스에서 실행되는 워크플로우를 시작하고 중지합니다.

**디렉터리:** /dataintegration

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample1.bat 또는 CompileSample1.sh

**Java 샘플을 실행하는 파일:** RunSample1.bat 또는 RunSample1.sh

**C# 샘플을 실행하는 파일:** Sample1.exe

다음 테이블에는 Sample1 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <code>-ns</code> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.
통합 서비스 도메인 이름	통합 서비스가 포함된 도메인의 이름입니다.
통합 서비스 이름	통합 서비스 의 이름입니다.
폴더 이름	리포지토리의 폴더 이름입니다.
워크플로우 이름	세션이 포함된 워크플로우의 이름입니다.
태스크 이름	시작할 태스크의 이름입니다.

## 다중 통합 서비스

/multiservers 디렉터리에 있는 샘플 프로그램은 리포지토리 서비스와 연결된 2개 이상의 통합 서비스에 로그인하는 방법을 보여줍니다. 동일한 기법을 사용하여 리포지토리 서비스와 연결된 통합 서비스에 개수에 관계없이 동시에 액세스할 수 있습니다.

### Sample1.java 및 Sample1.cs

이 샘플 프로그램은 리포지토리에 로그인하고 리포지토리 및 연결된 2개의 통합 서비스에 연결합니다. 샘플 프로그램은 데이터 통합 웹 서비스 작업을 사용하여 두 통합 서비스 모두에 대한 속성을 가져옵니다.

**참고:** 예제에 표시된 대로 데이터 통합 웹 서비스에 대해 2개의 프록시 개체를 작성하여 2개의 통합 서비스에 로그인해야 합니다. 로그인하려는 통합 서비스 각각에 대해 1개의 프록시 개체를 작성합니다.

**디렉터리:** /multiservers

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample1.bat 또는 CompileSample1.sh

**Java 샘플을 실행하는 파일:** RunSample1.bat 또는 RunSample1.sh

**C# 샘플을 실행하는 파일:** Sample1.exe

다음 테이블에는 Sample1 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <i>-ns</i> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.
통합 서비스 도메인 이름	통합 서비스가 포함된 도메인의 이름입니다.
통합 서비스 이름 1	리포지토리와 연결된 통합 서비스의 이름입니다.
통합 서비스 이름 2	리포지토리와 연결된 두 번째 통합 서비스의 이름입니다.

## 다중 스레딩

/multithreaded 디렉터리에 있는 샘플 프로그램은 다중 스레드로 프록시 개체를 사용하여 병렬로 작업을 수행하는 방법을 보여줍니다. 동일한 기법을 사용하여 클라이언트 응용 프로그램이 작업이 완료되기를 기다리는 동안 다른 작업을 계속 호출하고 실행하도록 설정할 수 있습니다. 예를 들어 클라이언트 응용 프로그램이 스레드에서 WaitTillWorkflowComplete 작업을 호출하는 경우 다른 스레드에서 다른 작업을 계속 수행할 수 있습니다.

## Sample1.java 및 Sample1.cs

이 샘플 프로그램은 리포지토리에 로그인하고 연결된 통합 서비스에 연결합니다. 샘플 프로그램은 스레드 2개를 시작하여 데이터 통합 웹 서비스 프록시 개체를 두 스레드에 모두 전달합니다. 한 스레드에서 샘플 프로그램은 통합 서비스에서 워크플로우를 시작하고 완료될 때까지 대기합니다. 다른 스레드에서는 통합 서비스의 속성을 가져옵니다. 이와 비슷하게 메타데이터 웹 서비스 프록시 개체를 다중 스레드로 사용할 수 있습니다.

**디렉터리:** /multithreaded

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample1.bat 또는 CompileSample1.sh

**Java 샘플을 실행하는 파일:** RunSample1.bat 또는 RunSample1.sh

**C# 샘플을 실행하는 파일:** Sample1.exe

다음 테이블에는 Sample1 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <code>-ns</code> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.
통합 서비스 도메인 이름	통합 서비스가 포함된 도메인의 이름입니다.
통합 서비스 이름	통합 서비스의 이름입니다.
폴더 이름	워크플로우가 포함된 리포지토리의 폴더 이름입니다.
워크플로우 이름	리포지토리의 워크플로우 이름입니다.

## 웹 서비스 협 테스트

/testsamples 디렉터리에 있는 샘플 프로그램은 PowerCenter 도메인에서 올바른 웹 서비스 협이 실행 중인지 확인하는 방법을 보여줍니다.

## Sample1.java 및 Sample1.cs

이 샘플 프로그램은 리포지토리에 로그인하고 연결된 통합 서비스에 연결합니다. 샘플 프로그램은 메타데이터 및 데이터 통합 웹 서비스 작업을 사용하여 리포지토리 서비스와 통합 서비스에 대한 정보를 가져옵니다.

**디렉터리:** /testsamples

**Java 및 C# 샘플을 컴파일하는 파일:** CompileSample1.bat 또는 CompileSample1.sh

**Java 샘플을 실행하는 파일:** RunSample1.bat 또는 RunSample1.sh

### C# 샘플을 실행하는 파일: Sample1.exe

다음 테이블에는 Sample1 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
보안 모드	응용 프로그램을 실행하는 보안 모드를 나타냅니다. 비보안 모드(HTTP)로 응용 프로그램을 실행하려면 <code>-ns</code> 인수를 전달합니다. 예제에서는 보안 모드(HTTPS)를 지원하지 않습니다.
호스트 이름	웹 서비스 협이 실행 중인 시스템의 이름 또는 IP 주소입니다.
포트 번호	웹 서비스 협이 실행 중인 포트 번호입니다.
리포지토리 도메인 이름	리포지토리 서비스가 포함된 도메인의 이름입니다.
리포지토리 이름	리포지토리 서비스의 이름입니다.
사용자 이름	리포지토리에 로그인할 사용자 이름입니다.
암호	리포지토리에 로그인할 사용자 이름의 암호입니다.
통합 서비스 도메인 이름	통합 서비스가 포함된 도메인의 이름입니다.
통합 서비스 이름	통합 서비스 의 이름입니다.

## 실시간 웹 서비스 샘플 프로그램 사용

실시간 웹 서비스 샘플 프로그램을 사용하려면 PowerCenter가 설치되고 실행 중이어야 합니다. PowerCenter 도메인에는 리포지토리 서비스와 연결된 웹 서비스 협이 포함되어 있어야 합니다.

실시간 웹 서비스 샘플 프로그램은 다음 디렉터리에 설치되어 있습니다.

```
/<PowerCenterInstallDir>/server/samples/WebServices
```

실시간 웹 서비스 예제에는 조회 테이블을 작성하는 파일과 샘플 프로그램에서 사용할 웹 서비스 워크플로우가 포함되어 있습니다.

다음 테이블에는 /RealTimeWebServices 디렉터리에 있는 파일과 디렉터리가 나열되어 있습니다.

디렉터리	설명
/samples/RealTimeWebServices/ImportXML	실시간 웹 서비스 샘플 프로그램에서 호출하는 웹 서비스 워크플로우가 포함되어 있습니다. 샘플 프로그램을 사용하려면 XML 파일을 리포지토리로 가져오고 웹 서비스 워크플로우에서 SQL 및 조회 변환을 위한 데이터베이스 연결을 설정합니다.
/lib	샘플 프로그램을 실행하는 데 필요한 라이브러리 파일이 포함되어 있습니다.
/samples/RealTimeWebServices/SQLScripts /SINGLEROWLOOKUP	단일 행 조회를 위해 샘플 프로그램에서 사용되는 조회 테이블을 작성하기 위한 SQL 스크립트가 포함되어 있습니다. SQL 스크립트를 실행하면 선택한 데이터베이스에 테이블이 작성됩니다.

디렉터리	설명
/samples/RealTimeWebServices/ SQLScripts /MULTIPLEROWLOOKUP	다중 행 조회를 위해 샘플 프로그램에서 사용되는 조회 테이블을 작성하기 위한 SQL 스크립트가 포함되어 있습니다. SQL 스크립트를 실행하면 선택한 데이터베이스에 테이블이 작성됩니다.
/samples/RealTimeWebServices/ Unprotected WebServices/axis/ <SampleProgramDirectory>	Java 샘플 프로그램이 포함되어 있습니다. 각 실시간 웹 서비스 샘플 프로그램의 소스 파일은 개별 디렉터리에서 찾을 수 있습니다. 각 디렉터리에는 샘플 프로그램을 컴파일하고 실행하기 위한 배치 파일 및 스크립트 파일과 샘플 프로그램에서 사용하는 프록시 클래스를 저장하기 위한 하위 폴더가 포함되어 있습니다.

실시간 웹 서비스 예제를 사용하려면 다음 단계를 수행해야 합니다.

1. 샘플 프로그램에서 조회 테이블로 사용할 데이터베이스 테이블을 작성합니다.
2. 웹 서비스 협과 연결된 리포지토리로 매핑과 웹 서비스 워크플로우를 가져옵니다.
3. m\_CustomerLookup\_MULTIPLEROW 매핑에서 SQL 변환을 위해 데이터베이스와 데이터 유형을 수정합니다.
4. 샘플 워크플로우의 데이터베이스 연결 설정을 지정합니다.
5. 실시간 웹 서비스 샘플 프로그램을 컴파일합니다.
6. 실시간 웹 서비스 샘플 프로그램을 실행합니다.

## 1단계. 조회 테이블 작성

일괄 웹 서비스 샘플 프로그램과 함께 제공되는 SQL 스크립트 파일을 사용하여 관계형 데이터베이스에 조회 테이블을 작성합니다. 다음 데이터베이스에 조회 테이블을 작성할 수 있습니다.

- IBM DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

**참고:** Teradata에서 조회 테이블을 작성하는 경우 데이터베이스 서버의 기본 모드를 ANSI로 설정해야 합니다.

다음 테이블에는 SQL 스크립트가 설명되어 있습니다.

스크립트 파일 이름	설명
CustomerLookup_SINGLEROW_<데이터베이스>.sql	샘플 프로그램에서 단일 행 조회에 사용할 SINGLEROWLOOKUP이라는 고객 테이블을 작성합니다.
CustomerLookup_MULTIPLEROW_<데이터베이스>.sql	샘플 프로그램에서 다중 행 조회에 사용할 MULTIPLEROWLOOKUP이라는 고객 테이블을 작성합니다.

**참고:** 데이터베이스 연결 설정. 샘플 워크플로우를 리포지토리로 가져온 후 변환을 위한 워크플로우의 데이터베이스 연결 설정을 현재 데이터베이스 설정과 일치하도록 수정해야 합니다.

## 2단계. 매핑 및 워크플로우 가져오기

실시간 웹 서비스 샘플 프로그램은 샘플 웹 서비스 워크플로우를 실행합니다. 샘플 프로그램을 사용하려면 웹 서비스 협과 연결된 리포지토리로 샘플 매핑 및 워크플로우를 가져옵니다.

다음 테이블에는 XML 파일이 설명되어 있습니다.

스크립트 파일 이름	설명
wf_CustomerLookup_SINGLEROW.XML	샘플 프로그램에서 단일 행 조회에 사용할 조회 변환이 있는 웹 서비스 워크플로우가 포함되어 있습니다.
wf_CustomerLookup_MULTIPLEROW.XML	샘플 프로그램에서 다중 행 조회에 사용할 SQL 변환이 있는 웹 서비스 워크플로우가 포함되어 있습니다.

## 3단계. SQL 변환을 위해 데이터베이스 및 데이터 유형 수정

다중 행 조회를 보여주는 이 웹 서비스 예제는 SQL 변환을 사용합니다. 사용하는 데이터베이스에서 SQL 변환의 포트에 사용 가능한 원시 데이터 유형을 결정합니다. 적절한 데이터베이스를 사용하도록 SQL 변환을 구성하고 적절한 원시 데이터 유형을 사용하도록 포트를 설정해야 합니다.

SQL 변환을 위해 데이터베이스 및 데이터 유형을 수정하려면 다음 단계를 수행합니다.

1. PowerCenter Designer에서 **m\_CustomerLookup\_MULTIPLEROW** 매핑을 매핑 디자이너에서 열고 **sql\_Customer** 변환 인스턴스를 편집합니다.
2. 변환 편집 창에서 **SQL** 설정 탭으로 이동하여 데이터베이스 유형 특성의 값을 예제에 사용 중인 데이터베이스로 설정합니다.
3. **SQL** 포트 탭으로 이동하여 포트의 데이터 유형이 올바른 원시 데이터 유형에 매핑되어 있는지 확인합니다.  
대부분의 데이터베이스에서 기본 원시 데이터 유형 매핑은 올바릅니다. **Microsoft SQL Server** 및 **Sybase**의 경우 문자열 데이터 유형을 **varchar** 원시 데이터 유형으로 매핑합니다.
4. **m\_CustomerLookup\_MULTIPLEROW** 매핑의 변경 내용을 저장합니다.  
매핑을 변경한 후 매핑을 실행하는 워크플로우를 새로 고칩니다.
5. PowerCenter 워크플로우 관리자에서, 매핑을 실행하는 워크플로우를 열고 매핑을 새로 고칩니다.

## 4단계. 데이터베이스 연결 설정 수정

가져온 워크플로우의 SQL 및 조회 변환은 1단계에서 작성한 샘플 조회 테이블에 연결할 수 있어야 합니다.

가져오기 프로세스는 샘플 워크플로우에서 변환을 위한 연결 개체를 가져오지 않습니다. 연결 개체를 작성하여 세션에 사용해야 합니다.

변환을 위한 연결 설정을 업데이트하려면 다음 단계를 수행합니다.

1. PowerCenter 워크플로우 관리자에서, 샘플 테이블에 연결할 연결 개체를 작성합니다.
2. **s\_m\_CustomerLookup\_SINGLEROW** 세션을 편집하고 **lcp\_Customer** 변환의 관계형 연결 정보를 업데이트합니다.  
관계형 연결을 새 연결 개체의 이름으로 설정합니다. 새 설정이 적용된 세션을 저장합니다.
3. **s\_m\_CustomerLookup\_MULTIPLEROW** 세션을 편집하고 **sql\_Customer** 변환의 관계형 연결 정보를 업데이트합니다.  
관계형 연결을 해당 연결 개체의 이름으로 설정합니다. 새 설정이 적용된 세션을 저장합니다.



## 5단계. 실시간 웹 서비스 샘플 프로그램 컴파일

샘플 Java 프로그램을 컴파일하려면 샘플 프로그램 디렉터리로 이동하여 컴파일 배치 파일 또는 스크립트 파일을 실행합니다. 컴파일하려는 샘플 프로그램의 이름과 일치하는 배치 파일 또는 스크립트 파일을 실행합니다.

예를 들어 /axis/CustomLookup\_SINGLEROW 디렉터리에 있는 Sample.java 프로그램을 컴파일하려면 해당 디렉터리로 이동하여 CompileSample.bat(Windows) 또는 CompileSample.sh(UNIX)를 실행합니다.

컴파일 프로세스는 동일한 디렉터리에서 샘플 프로그램에 대한 .class 파일을 작성합니다.

## 6단계. 실시간 웹 서비스 샘플 프로그램 실행

샘플 프로그램을 실행하는 시스템에 Java 버전 1.5.0\_11-b03이 설치되어 있어야 합니다. 샘플 프로그램을 실행할 때에는 웹 서비스 헷이 실행 중이어야 합니다.

샘플 Java 프로그램을 실행하려면 샘플 프로그램 디렉터리로 이동하여 실행하려는 샘플 프로그램에 대한 배치 파일 또는 스크립트 파일을 실행합니다. 예를 들어 /axis/CustomLookup\_MULTIPLEROW 디렉터리에 있는 Sample.java 프로그램을 실행하려면 해당 디렉터리로 이동하여 RunSample.bat(Windows) 또는 RunSample.sh(UNIX)를 실행합니다.

필수 매개 변수와 함께 샘플 프로그램을 실행합니다.

# 실시간 웹 서비스 예제

이 섹션에는 실시간 웹 서비스용 샘플 프로그램이 설명되어 있습니다. 각 디렉터리에는 실시간 웹 서비스의 다양한 사용 방법을 보여주는 샘플 프로그램이 포함되어 있습니다.

## 다중 행 조회

/CustomerLookup\_MULTIPLEROW 디렉터리의 샘플 프로그램은 클라이언트 응용 프로그램이 웹 서비스 워크플로우를 실행하여 조회를 수행하고 다중 데이터 행으로 응답을 처리하는 방법을 보여줍니다.

### Sample.java

이 샘플 프로그램은 데이터베이스에서 고객 ID를 조회하여 고객 정보를 출력하는 PowerCenter 웹 서비스 워크플로우를 호출합니다. 워크플로우는 SQL 변환을 사용하여 데이터베이스에서 다중 행을 검색합니다.

**디렉터리:** /CustomerLookup\_MULTIPLEROW

**Java 샘플을 컴파일하는 파일:** CompileSample.bat 또는 CompileSample.sh

**Java 샘플을 실행하는 파일:** RunSample.bat 또는 RunSample.sh

다음 테이블에는 **Sample** 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
고객 ID	조회할 고객의 ID입니다. 고객 ID는 정수로 전달합니다.
EndPoint URL	웹 서비스를 찾을 수 있는 URL입니다. 끝점 URL은 문자열로 전달합니다. 실시간 웹 서비스에 대한 끝점 URL은 웹 서비스 WSDL에 있는 service 요소의 soap:address 위치 요소에서 찾을 수 있습니다. 샘플 웹 서비스의 기본 끝점 URL은 <code>http://&lt;WSHostName&gt;:&lt;WSHPort&gt;/wsh/services/ts/CustomerLookup_MULTIPLEROW</code> 입니다. 웹 서비스 협이 HTTPS에서 실행 중인 경우 끝점 URL은 HTTPS로 시작합니다.

## 단일 행 조회

`/CustomerLookup_SINGLEROW` 디렉터리의 샘플 프로그램은 클라이언트 응용 프로그램이 웹 서비스 워크플로우를 호출하여 조회를 수행하고 단일 데이터 행으로 응답을 처리하는 방법을 보여줍니다.

### Sample.java

이 샘플 프로그램은 데이터베이스에서 고객 ID를 조회하여 고객 정보를 출력하는 **PowerCenter** 웹 서비스 워크플로우를 호출합니다. 매핑은 조회 변환을 사용하여 데이터베이스에서 한 행을 검색합니다.

**디렉터리:** `/CustomerLookup_SINGLEROW`

**Java 샘플을 컴파일하는 파일:** `CompileSample.bat` 또는 `CompileSample.sh`

**Java 샘플을 실행하는 파일:** `RunSample.bat` 또는 `RunSample.sh`

다음 테이블에는 **Sample** 응용 프로그램을 실행하는 데 사용하는 매개 변수가 설명되어 있습니다.

매개 변수	설명
고객 ID	조회할 고객의 ID입니다. 고객 ID는 정수로 전달합니다.
EndPoint URL	웹 서비스를 찾을 수 있는 URL입니다. 끝점 URL은 문자열로 전달합니다. 실시간 웹 서비스에 대한 끝점 URL은 웹 서비스 WSDL에 있는 service 요소의 soap:address 위치 요소에서 찾을 수 있습니다. 샘플 웹 서비스의 기본 끝점 URL은 <code>http://&lt;WSHostName&gt;:&lt;WSHPort&gt;/wsh/services/ts/CustomerLookup_SINGLEROW</code> 입니다. 웹 서비스 협이 HTTPS에서 실행 중인 경우 끝점 URL은 HTTPS로 시작합니다.

## 부록 B

# 웹 브라우저 구성

이 부록에 포함된 항목:

- [웹 브라우저 구성, 115](#)

## 웹 브라우저 구성

Informatica 플랫폼에서 Microsoft Internet Explorer 또는 Google Chrome을 사용하여 웹 서비스 헵 콘솔을 실행할 수 있습니다.

웹 서비스 헵 콘솔을 실행하려면 브라우저에서 다음 옵션을 구성합니다.

### 스크립팅 및 ActiveX

Microsoft Internet Explorer에서 다음 컨트롤을 활성화합니다.

- 활성 스크립팅
- 프로그래밍 방식 클립보드 액세스 허용
- ActiveX 컨트롤 및 플러그 인 실행
- 스크립팅에 안전한 것으로 표시된 ActiveX 컨트롤 스크립팅

컨트롤을 구성하려면 **도구 > 인터넷 옵션 > 보안 > 사용자 지정 수준**을 클릭합니다.

### 신뢰할 수 있는 사이트

Informatica 도메인이 Kerberos 인증을 사용하는 네트워크에서 실행되는 경우 Informatica 웹 응용 프로그램에 대한 액세스를 허용하도록 브라우저를 구성해야 합니다. Microsoft Internet Explorer, Microsoft Edge 및 Google Chrome에서 Informatica 웹 응용 프로그램의 URL을 신뢰할 수 있는 사이트의 목록에 추가합니다. Safari에서 Informatica 웹 응용 프로그램의 인증서를 키체인에 추가합니다. Windows에서 Chrome 버전 86.0.42x 이상을 사용 중인 경우 AuthServerWhitelist 및 AuthNegotiateDelegateWhitelist 정책도 설정해야 합니다.

# 인덱스

## B

표시  
웹 서비스 워크플로우 속성 [96](#)  
플랫 파일  
첨부 파일이 포함된 매핑 [93](#)  
해시된 암호  
예제 [71](#)  
웹 서비스 보안 [70](#), [71](#)  
협당 최대 실행 횟수  
웹 서비스 워크플로우 속성 [96](#)

## D

`deinitializeDlServerConnection`  
더 이상 사용되지 않는 작업 [39](#)

## G

`getNextLogSegment`  
더 이상 사용되지 않는 작업 [39](#)

## I

`initializeDlServerConnection`  
더 이상 사용되지 않는 작업 [39](#)

## N

단방향 매핑  
설명 [88](#)  
더 이상 사용되지 않는 작업  
`deinitializeDlServerConnection` [39](#)  
`getNextLogSegment` [33](#)  
`initializeDlServerConnection` [39](#)  
`resumeWorkflow` [43](#)  
`startSessionLogFetch` [45](#)  
`startWorkflowLogFetch` [49](#)  
데이터 통합 웹 서비스  
설명 [16](#)  
동시 실행  
웹 서비스 인스턴스 시작 [98](#)  
웹 서비스 인스턴스 종료 [98](#)  
매핑  
단방향 [88](#)  
요청-응답 [88](#)  
웹 서비스 매핑 유형 [88](#)  
준비됨 [89](#)  
첨부 파일이 포함된 WSDL [93](#)  
첨부 파일이 포함된 XML [93](#)  
첨부 파일이 포함된 플랫 또는 XML [93](#)

메시지 포트  
구성 [78](#)  
메타데이터 웹 서비스  
설명 [16](#)  
보호됨  
웹 서비스 워크플로우 속성 [96](#)  
사용자 이름 토큰  
웹 서비스 보안 [69](#)  
임시 값 [72](#)  
작성 날짜 요소 [72](#)  
사용자 자격 증명  
웹 서비스 보안 [69](#)  
샘플 프로그램  
실시간 웹 서비스 [110](#)  
일괄 웹 서비스 [103](#)  
서비스 시간 임계값  
웹 서비스 워크플로우 속성 [96](#)  
서비스 이름  
웹 서비스 워크플로우 속성 [96](#)  
실시간 웹 서비스  
샘플 프로그램 [110](#)  
설명 [16](#)  
실행  
일괄 웹 서비스 샘플 프로그램 [105](#)  
실행 가능  
웹 서비스 워크플로우 속성 [96](#)  
암호  
요약 [70](#), [72](#)  
일반 텍스트 [70](#)  
해시됨 [70](#), [71](#)  
요약 암호  
예제 [73](#)  
웹 서비스 보안 [70](#), [72](#)  
요청-응답 매핑  
SQL 변환 사용 [88](#)  
설명 [88](#)  
워크플로우  
웹 서비스 [96](#)  
웹 서비스  
데이터 통합 [16](#)  
매핑 유형 [88](#)  
메타데이터 [16](#)  
실시간 [16](#)  
일괄 [16](#)  
웹 서비스 공급자  
설명 [15](#)  
아키텍처 [16](#)  
웹 서비스 공급자 기록기  
구성 [100](#)  
웹 서비스 공급자 판독기  
구성 [99](#)  
웹 서비스 대상  
메시지 포트 구성 [78](#)  
웹 서비스 보안  
OASIS 표준 [69](#)

- 웹 서비스 보안 (계속)
  - 사용자 이름 토큰 [69](#)
  - 사용자 자격 증명 [69](#)
- 웹 서비스 소스
  - 메시지 포트 구성 [78](#)
- 웹 서비스 워크플로우
  - 구성 [96](#)
  - 문제 해결 [101](#)
  - 작성 [96](#)
- 웹 서비스 워크플로우 속성
  - 보호됨 [96](#)
  - 서비스 시간 임계값 [96](#)
  - 서비스 이름 [96](#)
  - 실행 가능 [96](#)
  - 웹 서비스 협 [96](#)
  - 제한 시간 [96](#)
  - 표시 [96](#)
  - 협당 최대 실행 횟수 [96](#)
- 웹 서비스 인스턴스
  - 시작 [98](#)
  - 종료 [98](#)
- 웹 서비스 협
  - 설명 [15](#)
- 웹 서비스 워크플로우 속성 [96](#)
- 일괄 웹 서비스
  - 샘플 프로그램 [103](#)
  - 샘플 프로그램 실행 [105](#)
  - 샘플 프로그램 컴파일 [104](#)
  - 설명 [16](#)
- 일반 텍스트 암호
  - 웹 서비스 보안 [70](#)
- 임시 값
  - 사용자 이름 토큰 보안 [72](#)
- 작성 날짜 요소
  - 사용자 이름 토큰 보안 [72](#)
- 제한 시간
  - 웹 서비스 워크플로우 속성 [96](#)
- 준비된 매핑
  - 설명 [89](#)
- 첨부 파일
  - SOAP 메시지 [95](#)
  - WSDL [93](#)
  - XML 매핑 [93](#)
  - 플랫 파일 매핑 [93](#)
- 컴파일
  - 일괄 웹 서비스 샘플 프로그램 [104](#)

- 파티션
  - 웹 서비스 세션 [101](#)
- 판독기
  - 웹 서비스 세션 구성 [99](#)

## O

- OASIS
  - 웹 서비스 보안 표준 [69](#)

## R

- resumeWorkflow
  - 더 이상 사용되지 않는 작업 [43](#)

## S

- SOAP
  - 첨부 파일 [95](#)
- SQL 변환
  - 요청-응답 매핑 [88](#)
- startSessionLogFetch
  - 더 이상 사용되지 않는 작업 [45](#)
- startWorkflowLogFetch
  - 더 이상 사용되지 않는 작업 [49](#)

## U

- UsernameToken 요소
  - 웹 서비스 보안 [70](#)

## ┐

- 구성
  - 웹 서비스 공급자 기록기 [100](#)
  - 웹 서비스 공급자 판독기 [99](#)
  - 웹 서비스 워크플로우 [96](#)
- 기록기
  - 웹 서비스 세션 구성 [100](#)