

TIBCO iProcess[®] Conductor

AOPD Message-Driven API

Developer's Guide

Software Release 11.2.1
August 2012

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, The Power of Now, TIBCO Adapter and TIBCO iProcess, are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2004-2012 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Preface	v
Changes from the Previous Release of This Guide	vi
Related Documentation	vii
TIBCO iProcess Conductor Documentation	vii
Other TIBCO Product Documentation	vii
Third-party Documentation	viii
Typographical Conventions	ix
Connecting with TIBCO Resources	xii
How to Join TIBCOCommunity	xii
How to Access TIBCO Documentation	xii
How to Contact TIBCO Support	xii
Chapter 1 Introduction	1
Product Overview	2
Pre-requisites	2
Supplied Files	5
XML Schema	5
Chapter 2 Development	7
Development Sequence	8
Placing an Order	8
Creating Plan Fragments	9
Creating an AOPD Request	11
Submitting and Processing the AOPD Request	12
Orchestrating the Plan	13
Amending an Order	14
Further Techniques	18
Initial Context	18
Sending a Request to the AOPD Request Queue	18
Receiving a Response from the AOPD Response Queue	20
Getting References	23
Constructing a Primary Key	25
Appendix A Constructing an AOPD Request	27
AOPD Request	28

Elements of the AOPD Request 29

Appendix B The AOPD Response 31

AOPD Response..... 32

Elements of the AOPD Response..... 33

Appendix C Dependencies 35

Types of Dependency 36

Must Start On Dependencies 37

Start No Earlier Than Dependencies 39

Point Dependencies 41

Appendix D Error Codes 43

Error Codes..... 44

Warning Codes 46

Appendix E XML Examples 47

Example Request XML 48

Example Request Cancellation XML..... 50

Example Response XML..... 52

Example Cancellation Response XML 70

Appendix F Sample JNDI Properties File 91

JNDI Properties File 92

Index 93

Preface

This manual provides developers with guidance on the general concepts and usage of the TIBCO iProcess Conductor AOPD Message Driven APIs. These are used to produce a program that can perform automatic execution plan development (AOPD) using TIBCO iProcess Conductor.

Topics

- [Changes from the Previous Release of This Guide, page vi](#)
- [Related Documentation, page vii](#)
- [Typographical Conventions, page ix](#)
- [Connecting with TIBCO Resources, page xii](#)

Changes from the Previous Release of This Guide

This section itemizes the major changes from the previous release of this guide.

There are no changes from the previous release of this guide.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO iProcess Conductor Documentation

The following documents form the TIBCO iProcess Conductor documentation set:

- *TIBCO iProcess Conductor Concepts* Read this manual to gain an understanding of the product that you can apply to the various tasks you may undertake.
- *TIBCO iProcess Conductor Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO iProcess Conductor Implementation* Read this guide for instructions on how to design, plan, and implement the fulfillment of orders.
- *TIBCO iProcess Conductor User's Guide* Read this guide for instructions on using the TIBCO iProcess Conductor user interface to orchestrate execution plans, create process components, amend orders, and so on.
- *TIBCO iProcess Conductor Administrator's Guide* Read this guide for instructions on common administrative tasks, such as archiving completed execution plans, managing users, and deploying the TIBCO iProcess Decisions rule sets.
- *TIBCO iProcess Conductor Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- *TIBCO iProcess Conductor Utility Framework User's Guide* Read this guide for details of the Utility Framework and the applications used to export and import plans and process components.
- *TIBCO iProcess Conductor AOPD Message-Driven API Developer's Guide* Read this guide for details of the TIBCO iProcess Conductor's execution plan interfaces, and the facilities for automatic execution plan development.

Other TIBCO Product Documentation

You may find it useful to refer to the documentation set for TIBCO iProcess[®] Suite, which is used with TIBCO iProcess Conductor.

Third-party Documentation

You may find it useful to read the documentation for the following third-party products:

- Oracle® Database
- Oracle® WebLogic Server
- JBoss® Application Server

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>iProcessConductor</i> <i>Dir</i>	The directory where you install TIBCO iProcess Conductor. For example: - on Windows systems, <code>c:\Program Files\iProcessConductor</code>. - on UNIX systems, <code>/opt/comDomain/iProcessConductor</code>.
<i>iProcessConductor</i> <i>Domain</i>	The target directory where you install the TIBCO iProcess Conductor domain. For example: - on Windows systems, <code>BEA_HOME\user_projects\domains\iPCDomain</code>; or <code>JBOSS_HOME\server\server_type\deploy</code> under JBoss. - on UNIX systems, <code>BEA_HOME/user_projects/domains/iPCDomain</code>. or <code>JBOSS_HOME/server/server_type/deploy</code> under JBoss.
<i>ORACLE_HOME</i>	The pathname to your Oracle home. See your database administrator for details. If you are installing iProcess Conductor to work with a local database, this pathname must point to the Oracle database server. If you are installing iProcess Conductor to work with a remote database, this pathname must point to an Oracle client.
<i>JAVA_HOME</i>	The pathname to the directory where you install Java. For example: - on Windows systems, <code>C:\Program Files\Java\jdk1.6.0_21</code>. - on UNIX systems, <code>/opt/Java/jdk1.6.0_21</code>. Note: When you install the product on AIX platforms and need to communicate with iProcess Engine: — set <i>JAVA_HOME</i> as the pathname to the directory of Java incorporated by iProcess Engine 11.3.1 if you want to communicate with iProcess Engine 11.3.1 — install JRE 1.6.0 IBM J9 2.4 AIX SR6, and set <i>JAVA_HOME</i> as the pathname to the installation directory of JRE 1.6.0 IBM J9 2.4 AIX SR6 if you want to communicate with iProcess Engine versions earlier than 11.3.1
If you are using JBoss:	
<i>JBOSS_HOME</i>	The pathname to the home directory of your JBoss installation. For example: - on Windows systems, <code>c:\jboss\jboss-4.2.1</code>. - on UNIX systems, <code>/opt/JBoss/JBoss-4.2.1</code>.

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
If you are using Oracle WebLogic:	
BEA_HOME	The pathname to the home directory of your Oracle WebLogic installation. For example: - on Windows systems, c:\bea. - on UNIX systems, /opt/bea. Note: Oracle WebLogic was previously known as BEA WebLogic. The existing variable name has been retained in the iProcess Conductor documentation.
WL_HOME	The directory where you install Oracle WebLogic. For example: - on Windows systems, c:\bea\wlserver_10.3. - on UNIX systems, /opt/bea/wlserver_10.3.
Other conventions:	
code font	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use MyCommand to start the foo process.
bold code font	Bold code font is used in the following ways: In procedures, to indicate what a user types. For example: Type admin. In large code samples, to indicate the parts of the sample that are of particular interest. In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]
italic font	Italic font is used in the following ways: To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>. To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal. To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i>
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
[]	An optional item in a command or code syntax. For example: MyCommand [optional_parameter] required_parameter
	A logical OR that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: MyCommand para1 param2 param3
{ }	A logical group of items in a command. Other syntax notations may appear within each logical group. For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4. MyCommand {param1 param2} {param3 param4} In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4: MyCommand {param1 param2} {param3 param4} In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4. MyCommand param1 [param2] {param3 param4}

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

<http://docs.tibco.com>

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 **Introduction**

This chapter introduces the TIBCO iProcess Conductor AOPD Message-Driven API, and the idea of automatic execution plan development.

Topics

- [Product Overview, page 2](#)
- [Supplied Files, page 5](#)

Product Overview

TIBCO iProcess Conductor handles complex business processes by creating and running execution plans. iProcess Conductor enables you to create execution plans from re-usable iProcess Engine process components, sub-goals and deadlines, and to execute them. See *TIBCO iProcess Conductor Concepts* and the rest of the TIBCO iProcess Conductor documentation set for further information on execution plans.

Automatic execution plan development (AOPD) enables you to automate the process of creating these execution plans. AOPD enables you to create dynamic execution plans using *plan fragments* - which are execution plan templates for parts of the overall plan - as building blocks.

This document provides developers with information on producing applications, plug-ins, iProcess Engine procedures or BusinessWorks processes that can interact with iProcess Conductor and perform AOPD using the APIs supplied.



What this document does not discuss is the use of external applications such as BusinessWorks or BusinessEvents to define the processes and rules to perform AOPD. For this, see the documentation provided for these applications.

Pre-requisites

The following requirements must be met for successful automatic execution plan development.

Skills Required

Developers intending to perform automatic execution plan development should have the following skills, which are assumed in this document:

- A thorough understanding of XML Schema Definition (XSD)
- Good knowledge of Java™ Architecture for XML Binding (JAXB)
- Development experience in a Java 2 Platform Enterprise Edition (J2EE) environment
- Good understanding of the concepts of TIBCO iProcess Conductor, and familiarity with its user documentation (see [TIBCO iProcess Conductor Documentation](#)).

Development Environment

Before you start automatic execution plan development, ensure you have the following software available:

- A fully configured iProcess Conductor environment. See the iProcess Conductor documentation set for full details of setting this up.
- The execution environment of the automatic execution plan development program should contain the following files in the classpath:

- \$FRAMEWKS/Jax-qname-1.4.jar
- \$FRAMEWKS/Jaxb-api-1.0.2.jar
- \$FRAMEWKS/Jaxb-impl-1.0.3.jar
- \$FRAMEWKS/Jaxb-libs-1.0.2.jar
- \$FRAMEWKS/Jaxp-api-1.3.jar
- \$FRAMEWKS/RelaxngDatatype-1.3.jar
- \$FRAMEWKS/Xalan-1.2.jar
- \$FRAMEWKS/XJC-err-2.0.jar
- \$FRAMEWKS/Xercesimpl-1.2.jar
- \$FRAMEWKS/XJC-2.0.jar
- \$FRAMEWKS/Namespace-1.4.jar
- \$FRAMEWKS/Xsdlb-1.4.jar
- \$FRAMEWKS/COMMON-util-2.0.jar
- \$FRAMEWKS/SCHEDULER-util-2.0.jar
- \$APP/FULFILMENT-util-2.0.jar

These files are available in the following locations, depending on the target application server:

JBoss Application Server		WebLogic Application Server
\$FRAMEWKS	<i>JBOSS_HOME/server/default/lib</i>	If you are using WebLogic 8.1: <i>JBOSS_HOME/user_projects/domains / iProcessConductorDomain/ appframeworks</i>
		If you are using WebLogic 9.2: <i>BEA_HOME/user_projects/domains/ iProcessConductorDomain/ lib</i>

	JBoss Application Server	WebLogic Application Server
\$APP	<i>JBOSS_HOME</i> /server/default/deploy/COM-ORCH-2.0.ear/APP-INF/lib	<i>BEA_HOME</i> /user_projects/domains/ <i>iProcessConductorDomain</i> /applications/ COM-ORCH-2.0.ear/APP-INF/lib
Additional jar files required in the classpath	<i>JBOSS_HOME</i> /client/jbossall-client.jar	<i>WL_HOME</i> /server/lib/weblogic.jar

- In order to communicate with the iProcess Conductor Orchestrator and OrderManager session beans, the following files are also needed:
 - ORDER-ejb-2.0-client.jar
 - ORCH-ejb-2.0-client.jar
 - OPD-ejb-2.0-client.jar
 - CONFIG-ejb-2.0.jar
 - METADATA-ejb-2.0.jar

These files are available in the *iProcessConductorDir*\COM-ORCH-2.0.ear\ORCH-web-2.0.war\WEB-INF\lib folder of your iProcess Conductor installation.
- Create a user that belongs to the COMUsers group. A user from this group will have the necessary credentials to lookup iProcess Conductor Session Facades. See *TIBCO iProcess Conductor: Administrator's Guide* for further information on creating users and on the COMUsers group.

Supplied Files

The following file is supplied with this release:

- iProcess Conductor XML Schema

XML Schema

The `ipc_schema.zip` file is provided with this release of TIBCO iProcess Conductor. You can find it in the documents folder of the iProcess Conductor distribution media. The zip file contains the iProcess Conductor XML Schema.

Copy this file to a folder on your hard disk, and extract the files contained in it. It will automatically create a folder called `ipc_schema` beneath the folder containing the zip file, and will extract the schema files to it.

Chapter 2 **Development**

This chapter describes the process of program development for automatic execution plan development (AOPD). It explains the AOPD process with the help of an example. Because the AOPD process is so flexible, each stage of the description also indicates the consequences of some alternative decisions, to help your understanding of how AOPD deals with different circumstances.

Topics

- [Development Sequence, page 8](#)
- [Further Techniques, page 18](#)

Development Sequence

An important feature of the AOPD process is that it enables you to create your own workflow to suit the particular needs of each unique order, rather than having to follow a set of rigid procedures.



Although the example has been explained below in a series of steps, it is important to note that with AOPD there are no fixed or pre-defined steps. There can be as many or as few steps in order management as are required by your particular and unique workflow.

When designing a program to perform automatic execution plan development using TIBCO iProcess Conductor, the following sequence of steps is typical:

1. Create and submit an order. See [Placing an Order on page 8](#).
2. Create one or more plan fragments. See [Creating Plan Fragments on page 9](#).
3. Create an AOPD request, either manually using iProcess Conductor or using an external system. See [Creating an AOPD Request on page 11](#).
4. Submit the AOPD request to the request queue for processing. See [Submitting and Processing the AOPD Request on page 12](#).
5. Orchestrate the resulting execution plan. See [Orchestrating the Plan on page 13](#).
6. Amend the order as necessary. See [Amending an Order on page 14](#).

Placing an Order

Before creating an AOPD request, you need to create and submit an order. An order can be created either from a JMS message submitted to iProcess Conductor, or through the normal user interface as described in *TIBCO iProcess Conductor User's Guide*.

A plan fragment can be mapped on to one or more order lines within the order. In the example shown in [Figure 1](#), an order is submitted for the provisioning of a router and a static IP address.

Figure 1 Placing an Order

Order Consolidated View (ORD1000314)

Header

Order Reference: AOPDOrder

Customer Reference: AOPD

Status: Plan Development

Status Changed: 07-Feb-2007 16:08:53

Originator: swadmin

Execution Plan ID: AOPDOrder-1170864563346

Execution Plan Status: Not Started

Required By Date:

Invoice Address: Not defined

Delivery Address: Not defined

SLA: No service level agreements defined

UDF: No User Defined Fields defined

Notes:

Order Lines

	Line No	Product ID	Action	Required Date	Status
+	A	Router	Provide		Pending
+	B	StaticIP	Provide		Pending

Order requests

+ Order request: AOPDOrder

Creating Plan Fragments

Plan fragments are execution plan templates that you can use as building blocks in AOPD. Using such fragments makes it easier to develop an execution plan that will fulfill the order. Multiple plan fragments can be combined to form one consolidated plan which then fulfills the order.

In this example, two plan fragments are created, one to deal with the provision of the router ([Figure 2](#)), and the other for the static IP address ([Figure 3](#)).

Figure 2 A plan fragment for provision of a router

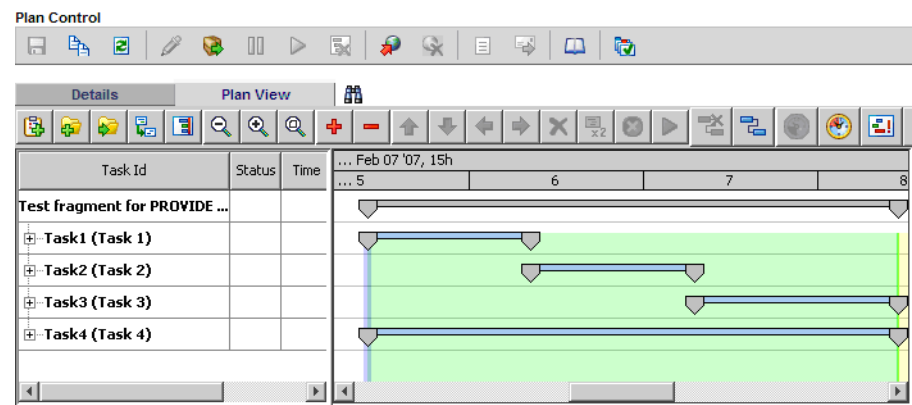
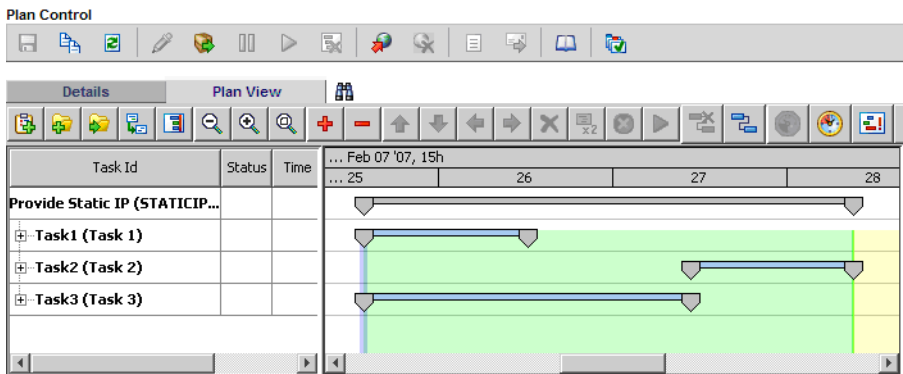


Figure 3 A plan fragment for provision of a static IP address



Plan fragments are imported as summary groups within the consolidated plan. This makes it easier to maintain the overall plan and monitor progress of the overall execution plan.



Plan fragments must either be in the PENDING or TEMPLATE state. Attempting to use a plan fragment in any other state will result in an error and the request will be terminated. See *TIBCO iProcess Conductor Concepts* for details about execution plan status.

Creating an AOPD Request

Once the order and the plan fragments are in place, the next step is to create and submit an AOPD Request. This document can be created manually or through an external application, a process engine (such as TIBCO iProcess Engine or TIBCO BusinessWorks), or a rules engine (such as TIBCO BusinessEvents). The way that you would use these external applications to create such a request is beyond the scope of this manual; see the documentation supplied with the external product for information.

The XML document - see [Example Request XML on page 48](#) for an example - describes the nature of the order request. It defines the outline of the execution plan that will fulfill this order. The major constructs of the request are:

- **Order Identifier:** This is the primary key of the order for which the AOPD request is generated. In our example this is ORD1000314 (see [Figure 1](#)).
- **Items:** Each plan fragment required in the execution plan should be included within an `item` tag. Each plan fragment is imported as a summary group; the `taskId` element provides an identifier for this group.



The task ID must be unique within the scope of the request.

- **User Defined Fields (UDFs):** These name-value pairs are a useful way of providing execution information to the plan fragment. This UDF information will be passed on to each group or sub-task within the plan fragment identified by this item. These fields are also stored as input field information on the start milestone of each subtask. This field information is passed into the process and can thus be made available to the process
- **Order Line ID:** Each item should fulfill at the least one order line within the order. An item can fulfill more than one order line. In our example we have two items which fulfill order lines A and B (see [Figure 1](#)).
- **Dependencies:** Dependencies are a useful way of controlling flow within the execution plan. In our example, the static IP address can only be issued once the router has been delivered and hence the item which deals with the provisioning of the static IP address (`ProvideStaticIP`) has an explicit dependency on the provisioning of the router (`ProvideRouter`)

Refer to the sample request document used in this example - see [Example Request XML on page 48](#). This request maps to the order shown in [Figure 1](#) and makes use of the plan fragments created above. This request can be used as a template for creating future requests.

See [Appendix A, Constructing an AOPD Request](#) which explains the various parts of the AOPD request schema in detail. Once the AOPD request has been created, you must ensure that it conforms to the schema. If the request document is invalid, the request will not be processed and a response message will be sent out to the response queue with the appropriate error code. See [Appendix D, Error Codes](#) for more information on error codes.

Submitting and Processing the AOPD Request

When the request document has been constructed, you can submit it to the AOPD request queue. This can be done by sending it directly through to `AOPDRequestQueue` or by using the `AOPDRequestInjector` client application to send the document to the AOPD request queue. For details on how to do this, see [Sending a Request to the AOPD Request Queue on page 18](#).

Once the request document has been submitted, it will be processed by the iProcess Conductor AOPD message driven APIs and a response message will be sent out with the output to the AOPD response queue.

The initial AOPD request for an order will result in the creation of the execution plan for the order. Subsequent AOPD request submissions for the same order will be considered to be amendments, and the execution plan will be modified.

In our example, since this is the initial AOPD request submission for this order, it will result in the creation of the execution plan that will fulfill the order. Some key points to note within the newly created execution plan are:

- The plan contains two summary groups, each fulfilling one order line as defined in the AOPD request document.
- The dependency that exists between the two groups. The Plan chart display on the execution plan shown in [Figure 4](#) clearly shows that the provisioning of the static IP address cannot be initiated till the provisioning of the router has completed.
- The plan has been set to the PENDING state and is ready for orchestration.

In addition to the execution plan being created, an AOPD response message is also sent out to the AOPD response queue. A response message will always be published for each request, irrespective of the outcome.

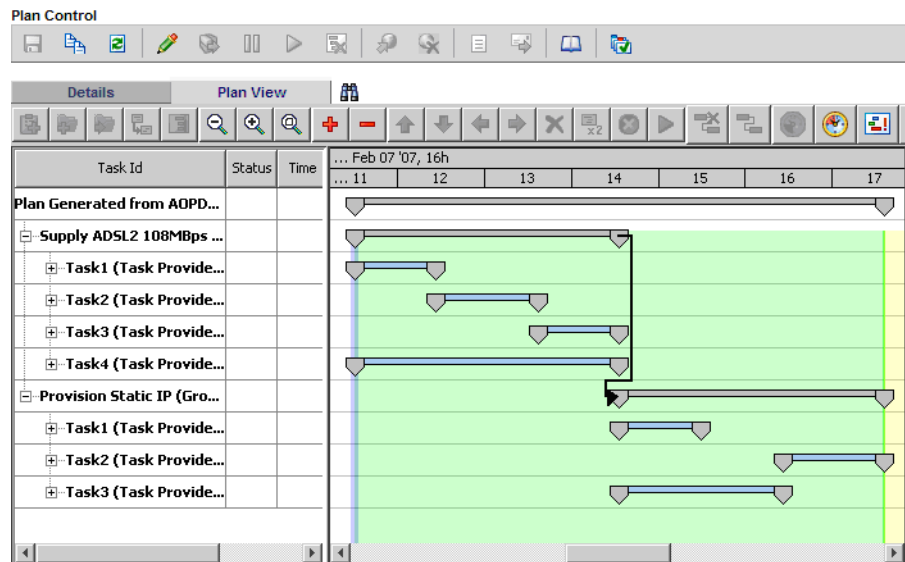
Messages can be read by subscribing to the queue. This can be done manually or using the `AOPDResponseListener` client application. For details on how to do this, see [Receiving a Response from the AOPD Response Queue on page 20](#).

The response document provides the following information:

- The outcome of the AOPD process. This can be one of SUCCESS, WARNING or ERROR.

- The order for which the request has been submitted.
- The generated execution plan.
- The order fulfillment details. These can either be the details of the initial order request, or the latest amendment.
- Key notes that happened while processing the request. Warnings and errors are also documented in this section. For more information on the response schema, see [Appendix B, The AOPD Response](#).

Figure 4 Generated Execution Plan



Orchestrating the Plan

The generated execution plan is now ready for orchestration.

Before you can orchestrate the plan, the order must have gone successfully through the order management lifecycle and should be in the EXECUTION state. This marks the order as ready for fulfillment. Order Management can be carried out using an iProcess Engine process, a BusinessWorks process, or plain Java code.

To orchestrate the plan, send a Start plan message to the Orchestrator. This can be done through an iProcess Engine process using the EAIOrch step.

Amending an Order

An order can be amended at any point before completion. However, the order should be suspended prior to any amendments being done. If orchestration is in progress, suspending the order will also suspend the execution plan.

New order lines can also be added if required.

If the action of an order line is being amended, the only valid change is to set the action status to CANCEL. Changing the action status to anything else would result in a warning and the order line will be skipped. In the example below (Figure 5), order line B is cancelled.

Figure 5 Amended Order

Order Consolidated View (ORD1000314)

Header

Order Reference:

AOPDOrder

Customer Reference:

AOPD

Status:

Plan Development

Status Changed:

07-Feb-2007 16:46:40

Originator:

swadmin

Execution Plan ID:

AOPDOrder-1170864563346

Execution Plan Status:

Not Started

Required By Date:

Invoice Address:

Not defined

Delivery Address:

Not defined

SLA:

No service level agreements defined

UDF:

No User Defined Fields defined

Notes:

Order Lines

	Line No	Product ID	Action	Required Date	Status
+	A	Router	Provide		Pending
+	B	StaticIP	Cancel		Pending

Order requests

+

Order request: AOPDOrder

+

Amendment Order request: AOPDOrder

When cancelling an order line, you must first ensure that `planFragmentUniqueId` in the request document contains a valid plan fragment. In this case, `planFragmentUniqueId` will be considered as the corrective plan fragment. The corrective plan fragment is included in the execution plan if the summary group or any sub-task within the group is complete.

- If the group is active but has completed sub-tasks, the active group is allowed to complete and the corrective plan fragment (imported as a group) is dependent on the active group completing. The corrective plan fragment is imported into a summary group, the task ID of which is prefixed with `Cease`.

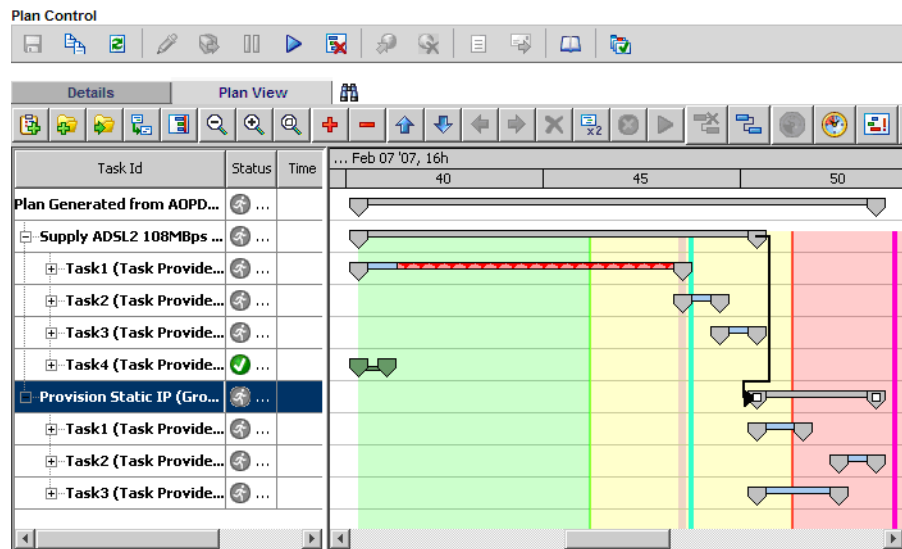
- If the summary group is active and has no completed sub-tasks, the group and its underlying sub-tasks are set to CANCEL. The corrective plan fragment needs to be defined, as in the previous case.

If no corrective action is required, a corrective plan fragment needs to be defined but will not be used. **NO_RECIPROCAL_ACTION** should be entered as `planFragmentUniqueId` in the AOPD request document. This will ignore the status of the group and cancel it.

In our example, the AOPD request document defines the corrective plan fragment for task `ProvideStaticIP` as `STATICIP_CANCEL`. In the runtime status of the plan as shown in [Figure 6](#), the `ProvideStaticIP` summary group is not completed and does not have any completed sub-tasks. Hence, the corrective plan fragment will not be included as part of the execution plan. The summary group and the underlying sub-tasks will be set to CANCEL.

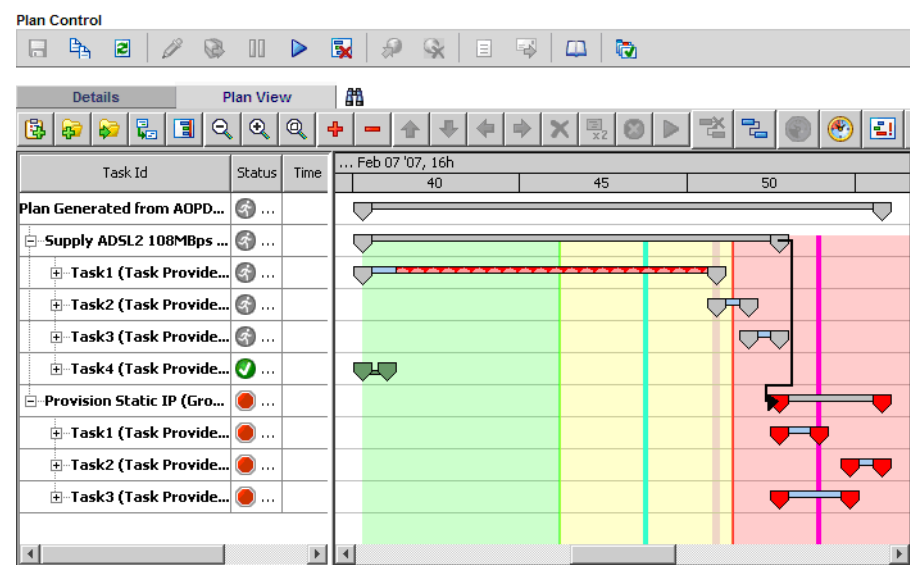
See [Example Request Cancellation XML on page 50](#) for an example of the XML document generated.

Figure 6 Suspended Execution Plan



The procedure to submit an AOPD request document for the amendment is the same as discussed and illustrated for the original request - see [Submitting and Processing the AOPD Request on page 12](#). The request document can be directly submitted to the AOPD request queue or through the `AOPDRequestInjector` client application. [Figure 7](#) shows the execution plan after the AOPD request document has been processed. The `ProvideStaticIP` summary group and all its sub-tasks have been set to CANCEL.

Figure 7 Execution Plan with Group Marked as Cancel



The AOPD response document sent to the AOPD response queue contains the complete execution plan. See [Appendix E, XML Examples](#) for an example.

The execution plan needs to be re-activated so that processing can continue. Re-activating the plan will also re-activate the order. [Figure 8](#) shows the completed execution plan. The summary group `ProvideStaticIP` and all its sub-tasks have been cancelled.

Further Techniques

The following methods may also be useful.

Initial Context

The following code snippet shows how to get the initial context.

```
import javax.naming.InitialContext;
import javax.naming.NamingException;

...

public Context getContext() {

    // The JBOSS application server requires the reference to
    // the auth.conf file which can be found in the
    // $JBOSS_HOME/client directory

    // Uncomment the line below if running on a
    // JBOSS application server
    // System.setProperty("java.security.auth.login.config",
    file:///e:/jboss/client/auth.conf);

    try {
        return new InitialContext(env);
    } catch (NamingException e) {
        // Handle exception here
    }
}
```



This requires the `jndi.properties` file to be present in the classpath. A sample `jndi.properties` file can be seen at [Appendix F, Sample JNDI Properties File](#).

Sending a Request to the AOPD Request Queue

An AOPD request document can be submitted to the AOPD request queue (AOPDProcessQueue) either manually or by using the AOPDRequestInjector client application.

Submitting the Request Manually

The following code snippet shows how to submit a request document manually.

```
import javax.jms.JMSEException;
import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

...

private static final String targetQueue = "AOPDProcessQueue";

public void sendRequest(String xml) {

    QueueConnection con = null;
    QueueSession session = null;
    QueueSender sender = null;
    try {
        Context ctx = getContext();
        QueueConnectionFactory f = (QueueConnectionFactory) ctx
            .lookup("ConnectionFactory");
        Queue q = (Queue) ctx.lookup(targetQueue);

        // Create JMS Connection to AOPD request queue
        con = f.createQueueConnection();
        session = con.createQueueSession(false,
            Session.AUTO_ACKNOWLEDGE);
        sender = session.createSender(q);

        TextMessage message = session.createTextMessage();
        message.setText(xml);

        System.out.println("\nSending message: \n" + message.getText());
        sender.send(message);

    } catch (NamingException e) {
        // Handle exception here
        e.printStackTrace();
    } catch (JMSEException e) {
        // Handle exception here
        e.printStackTrace();
    }
}
```

```

} finally {
    if (con != null) {
        try {
            sender.close();
            session.close();
            con.close();
        } catch (JMSEException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

Using the Request Injector Client

The following code snippet shows how to submit a request document using the AOPD request injector client.

```

import com.staffware.frameworks.opd.client.AOPDRequestInjector;

...

public static void main(String[] args) throws Exception {
    AOPDRequestInjector injector = new AOPDRequestInjector();
    injector.inject(args[0]);
}

```



This requires the `jndi.properties` file to be present in the classpath. A sample `jndi.properties` file can be seen at [Appendix F, Sample JNDI Properties File](#).

Receiving a Response from the AOPD Response Queue

Response messages are sent out to the AOPD response queue (AOPDResponseQueue) irrespective of the outcome of the processing. A response message is always sent for each request. The queue can be monitored manually by:

- Subscribing to the queue.
- Using the AOPDResponseListener to monitor the messages coming on to the queue.
- Writing custom message listeners in which case the AOPDResponseListener can be overridden.

Subscribe to the AOPD Response Queue

The following code snippet shows how to subscribe to the AOPD response queue.

```
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.Session;
import javax.jms.Topic;
import javax.jms.TopicConnection;
import javax.jms.TopicConnectionFactory;
import javax.jms.TopicSession;
import javax.jms.TopicSubscriber;
import javax.naming.Context;

...

private static final String responseQueue = "AOPDResponseQueue";

public void listen() throws Throwable {
    Context ctx = getContext();
    TopicConnectionFactory f = (TopicConnectionFactory)ctx
        .lookup("ConnectionFactory");
    TopicConnection con = f.createTopicConnection();
    TopicSession s = con.createTopicSession(false,
        Session.AUTO_ACKNOWLEDGE);
    Topic t = (Topic) ctx.lookup(responseQueue);
    TopicSubscriber recv = s.createSubscriber(t);
    recv.setMessageListener(this);
    con.start();
    System.out.println("Receiver initialized! Waiting ...");
}

public void onMessage(Message arg0) {
    // Developer should add custom logic here
}
```

Using the AOPDResponseListener

AOPDResponseListener can be used as an application to monitor messages sent to the AOPD response queue.

```
java com.staffware.frameworks.opd.client.AOPDResponseListener
```

Ensure that:

- All supporting jars are present in the classpath.
- The `jndi.properties` file is present in the classpath.

Overriding the AOPDResponseListener

You can use the following simple code snippet to override the AOPDResponseListener to write custom listeners.

```
import javax.jms.TextMessage;

import com.staffware.frameworks.gen.valueobjects.AopdResponse;
import com.staffware.frameworks.opd.client.AOPDResponseListener;

public class CustomResponseListener extends AOPDResponseListener {

    // This method is invoked every time a message arrives on the
    // AOPD response queue. The AopdResponse object is the JAXB
    // version of the response xml document.
    protected void process(TextMessage m, AopdResponse response) {
        // Developer should add custom logic here
    }
}
```



This requires the `jndi.properties` file to be present in the classpath. A sample `jndi.properties` file can be seen at [Appendix F, Sample JNDI Properties File](#).

Getting References

The following code snippet shows how to get a reference to the Execution Plan Façade.

```
import java.rmi.RemoteException;

import javax.ejb.CreateException;
import javax.naming.Context;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;

import com.staffware.frameworks.orchestrator.execution.interfaces.ExecutionPlanFacade;
import com.staffware.frameworks.orchestrator.execution.interfaces.ExecutionPlanFacadeHome;

...

public ExecutionPlanFacade getPlanFacade() {
    Context ctx = getContext();
    String jndi = ExecutionPlanFacadeHome.JNDI_NAME;
    try {
        Object obj = ctx.lookup(jndi);
        ExecutionPlanFacadeHome home = (ExecutionPlanFacadeHome)
            PortableRemoteObject.narrow(obj,
                ExecutionPlanFacadeHome.class);
        return home.create();
    } catch (NamingException e) {
        // Handle Exception
    } catch (RemoteException e) {
        // Handle Exception
    } catch (CreateException e) {
        // Handle Exception
    }
}
```

The following code snippet shows how to get a reference to the Order Façade.

```
import java.rmi.RemoteException;

import javax.ejb.CreateException;
import javax.naming.Context;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;

import com.staffware.frameworks.order.interfaces.OrderFacade;
import com.staffware.frameworks.order.interfaces.OrderFacadeHome;

...

public OrderFacade getOrderFacade() {
    Context ctx = getContext();
    String jndi = OrderFacadeHome.JNDI_NAME;
    try {
        Object obj = ctx.lookup(jndi);
        OrderFacadeHome home = (OrderFacadeHome)
            PortableRemoteObject.narrow(obj,
                OrderFacadeHome.class);
        return home.create();
    } catch (NamingException e) {
        // Handle Exception
    } catch (RemoteException e) {
        // Handle Exception
    } catch (CreateException e) {
        // Handle Exception
    }
}
```

Constructing a Primary Key

The following code snippet shows how to construct a primary key object.

```
import javax.xml.bind.JAXBException;

import com.staffware.frameworks.gen.valueobjects.ManagedEntityPK;
import com.staffware.frameworks.gen.valueobjects.ObjectFactory;

...

public ManagedEntityPK createPrimaryKey(String id) {
    ObjectFactory factory = new ObjectFactory();
    try {
        ManagedEntityPK pk = factory.createManagedEntityPK();
        pk.setId(id);
        return pk;
    } catch (JAXBException e) {
        // Handle Exception
    }
}
```


Appendix A **Constructing an AOPD Request**

This appendix describes the elements of the AOPD request.

Topics

- [AOPD Request, page 28](#)
- [Elements of the AOPD Request, page 29](#)

AOPD Request

The XML document below represents a sample AOPD Request. You can use this as a template to create a custom request.



The XML tags highlighted in blue have to be customized for each execution plan. Each such tag is hyperlinked to its corresponding explanations in the [Elements of the AOPD Request](#) section.

```
<aopdRequest xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
  <orderId>ORD1000063</orderId>
  <originator>AOPDTest</originator>
  <items>
    <item>
      <taskId>Talk500Mobile-4</taskId>
      <description>Group Talk500Mobile-4</description>
    </item>
    <planFragmentUniqueId>EP_Talk500Mobile_PROVIDE</planFragmentUniqueId>
    <udfs>
      <UDF>
        <name>itemSpecId</name>
        <value>Talk500Mobile</value>
      </UDF>
      <UDF>
        <name>action</name>
        <value>CEASE</value>
      </UDF>
      <UDF>
        <name>custItemId</name>
        <value>XYZ</value>
      </UDF>
    </udfs>
    <orderLineIds>
      <orderLineId>A</orderLineId>
    </orderLineIds>
    <dependencies>
      <dependency>
        <point>
          <taskId>Talk500Mobile-5</taskId>
        </point>
      </dependency>
    </dependencies>
  </items>
</aopdRequest>
```


Elements of the AOPD Request

This section describes the elements of the AOPD request which must be customized for each individual request.

Element Name	Element Value
<code>aopdRequest->orderId</code>	This is the primary key of the order for which the AOPD request is generated. This can be determined by examining the order xml (<i>order->primaryKey->id</i>). The order should be present in the iProcess Conductor database before you submit this request. It can either be created manually or through a fulfillment system process.
<code>aopdRequest->originator</code>	The author of this request. This will be maintained over the subsequent execution plan and order modifications.
<code>aopdRequest->items->item</code>	Each item in this request will map to a summary group in the developed plan. Each item contains information necessary to construct a part of the plan.
<code>aopdRequest->items->item->taskId</code>	The task ID of the summary group that will be created. This ID must: • Be unique within the scope of this request. • Not be more than 40 characters • Contain only alpha-numeric characters.
<code>aopdRequest->items->item->description</code>	This is an optional element used to describe this task. If this is not specified, the task ID will be used as the description.
<code>aopdRequest->items->item->planFragmentUniqueId</code>	The unique ID of the plan to be imported as part of the summary group. The plan must exist in the iProcess Conductor database before you submit the request. Only plans in the PENDING and TEMPLATE states can be used as plan fragments. If the plan is in any other state, it will be marked as an error and the processing of this request will be terminated. The response message will contain a corresponding error code. Refer to Appendix D, Error Codes for more information on error codes.

Element Name	Element Value
<code>aopdRequest->items->item->udfs ->UDF</code>	Specifies the User-Defined Fields for this item. Through this, custom user information can be passed to the generated plan, and in turn on to the tasks and subtasks. These fields are also stored as input field information on the start milestone of each subtask. This field information is passed into the process and can thus be made available to the process. A UDF must contain a name and value pair, and must be in the format: <pre><UDF> <name>itemSpecId</name> <value>EmailServer</value> </UDF></pre> For AOPD, there must be at least two UDF entries, and a third is optional: 1. Item specification ID (itemSpecId) - The specification that will be fulfilled by the summary group. This may map to an order line within the submitted order. 2. Action - This can be one of PROVIDE , CEASE , UPDATE or CANCEL . 3. Customer Item ID (custItemId) - The customer item ID to which this AOPD request item relates. This is an optional UDF and is required only if the action is CEASE or UPDATE. There is no limit to the number of user-defined fields that can be added. However, the size of each user defined field must not exceed 4 KB.
<code>aopdRequest->items->item->udfs ->UDF->name</code>	The name for the UDF.
<code>aopdRequest->items->item->udfs ->UDF->value</code>	The value for the UDF.
<code>aopdRequest->items->item->orderLineIds->orderLineId</code>	This maps to one or more lines within the order.
<code>aopdRequest->items->item->dependencies</code>	Defines the dependencies on this item. See Appendix C, Dependencies for more information on the different types of dependencies available.

Appendix B **The AOPD Response**

This appendix describes the elements of the AOPD response.

Topics

- [AOPD Response, page 32](#)
- [Elements of the AOPD Response, page 33](#)

AOPD Response

The XML document below represents a sample AOPD response. You can use this as a guide to the responses you might receive.



The XML tags highlighted in blue have to be customized for each execution plan. Each such tag is hyperlinked to its corresponding explanations in the [Elements of the AOPD Response](#) section.

```
<aopdResponse xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
  <status/>
  <detail>
    <order>
      ...
    </order>
    <orderFulfilment>
      ...
    </orderFulfilment>
    <generatedPlan>
      ...
    </generatedPlan>
    <amendment/>
  </detail>
  <notes>
    <note>
      <code/>
      <note/>
      <type/>
    </note>
  </notes>
</aopdResponse>
```

Elements of the AOPD Response

This section describes the elements of the AOPD response, the contents of which will vary between individual responses.

Element Name	Element Value
<code>aopdResponse->status</code>	The outcome of the AOPD request. The status field can be SUCCESS, WARNING or ERROR. In the case of WARNING and ERROR there will be notes present which will describe the problems encountered while processing the request.
<code>aopdResponse->detail</code>	The output of the AOPD request. There might be no output present, if an error was encountered while processing the request.
<code>aopdResponse->detail->order</code>	The order for which the AOPD request was submitted.
<code>aopdResponse->detail->orderFulfilment</code>	The fulfilment details of the order request or of the latest amendment. This might be null, if no fulfilment details were found within the order.
<code>aopdResponse->detail->generatedPlan</code>	The generated execution plan. This will be the newly created plan if an initial AOPD request was submitted. If the request was an amendment, it will contain the modifications to the plan.
<code>aopdResponse->detail->amendment</code>	Whether or not the AOPD request submitted was an amendment. This will be either true or false.
<code>aopdResponse->notes</code>	Collection of messages during the processing of the AOPD request. Three types of messages can be logged: • INFO messages - these are purely for information. These messages do not affect the AOPD process in any way. • WARNING messages - these are warning messages. Although, the process will complete, it is recommended that the warnings are studied if they have any impact on the processing. See Error Codes on page 44 for the list of WARNING codes. • ERROR messages - these are fatal exceptions which will terminate the AOPD process for the submitted request. See Warning Codes on page 46 for the list of error codes.

Element Name	Element Value
aopdResponse->notes->note->code	The error or warning code. See Appendix D, Error Codes for a complete list. This may not be present if all the messages returned are INFO messages.
aopdResponse->notes->note->note	A brief description of the message.
aopdResponse->notes->note->type	This can be one of INFO, WARNING or ERROR.

Appendix C **Dependencies**

This appendix describes the dependency types and details for each type of dependency.

Topics

- [Types of Dependency, page 36](#)
- [Must Start On Dependencies, page 37](#)
- [Start No Earlier Than Dependencies, page 39](#)
- [Point Dependencies, page 41](#)

Types of Dependency

Dependencies can be placed on a milestone or on a summary group. These dependencies can be of any of three types:

- Must Start On dependencies
- Start No Earlier Than dependencies
- Point dependencies

See *TIBCO iProcess Conductor Concepts* for more information about dependencies.

Must Start On Dependencies

Use this sort of dependency when the summary group or milestone must start on the date and time specified. If it cannot for some reason (for example, because a previous group completes late), a jeopardy condition is triggered. The XML fragment below shows an example Must Start On dependency

```
<time>
  <type>START_ON</type>
  <timeDelta>120000</timeDelta>
  <absoluteTime/>
  <consequentialActions>
    <def>
      <type>section.THRESHOLD</type>
      <actions>
        <action>JMS_MESSAGE</action>
        <action>WORK_ITEM</action>
      </actions>
    </def>
  </consequentialActions>
</time>
```

Element Name	Element Value
time->timeDelta	The time in milliseconds, measured from plan activation, that the milestone or summary group should start execution.
time->absoluteTime	The absolute time when the milestone or summary group should begin executing. If both timeDelta and absoluteTime are defined, priority is given to the timeDelta value.

Element Name	Element Value
<code>time->consequentialActions->def->type</code>	The jeopardy configuration for this dependency if the milestone or summary group does not start at the configured date and time. This can be one of: <code>section.THRESHOLD</code>: Event generated when the time has passed and the milestone or summary group has not started execution. <code>section.PREDICTION_CROSSOVER_UP</code>: Event generated when the dependency is predicted to be violated in the upward direction. <code>section.PREDICTION_CROSSOVER_DOWN</code>: Event generated when the dependency is predicted to be violated in the downward direction. See <i>TIBCO iProcess Conductor Concepts</i> for information on jeopardy event types.
<code>time->consequentialActions->def->actions->action</code>	The action to be taken when the particular jeopardy event described above is detected. This can be one of: <code>SUSPEND_PLAN</code>: Suspend the plan. <code>JMS_MESSAGE</code>: Send a JMS message. This will be sent out to the <code>IPCJeopardyTopic</code>. <code>WORK_ITEM</code>: Start an iProcess Engine corrective action case. See <i>TIBCO iProcess Conductor Concepts</i> for more information on jeopardy event actions.

Start No Earlier Than Dependencies

Use this sort of dependency when the milestone or summary group must not begin execution before the date/time specified. If the date/time is reached and other dependencies to the milestone have not been satisfied, the milestone remains in PENDING status. The XML fragment below shows an example Start No Earlier Than dependency.

```
<time>
  <type>START_NO_EARLIER_THAN</type>
  <timeDelta>120000</timeDelta>
  <absoluteTime/>
  <consequentialActions>
    <def>
      <type>section.THRESHOLD</type>
      <actions>
        <action>JMS_MESSAGE</action>
        <action>WORK_ITEM</action>
      </actions>
    </def>
  </consequentialActions>
</time>
```

Element Name	Element Value
time->timeDelta	The time in milliseconds, measured from plan activation, that the milestone or summary group should start execution.
time->absoluteTime	The absolute time when the milestone or summary group should begin executing. If both timeDelta and absoluteTime are defined, priority is given to the timeDelta value.
time->consequentialActions->def->type	The jeopardy configuration for this dependency if the milestone or summary group starts earlier than the configured time. This can be: • section.THRESHOLD: Event generated when the time has passed and the milestone or summary group has started execution. See <i>TIBCO iProcess Conductor Concepts</i> for information on jeopardy event types.

Element Name	Element Value
time->consequentialActions->def->actions->action	The action to be taken when the particular jeopardy event described above is detected. This can be one of: • SUSPEND_PLAN: Suspend the plan. • JMS_MESSAGE: Send a JMS message. This will be sent out to the IPCJeopardyTopic. • WORK_ITEM: Start an iProcess Engine corrective action case. See <i>TIBCO iProcess Conductor Concepts</i> for information on jeopardy event actions.

Point Dependencies

Point dependencies can be defined between two summary groups. The XML fragment below shows an example point dependency.

```
<point>  
  <timeDelta>0</timeDelta>  
  <taskId>SIMCard</taskId>  
</point>
```

Element Name	Element Value
point->timeDelta	The time in milliseconds, measured from plan activation, that the summary group should wait after the dependency is satisfied. Note that: • Although the schema does not mandate it, you are recommended to set the timeDelta to zero or a greater value. If you do not, you may not be able to view the summary group dependency screen correctly in iProcess Conductor. • Also, though the schema does not mandate it, the timeDelta tag should exist. If it does not exist, you may not be able to view the plan correctly in the Plan view.
point->taskId	The ID of the 'dependent on' summary group.

See *TIBCO iProcess Conductor Concepts* for more information on inter-group dependencies.

Appendix D **Error Codes**

The iProcess Conductor AOPD Message-Driven API provides error codes for each warning and error condition. This makes it easier to identify and correct any problems.

Topics

- [Error Codes, page 44](#)
- [Warning Codes, page 46](#)

Error Codes

If an error is detected, the processing of the request is terminated and the response is sent out to the AOPD response queue. The response will have a status of `ERROR`. The response may or may not contain detail, so it is recommended the status of the response is checked. In the case of an error it is likely that the detail will be `null` or will contain invalid information.

AOPD error codes start with the string `AOPD-ERR-xxx`. A detailed list of error codes is given in the table below.

Error Code	Description and Reason	Corrective Action
AOPD-ERR-INV-REQ	Invalid request! [<i>XML DOC</i>] An invalid request was sent to the AOPD request queue. The request document <i>XML DOC</i> will be included as part of the error response.	Check that the request conforms to the <code>aopd.xsd</code> schema.
AOPD-ERR-INV-MSG-TYPE	Invalid message type! Message [<i>xxx</i>] not a text message! An invalid request was sent to the AOPD request queue. The request must be sent as a <code>TextMessage</code> .	Ensure that the message is sent as a <code>TextMessage</code> .
AOPD-ERR-INV-FRAG-STATUS	Plan fragment [<i>xxx</i>] has an invalid status [<i>STATUS</i>]! Should be <code>PENDING</code> or <code>TEMPLATE</code> ! The plan fragment <i>xxx</i> defined in the AOPD request document is in an invalid state. Plan fragments must be in either <code>PENDING</code> or <code>TEMPLATE</code> status.	Ensure that the correct plan fragment is specified in the request document. Check for typographical errors. If the fragment is in a state other than <code>PENDING</code> or <code>TEMPLATE</code> , make a copy of the plan, save it and either publish it to <code>TEMPLATE</code> or set it to <code>PENDING</code> .

Error Code	Description and Reason	Corrective Action
AOPD-ERR-INV-FRAG	Plan fragment [xxx] not defined! The plan fragment <i>xxx</i> is not found in the iProcess Conductor database.	Ensure that the AOPD request is sent to the correct iProcess Conductor server. Ensure that the correct plan fragment is specified in the request document, and check for typographical errors. Create the plan fragment and re-submit the request.
AOPD-ERR-INV-ORDER	Order [xxx] not defined! The order <i>xxx</i> defined in the request document is not found in the iProcess Conductor database.	Ensure that the AOPD request is sent to the correct iProcess Conductor server. Ensure that the correct order ID is specified in the request document, and check for typographical errors. Create the order and re-submit the request.
AOPD-ERR-UNKNOWN	Unknown error [xxx]! An unknown error occurred while processing the AOPD request document. The description, if available, will be provided in square brackets.	Check the server trace files for more information.

Warning Codes

If a warning is generated, the processing of the request is completed and the response is sent out to the AOPD response queue. All warnings encountered during the processing of the request will be sent out together with the response. The response will have a status of WARNING. The response will contain valid detail, but it is recommended that the warnings are checked.

AOPD warning codes start with the string AOPD-WARN-xxx. A detailed list of error codes is given in the table below.

Error Code	Description and Reason	Corrective Action
AOPD-WARN-INV-ACT	Cannot change action for order line [xxx] from [aaa] to [bbb]! An illegal action has been encountered for an order line. The processing will be skipped for the order line xxx.	Check the request for typographical errors for the order line action. The order line action can only be changed to CANCEL.
AOPD-WARN-INV-SPEC	Specification [xxx] not found in the amended order! The specification [xxx] was not found in the amended order.	Ensure that all specifications are part of the amendment. Check the request for typographical errors.
AOPD-WARN-FUL-PROC	No fulfillment details found! No fulfillment details were found as part of the order request or the latest amendment.	If the order is generated as part of an iProcess Engine process, it is recommended to include the fulfillment details either as part of the order request or as part of the latest amendment.

Appendix E **XML Examples**

This appendix contains examples of the type of XML documents used in AOPD.

Topics

- [Example Request XML, page 48](#)
- [Example Request Cancellation XML, page 50](#)
- [Example Response XML, page 52](#)
- [Example Cancellation Response XML, page 70](#)

Example Request XML

The XML document below represents an example AOPD request corresponding to the example used in [Creating an AOPD Request on page 11](#), and making use of the plan fragments created in [Creating Plan Fragments on page 9](#). This request can be used as a template for creating future requests.

```

<aopdRequest xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
  <orderId>ORD1000314</orderId>
  <originator>AOPDTest</originator>
  <items>
    <item>
      <taskId>ProvideRouter</taskId>
      <description>Supply ADSL2 108Mbps Wireless Router</description>
      <planFragmentUniqueId>ROUTER_PROVIDE</planFragmentUniqueId>
      <udfs>
        <UDF>
          <name>itemSpecId</name>
          <value>Router</value>
        </UDF>
        <UDF>
          <name>action</name>
          <value>PROVIDE</value>
        </UDF>
      </udfs>
      <orderLineIds>
        <orderLineId>A</orderLineId>
      </orderLineIds>
    </item>
    <item>
      <taskId>ProvideStaticIP</taskId>
      <description>Provision Static IP</description>
      <planFragmentUniqueId>STATICIP_PROVIDE</planFragmentUniqueId>
      <udfs>
        <UDF>
          <name>itemSpecId</name>
          <value>StaticIP</value>
        </UDF>
        <UDF>
          <name>action</name>
          <value>PROVIDE</value>
        </UDF>
      </udfs>
      <orderLineIds>
        <orderLineId>B</orderLineId>
      </orderLineIds>
      <dependencies>
        <dependency>
          <point>
            <taskId>ProvideRouter</taskId>
          </point>
        </dependency>
      </dependencies>
    </item>
  </items>
</aopdRequest>

```

Example Request Cancellation XML

The XML document below represents an example AOPD cancellation request corresponding to the example used in [Amending an Order on page 14](#). This request can be used as a template for creating future amendment requests.

```

<aopdRequest xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
  <orderId>ORD1000314</orderId>
  <originator>AOPDTest</originator>
  <items>
    <item>
      <taskId>ProvideRouter</taskId>
      <description>Supply ADSL2 108Mbps Wireless Router</description>
      <planFragmentUniqueId>ROUTER_PROVIDE</planFragmentUniqueId>
      <udfs>
        <UDF>
          <name>itemSpecId</name>
          <value>Router</value>
        </UDF>
        <UDF>
          <name>action</name>
          <value>PROVIDE</value>
        </UDF>
      </udfs>
      <orderLineIds>
        <orderLineId>A</orderLineId>
      </orderLineIds>
    </item>
    <item>
      <taskId>ProvideStaticIP</taskId>
      <description>Provision Static IP</description>
      <planFragmentUniqueId>STATICIP_CANCEL</planFragmentUniqueId>
      <udfs>
        <UDF>
          <name>itemSpecId</name>
          <value>StaticIP</value>
        </UDF>
        <UDF>
          <name>action</name>
          <value>CANCEL</value>
        </UDF>
      </udfs>
      <orderLineIds>
        <orderLineId>B</orderLineId>
      </orderLineIds>
      <dependencies>
        <dependency>
          <point>
            <taskId>ProvideRouter</taskId>
          </point>
        </dependency>
      </dependencies>
    </item>
  </items>
</aopdRequest>

```

Example Response XML

The XML document below represents an example response to an AOPD request. It corresponds to the example used elsewhere in this document.


```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<aopdResponse xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
<status>SUCCESS</status>
- <detail>
<order>
<lastUpdateVersionNumber>17</lastUpdateVersionNumber>
  <primaryKey>
<id>ORD1000314</id>
    </primaryKey>
    <header>
<orderHeader>
<orderRef>AOPDOrder</orderRef>
<customerRef>AOPD</customerRef>
    </orderHeader>
    <status>OPD</status>
    <statusChanged>2012-02-07T16:46:40</statusChanged>
    <stateChanger>7114</stateChanger>
    <suspended>false</suspended>
    <originator>swadmin</originator>
    <planUniqueId>AOPDOrder-1172507110483</planUniqueId>
    <orderPlanStatus>ORDER PLAN NOT STARTED</orderPlanStatus>
  </header>
  <lines>
<line>
<orderLine>
  <lineNumber>A</lineNumber>
  <productId>Router</productId>
  <quantity>1</quantity>
  <UOM>unit</UOM>
  <orderLineAction>PROVIDE</orderLineAction>
  </orderLine>
  <status>PENDING</status>
  <statusChanged>2012-02-07T15:17:50</statusChanged>
</line>
<line>
<line>
<orderLine>
  <lineNumber>B</lineNumber>
  <productId>StaticIP</productId>
  <quantity>1</quantity>
  <UOM>IP</UOM>
  <orderLineAction>PROVIDE</orderLineAction>
  </orderLine>
  <status>PENDING</status>
  <statusChanged>2012-02-07T15:18:25</statusChanged>
</line>
</lines>
  <request>
<orderRequest>
<orderHeader>
  <orderRef>AOPDOrder</orderRef>
  <customerRef>AOPD</customerRef>
</orderHeader>

```

```

<orderLines>
<orderLine>
  <lineNumber>A</lineNumber>
  <productId>Router</productId>
  <quantity>1</quantity>
  <UOM>unit</UOM>
  <orderLineAction>PROVIDE</orderLineAction>
</orderLine>
<orderLine>
  <lineNumber>B</lineNumber>
  <productId>StaticIP</productId>
  <quantity>1</quantity>
  <UOM>IP</UOM>
  <orderLineAction>PROVIDE</orderLineAction>
</orderLine>
</orderLines>
  </orderRequest>
<fulfilment>
<fulfilmentProc>
<procName>BEORDFUL</procName>
<description>Generated Order Request</description>
<system>
<id>iProcess_11.1</id>
  <properties>
<property name="eventStep">CANCEL</property>
<property name="version.major">-1</property>
<property name="version.minor">-1</property>
<property name="caseNumber">7101</property>
<property name="mainCaseNumber">7101</property>
<property name="mainProcName">BEORDFUL</property>
</properties>
  </system>
  <fields />
</fulfilmentProc>
<originator>swadmin</originator>
<submissionDate>2012-02-07T16:08:49</submissionDate>
  <advisoryStrings>
<advisoryString>
  <advice>Automatic Order Plan development scheduled for this order</advice>
</advisoryString>
</advisoryStrings>
<suspendable>true</suspendable>
  </fulfilment>
</request>
</order>
<orderFulfilment>
  <procName>BEORDFUL</procName>
<description>Generated Order Request</description>
<system>
<id>iProcess_11.1</id>

```

```

<properties>
<property name="eventStep">CANCEL</property>
<property name="version.major">-1</property>
<property name="version.minor">-1</property>
<property name="caseNumber">7101</property>
<property name="mainCaseNumber">7101</property>
<property name="mainProcName">BEORDFUL</property>
</properties>
</system>
<fields />
</orderFulfilment>
<generatedPlan>
<lastUpdateVersionNumber>1</lastUpdateVersionNumber>
<primaryKey>
<id>1000578</id>
</primaryKey>
<executionPlanHeader>
<uniqueID>AOPDOrder-1172507110483</uniqueID>
<originator>AOPDTest</originator>
<masterRecordID>ORD1000314</masterRecordID>
<planCreationDate>2012-02-26T16:25:10</planCreationDate>
<status>PENDING</status>
<statusChanged>2012-02-26T16:25:10</statusChanged>
<description>Plan Generated from AOPDOrder</description>
<fulfilmentProc>
<procName>TBD</procName>
<description>Fulfilment proc</description>
<system>
<id>iProcess_11.1</id>
<properties />
</system>
</fulfilmentProc>
<schedule>
<jeopardy>
<startDate>2012-02-26T16:25:10</startDate>
<endDate>2012-02-26T16:31:10</endDate>
<projectedDuration>
<typical>360000</typical>
<maxAllowed>660000</maxAllowed>
</projectedDuration>
<designTimeDuration>
<typical>360000</typical>
<maxAllowed>660000</maxAllowed>
</designTimeDuration>
<jeopardyState>UNKNOWN</jeopardyState>
</jeopardy>
</schedule>
</executionPlanHeader>
<executionPlanItems>
<summary>
<taskId>ProvideRouter</taskId>
<description>Supply ADSL2 108Mbps Wireless Router</description>
<status>PENDING</status>
<statusChanged>2012-02-26T16:25:10</statusChanged>

```

```

    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <orderLineIds>
    <orderLineId>A</orderLineId>
    </orderLineIds>
    <executionPlanItems>
    <task>
    <taskId>ProvideRouter.1</taskId>
    <description>Task1</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iProcess_11.1</id>
    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
    </system>
    </procInfo>
    <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>

```

```

<point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
<fields>
<field io="OUT" name="itemSpecId">Router</field>
<field io="OUT" name="action">PROVIDE</field>
</fields>
</startMilestone>
<endMilestone>
<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
<properties>
<property name="stepType" />
<property name="notifyOnly" />
</properties>
<point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
<fields />
</endMilestone>
<sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
<schedule>
<typicalTiming>
<duration>60000</duration>
</typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
</maxAllowedTiming>
</schedule>
</section>
</sections>
<schedule>
<typicalTiming>
<duration>60000</duration>
</typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
</maxAllowedTiming>
</schedule>
</processVersion>
</task>
<task>
<taskId>ProvideRouter.2</taskId>
<description>Task2</description>
<status>PENDING</status>
<statusChanged>2012-02-07T14:58:53</statusChanged>
<UDFs>
<UDF>
<name>itemSpecId</name>
<value>Router</value>
</UDF>

```

```

<UDF>
<name>action</name>
<value>PROVIDE</value>
</UDF>
</UDFs>
<objectXrefs />
<itemId />
<action>PROVIDE</action>
<uniqueCompID>IT3STP</uniqueCompID>
  <processVersion>
    <procInfo>
      <procName>TBD</procName>
      <description>TBD</description>
      <system>
        <id>iPROCESS_11.1</id>
        <properties>
          <property name="majorVersion">-1</property>
          <property name="minorVersion">-1</property>
          <property isEditable="true" name="exceptionHandler" />
          <property name="caseNumber">0</property>
        </properties>
      </system>
    </procInfo>
    <startMilestone>
      <description>Start Task</description>
      <id>START</id>
      <status>PENDING</status>
      <properties>
        <property name="stepType" />
        <property name="notifyOnly" />
      </properties>
      <point>
        <dependencies>
          <dependency>
            <point>
              <timeDelta>0</timeDelta>
              <taskId>ProvideRouter.1</taskId>
              <type>START_TO_FINISH</type>
              <milestoneId>END</milestoneId>
            </point>
          </dependency>
        </dependencies>
        <releaseMillis>0</releaseMillis>
      </point>
      <fields>
        <field io="OUT" name="itemSpecId">Router</field>
        <field io="OUT" name="action">PROVIDE</field>
      </fields>
    </startMilestone>
    <endMilestone>
      <description>Task Complete</description>
      <id>END</id>
      <status>PENDING</status>
      <properties>
        <property name="stepType" />
        <property name="notifyOnly" />
      </properties>

```

```

    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields />
    </endMilestone>
    <sections>
    <section>
    <startMilestoneId>START</startMilestoneId>
    <finishMilestoneId>END</finishMilestoneId>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </section>
    </sections>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </processVersion>
    </task>
    <task>
    <taskId>ProvideRouter.3</taskId>
    <description>Task3</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iPROCESS_11.1</id>

```

```

    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
  </system>
</procInfo>
  <startMilestone>
<description>Start Task</description>
<id>START</id>
<status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
<dependencies>
  <dependency>
    <point>
<timeDelta>0</timeDelta>
<taskId>ProvideRouter.2</taskId>
<type>START_TO_FINISH</type>
<milestoneId>END</milestoneId>
    </point>
  </dependency>
  </dependencies>
  <releaseMillis>0</releaseMillis>
</point>
  <fields>
    <field io="OUT" name="itemSpecId">Router</field>
    <field io="OUT" name="action">PROVIDE</field>
  </fields>
</startMilestone>
  <endMilestone>
<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
  <fields />
</endMilestone>
  <sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>

```



```

    <schedule>
      <typicalTiming>
        <duration>60000</duration>
      </typicalTiming>
      <maxAllowedTiming>
        <duration>120000</duration>
      </maxAllowedTiming>
    </schedule>
  </section>
</sections>
  <schedule>
    <typicalTiming>
      <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>120000</duration>
    </maxAllowedTiming>
  </schedule>
</processVersion>
</task>
<task>
  <taskId>ProvideRouter.4</taskId>
  <description>Task4</description>
  <status>PENDING</status>
  <statusChanged>2012-02-07T14:58:53</statusChanged>
  <UDFs>
    <UDF>
      <name>itemSpecId</name>
      <value>Router</value>
    </UDF>
    <UDF>
      <name>action</name>
      <value>PROVIDE</value>
    </UDF>
  </UDFs>
  <objectXrefs />
  <itemId />
  <action>PROVIDE</action>
  <uniqueCompID>IT3STP</uniqueCompID>
  <processVersion>
    <procInfo>
      <procName>TBD</procName>
      <description>TBD</description>
      <system>
        <id>iPROCESS_11.1</id>
        <properties>
          <property name="majorVersion">-1</property>
          <property name="minorVersion">-1</property>
          <property isEditable="true" name="exceptionHandler" />
          <property name="caseNumber">0</property>
        </properties>
      </system>
    </procInfo>
  </processVersion>

```

```

<startMilestone>
<description>Start Task</description>
<id>START</id>
<status>PENDING</status>
  <properties>
<property name="stepType" />
<property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
  </point>
  <fields>
<field io="OUT" name="itemSpecId">Router</field>
<field io="OUT" name="action">PROVIDE</field>
  </fields>
</startMilestone>
<endMilestone>
<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
  <properties>
<property name="stepType" />
<property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
  </point>
  <fields />
</endMilestone>
<sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
  <schedule>
<typicalTiming>
<duration>180000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
  </typicalTiming>
<maxAllowedTiming>
<duration>360000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
  </maxAllowedTiming>
<actualDuration>0</actualDuration>
</schedule>
</section>
</sections>

```

```

    <schedule>
      <typicalTiming>
        <duration>180000</duration>
        <earlyStart>0</earlyStart>
        <lateStart>0</lateStart>
        <isOnCriticalPath>true</isOnCriticalPath>
      </typicalTiming>
      <maxAllowedTiming>
        <duration>360000</duration>
        <earlyStart>0</earlyStart>
        <lateStart>0</lateStart>
        <isOnCriticalPath>true</isOnCriticalPath>
      </maxAllowedTiming>
      <actualDuration>0</actualDuration>
    </schedule>
    <isScheduleOverride>true</isScheduleOverride>
  </processVersion>
</task>
</executionPlanItems>
<startPoint />
  <schedule>
    <typicalTiming>
      <duration>180000</duration>
      <earlyStart>0</earlyStart>
      <lateStart>0</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>360000</duration>
      <earlyStart>0</earlyStart>
      <lateStart>0</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </maxAllowedTiming>
  </schedule>
  <planFragmentUniqueId>ROUTER_PROVIDE</planFragmentUniqueId>
</summary>
  <summary>
    <taskId>ProvideStaticIP</taskId>
    <description>Provision Static IP</description>
    <status>PENDING</status>
    <statusChanged>2012-02-26T16:25:10</statusChanged>
    <UDFs>
      <UDF>
        <name>itemSpecId</name>
        <value>StaticIP</value>
      </UDF>
      <UDF>
        <name>action</name>
        <value>PROVIDE</value>
      </UDF>
    </UDFs>
    <orderLineIds>
      <orderLineId>B</orderLineId>
    </orderLineIds>

```

```

    <executionPlanItems>
    <task>
    <taskId>ProvideStaticIP.1</taskId>
    <description>Task1</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>StaticIP</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iPROCESS_11.1</id>
    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
    </system>
    </procInfo>
    <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>
    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields>
    <field io="OUT" name="itemSpecId">StaticIP</field>
    <field io="OUT" name="action">PROVIDE</field>
    </fields>
    </startMilestone>
    <endMilestone>
    <description>Task Complete</description>
    <id>END</id>
    <status>PENDING</status>

```

```

    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>
    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields />
    </endMilestone>
    <sections>
    <section>
    <startMilestoneId>START</startMilestoneId>
    <finishMilestoneId>END</finishMilestoneId>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </section>
    </sections>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </processVersion>
    </task>
    <task>
    <taskId>ProvideStaticIP.2</taskId>
    <description>Task2</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>StaticIP</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>

```

```

    <procName>TBD</procName>
  <description>TBD</description>
  <system>
    <id>iPROCESS_11.1</id>
    <properties>
      <property name="majorVersion">-1</property>
      <property name="minorVersion">-1</property>
      <property isEditable="true" name="exceptionHandler" />
      <property name="caseNumber">0</property>
    </properties>
  </system>
</procInfo>
  <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
      <property name="stepType" />
      <property name="notifyOnly" />
    </properties>
    <point>
      <dependencies>
        <dependency>
          <point>
            <timeDelta>0</timeDelta>
            <taskId>ProvideStaticIP.1</taskId>
            <type>START_TO_FINISH</type>
            <milestoneId>END</milestoneId>
          </point>
        </dependency>
      </dependencies>
      <point>
        <timeDelta>0</timeDelta>
        <taskId>ProvideStaticIP.3</taskId>
        <type>START_TO_FINISH</type>
        <milestoneId>END</milestoneId>
      </point>
    </point>
    <fields>
      <field io="OUT" name="itemSpecId">StaticIP</field>
      <field io="OUT" name="action">PROVIDE</field>
    </fields>
  </startMilestone>
  <endMilestone>
    <description>Task Complete</description>
    <id>END</id>
    <status>PENDING</status>
    <properties>
      <property name="stepType" />
      <property name="notifyOnly" />
    </properties>
  </endMilestone>

```

```

    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields />
    </endMilestone>
    <sections>
    <section>
    <startMilestoneId>START</startMilestoneId>
    <finishMilestoneId>END</finishMilestoneId>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </section>
    </sections>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </processVersion>
    </task>
    <task>
    <taskId>ProvideStaticIP.3</taskId>
    <description>Task3</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>StaticIP</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iPROCESS_11.1</id>

```

```

    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
  </system>
  </procInfo>
  <startMilestone>
  <description>Start Task</description>
  <id>START</id>
  <status>PENDING</status>
  <properties>
  <property name="stepType" />
  <property name="notifyOnly" />
  </properties>
  <point>
  <dependencies />
  <releaseMillis>0</releaseMillis>
  </point>
  <fields>
  <field io="OUT" name="itemSpecId">StaticIP</field>
  <field io="OUT" name="action">PROVIDE</field>
  </fields>
  </startMilestone>
</endMilestone>
<description>Task Complete</description>
  <id>END</id>
  <status>PENDING</status>
  <properties>
  <property name="stepType" />
  <property name="notifyOnly" />
  </properties>
  <point>
  <dependencies />
  <releaseMillis>0</releaseMillis>
  </point>
  <fields />
</endMilestone>
  <sections>
  <section>
  <startMilestoneId>START</startMilestoneId>
  <finishMilestoneId>END</finishMilestoneId>
  <schedule>
  <typicalTiming>
  <duration>120000</duration>
  <earlyStart>180000</earlyStart>
  <lateStart>180000</lateStart>
  <isOnCriticalPath>true</isOnCriticalPath>
  </typicalTiming>
  <maxAllowedTiming>
  <duration>180000</duration>
  <earlyStart>360000</earlyStart>
  <lateStart>360000</lateStart>
  <isOnCriticalPath>true</isOnCriticalPath>
  </maxAllowedTiming>

```



```

    <actualDuration>0</actualDuration>
  </schedule>
</section>
</sections>
  <schedule>
    <typicalTiming>
      <duration>120000</duration>
      <earlyStart>180000</earlyStart>
      <lateStart>180000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>180000</duration>
      <earlyStart>360000</earlyStart>
      <lateStart>360000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </maxAllowedTiming>
    <actualDuration>0</actualDuration>
  </schedule>
  <isScheduleOverride>true</isScheduleOverride>
  </processVersion>
</task>
</executionPlanItems>
<startPoint>
  <dependencies>
    <dependency>
      <point>
        <timeDelta>0</timeDelta>
        <taskId>ProvideRouter</taskId>
        <type>FINISH_TO_START</type>
      </point>
    </dependency>
  </dependencies>
</startPoint>
  <schedule>
    <typicalTiming>
      <duration>180000</duration>
      <earlyStart>180000</earlyStart>
      <lateStart>180000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>300000</duration>
      <earlyStart>360000</earlyStart>
      <lateStart>360000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </maxAllowedTiming>
  </schedule>
<planFragmentUniqueId>STATICIP_PROVIDE</planFragmentUniqueId>
</summary>
</executionPlanItems>
</generatedPlan>
<amendment>false</amendment>
</detail>
</aopdResponse>

```

Example Cancellation Response XML

The XML document below represents an example response to an AOPD cancellation. It corresponds to the example used elsewhere in this document.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<aopdResponse xmlns="http://www.staffware.com/frameworks/gen/valueobjects">
<status>SUCCESS</status>
- <detail>
<order>
<lastUpdateVersionNumber>23</lastUpdateVersionNumber>
  <primaryKey>
<id>ORD1000314</id>
    </primaryKey>
    <header>
<orderHeader>
<orderRef>AOPDOrder</orderRef>
<customerRef>AOPD</customerRef>
    </orderHeader>
    <status>OPD</status>
    <statusChanged>2012-02-26T16:39:07</statusChanged>
    <stateChanger>7305</stateChanger>
    <suspended>false</suspended>
    <originator>swadmin</originator>
    <planUniqueId>AOPDOrder-1172507110483</planUniqueId>
    <orderPlanStatus>ORDER PLAN NOT STARTED</orderPlanStatus>
  </header>
  <lines>
  <line>
<orderLine>
  <lineNumber>A</lineNumber>
  <productId>Router</productId>
  <quantity>1</quantity>
  <UOM>unit</UOM>
  <orderLineAction>PROVIDE</orderLineAction>
  </orderLine>
  <status>PENDING</status>
  <statusChanged>2012-02-07T15:17:50</statusChanged>
  </line>
<line>
<line>
<orderLine>
  <lineNumber>B</lineNumber>
  <productId>StaticIP</productId>
  <quantity>1</quantity>
  <UOM>IP</UOM>
  <orderLineAction>CANCEL</orderLineAction>
  </orderLine>
  <status>PENDING</status>
  <statusChanged>2012-02-07T15:18:25</statusChanged>
  </line>
</lines>
  <request>
<orderRequest>
<orderHeader>
  <orderRef>AOPDOrder</orderRef>
  <customerRef>AOPD</customerRef>
  </orderHeader>

```

```

    <orderLines>
    <orderLine>
      <lineNumber>A</lineNumber>
      <productId>Router</productId>
      <quantity>1</quantity>
      <UOM>unit</UOM>
      <orderLineAction>PROVIDE</orderLineAction>
    </orderLine>
    <orderLine>
      <lineNumber>B</lineNumber>
      <productId>StaticIP</productId>
      <quantity>1</quantity>
      <UOM>IP</UOM>
      <orderLineAction>PROVIDE</orderLineAction>
    </orderLine>
    </orderLines>
  </orderRequest>
</fulfilment>
<fulfilmentProc>
<procName>BEORDFUL</procName>
<description>Generated Order Request</description>
<system>
<id>iProcess_11.1</id>
  <properties>
    <property name="eventStep">CANCEL</property>
    <property name="version.major">-1</property>
    <property name="version.minor">-1</property>
    <property name="caseNumber">7101</property>
    <property name="mainCaseNumber">7101</property>
    <property name="mainProcName">BEORDFUL</property>
  </properties>
</system>
  <fields />
</fulfilmentProc>
<originator>swadmin</originator>
<submissionDate>2012-02-07T16:08:49</submissionDate>
  <advisoryStrings>
    <advisoryString>
      <advice>Automatic Order Plan development scheduled for this order</advice>
    </advisoryString>
  </advisoryStrings>
<suspendable>true</suspendable>
</fulfilment>
</request>
<amendments>
<orderAmendment>
<orderRequest>
<orderHeader>
  <orderRef>AOPDOrder</orderRef>
  <customerRef>AOPD</customerRef>
</orderHeader>
<orderLines>
  <orderLine>
    <lineNumber>A</lineNumber>
    <productId>Router</productId>

```

```

<productId>Router</productId>
<quantity>1</quantity>
<UOM>unit</UOM>
<orderLineAction>CANCEL</orderLineAction>
  </orderLine>
</orderlines>
</orderRequest>
  <fulfilment>
    <fulfilmentProc>
      <procName>BEORDFUL</procName>
      <description>Generated Order Request</description>
      <system>
        <id>iPROCESS_11.1</id>
        <properties>
          <property name="eventStep">CANCEL</property>
          <property name="version.major">-1</property>
          <property name="version.minor">-1</property>
          <property name="caseNumber">7305</property>
          <property name="mainCaseNumber">7305</property>
          <property name="mainProcName">BEORDFUL</property>
        </properties>
      </system>
    </fulfilmentProc>
    <originator>swadmin</originator>
    <submissionDate>2012-02-26T16:38:57</submissionDate>
    <advisoryStrings>
      <advisoryString>
        <advice>Automatic Order Plan development scheduled for this order</advice>
      </advisoryString>
    </advisoryStrings>
    <suspendable>true</suspendable>
  </fulfilment>
</orderAmendment>
  </amendments>
</order>
<orderFulfilment>
  <procName>BEORDFUL</procName>
  <description>Generated Order Request</description>
  <system>
    <id>iProcess_11.1</id>
    <properties>
      <property name="eventStep">CANCEL</property>
      <property name="version.major">-1</property>
      <property name="version.minor">-1</property>
      <property name="caseNumber">7305</property>
      <property name="mainCaseNumber">7305</property>
      <property name="mainProcName">BEORDFUL</property>
    </properties>
  </system>
  <fields />
</orderFulfilment>
<generatedPlan>
  <lastUpdateVersionNumber>2</lastUpdateVersionNumber>

```

```

<primaryKey>
<id>1000578</id>
</primaryKey>
<executionPlanHeader>
<uniqueID>AOPDOrder-1172507110483</uniqueID>
<originator>AOPDTest</originator>
<masterRecordID>ORD1000314</masterRecordID>
<planCreationDate>2012-02-26T16:25:10</planCreationDate>
<status>PENDING</status>
<statusChanged>2012-02-26T16:25:10</statusChanged>
<description>Plan Generated from AOPDOrder</description>
<fulfilmentProc>
<procName>TBD</procName>
<description>Fulfilment proc</description>
<system>
<id>iProcess_11.1</id>
<properties />
</system>
</fulfilmentProc>
<schedule>
<jeopardy>
<startDate>2012-02-26T16:25:10</startDate>
<endDate>2012-02-26T16:31:10</endDate>
<projectedDuration>
<typical>360000</typical>
<maxAllowed>660000</maxAllowed>
</projectedDuration>
<designTimeDuration>
<typical>360000</typical>
<maxAllowed>660000</maxAllowed>
</designTimeDuration>
<jeopardyState>GREEN</jeopardyState>
</jeopardy>
</schedule>
</executionPlanHeader>
<executionPlanItems>
<summary>
<taskId>ProvideRouter</taskId>
<description>Supply ADSL2 108MBps Wireless Router</description>
<status>PENDING</status>
<statusChanged>2012-02-26T16:25:10</statusChanged>
<UDFs>
<UDF>
<name>itemSpecId</name>
<value>Router</value>
</UDF>
<UDF>
<name>action</name>
<value>PROVIDE</value>
</UDF>
</UDFs>

```

```

    <orderLineIds>
    <orderLineId>A</orderLineId>
    </orderLineIds>
    <executionPlanItems>
    <task>
    <taskId>ProvideRouter</taskId>
    <description>Supply ADSL2 108mbps Wireless Router</description>
    <status>PENDING</status>
    <statusChanged>2012-02-26T16:25:10</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <orderLineIds>
    <orderLineId>A</orderLineId>
    </orderLineIds>
    <executionPlanItems>
    <task>
    <taskId>ProvideRouter.1</taskId>
    <description>Task1</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iProcess_11.1</id>
    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
    </system>
    </procInfo>

```

```

<startMilestone>
<description>Start Task</description>
<id>START</id>
<status>PENDING</status>
  <properties>
<property name="stepType" />
<property name="notifyOnly" />
  </properties>
<point>
<dependencies />
<releaseMillis>0</releaseMillis>
  </point>
<fields>
<field io="OUT" name="itemSpecId">Router</field>
<field io="OUT" name="action">PROVIDE</field>
  </fields>
</startMilestone>
  <endMilestone>
<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
  <properties>
<property name="stepType" />
<property name="notifyOnly" />
  </properties>
<point>
<dependencies />
<releaseMillis>0</releaseMillis>
  </point>
<fields />
</endMilestone>
  <sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
  <schedule>
<typicalTiming>
<duration>60000</duration>
  </typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
  </maxAllowedTiming>
</schedule>
</section>
</sections>
  <schedule>
<typicalTiming>
<duration>60000</duration>
  </typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
  </maxAllowedTiming>
</schedule>
</processVersion>
</task>

```



```

    <task>
    <taskId>ProvideRouter.2</taskId>
    <description>Task2</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>Router</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iPROCESS_11.1</id>
    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
    </system>
    </procInfo>
    <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>
    <point>
    <dependencies>
    <dependency>
    <point>
    <timeDelta>0</timeDelta>
    <taskId>ProvideRouter.1</taskId>
    <type>START_TO_FINISH</type>
    <milestoneId>END</milestoneId>
    </point>
    </dependency>
    </dependencies>
    <releaseMillis>0</releaseMillis>
    </point>

```

```

    <fields>
    <field io="OUT" name="itemSpecId">Router</field>
    <field io="OUT" name="action">PROVIDE</field>
    </fields>
  </startMilestone>
  <endMilestone>
  <description>Task Complete</description>
  <id>END</id>
  <status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
  <dependencies />
  <releaseMillis>0</releaseMillis>
  </point>
  <fields />
</endMilestone>
  <sections>
    <section>
      <startMilestoneId>START</startMilestoneId>
      <finishMilestoneId>END</finishMilestoneId>
      <schedule>
        <typicalTiming>
          <duration>60000</duration>
        </typicalTiming>
        <maxAllowedTiming>
          <duration>120000</duration>
        </maxAllowedTiming>
      </schedule>
    </section>
  </sections>
  <schedule>
    <typicalTiming>
      <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>120000</duration>
    </maxAllowedTiming>
  </schedule>
</processVersion>
</task>
<task>
  <taskId>ProvideRouter.3</taskId>
  <description>Task3</description>
  <status>PENDING</status>
  <statusChanged>2012-02-07T14:58:53</statusChanged>
  <UDFs>
    <UDF>
      <name>itemSpecId</name>
      <value>Router</value>
    </UDF>
    <UDF>
      <name>action</name>
      <value>PROVIDE</value>
    </UDF>
  </UDFs>

```

```

    </UDF>
</UDFs>
<objectXrefs />
<itemId />
<action>PROVIDE</action>
<uniqueCompID>IT3STP</uniqueCompID>
  <processVersion>
  <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
      <id>iPROCESS_11.1</id>
      <properties>
        <property name="majorVersion">-1</property>
        <property name="minorVersion">-1</property>
        <property isEditable="true" name="exceptionHandler" />
        <property name="caseNumber">0</property>
      </properties>
    </system>
  </procInfo>
    <startMilestone>
      <description>Start Task</description>
      <id>START</id>
      <status>PENDING</status>
      <properties>
        <property name="stepType" />
        <property name="notifyOnly" />
      </properties>
      <point>
        <dependencies>
          <dependency>
            <point>
              <timeDelta>0</timeDelta>
              <taskId>ProvideRouter.2</taskId>
              <type>START_TO_FINISH</type>
              <milestoneId>END</milestoneId>
            </point>
          </dependency>
        </dependencies>
        <releaseMillis>0</releaseMillis>
      </point>
      <fields>
        <field io="OUT" name="itemSpecId">Router</field>
        <field io="OUT" name="action">PROVIDE</field>
      </fields>
    </startMilestone>
    <endMilestone>
      <description>Task Complete</description>
      <id>END</id>
      <status>PENDING</status>
      <properties>
        <property name="stepType" />
        <property name="notifyOnly" />
      </properties>
      <point>
        <dependencies />
        <releaseMillis>0</releaseMillis>

```

```

</point>
<fields />
</endMilestone>
  <sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
  <schedule>
<typicalTiming>
<duration>60000</duration>
  </typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
  </maxAllowedTiming>
</schedule>
  </section>
</sections>
  <schedule>
<typicalTiming>
<duration>60000</duration>
  </typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
  </maxAllowedTiming>
</schedule>
</processVersion>
</task>
<task>
<taskId>ProvideRouter.4</taskId>
<description>Task4</description>
<status>PENDING</status>
<statusChanged>2012-02-07T14:58:53</statusChanged>
  <UDFs>
<UDF>
<name>itemSpecId</name>
<value>Router</value>
  </UDF>
<UDF>
<name>action</name>
<value>PROVIDE</value>
  </UDF>
</UDFs>
<objectXrefs />
<itemId />
<action>PROVIDE</action>
<uniqueCompID>IT3STP</uniqueCompID>
  <processVersion>
<procInfo>
<procName>TBD</procName>
<description>TBD</description>
  <system>
<id>iPROCESS_11.1</id>
  <properties>

```

```

<property name="majorVersion">-1</property>
<property name="minorVersion">-1</property>
<property isEditable="true" name="exceptionHandler" />
<property name="caseNumber">0</property>
</properties>
</system>
  </procInfo>
<startMilestone>
<description>Start Task</description>
<id>START</id>
<status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
  <fields>
<field io="OUT" name="itemSpecId">Router</field>
<field io="OUT" name="action">PROVIDE</field>
</fields>
  </startMilestone>
<endMilestone>
<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
<fields />
  </endMilestone>
<sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
  <schedule>
<typicalTiming>
<duration>180000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
  </typicalTiming>
<maxAllowedTiming>
<duration>360000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
  </maxAllowedTiming>

```

```

<actualDuration>0</actualDuration>
</schedule>
</section>
</sections>
<schedule>
<typicalTiming>
<duration>180000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
</typicalTiming>
<maxAllowedTiming>
<duration>360000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
</maxAllowedTiming>
<actualDuration>0</actualDuration>
</schedule>
<isScheduleOverride>true</isScheduleOverride>
</processVersion>
</task>
</executionPlanItems>
<startPoint />
<schedule>
<typicalTiming>
<duration>180000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
</typicalTiming>
<maxAllowedTiming>
<duration>360000</duration>
<earlyStart>0</earlyStart>
<lateStart>0</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
</maxAllowedTiming>
</schedule>
<planFragmentUniqueId>ROUTER_PROVIDE</planFragmentUniqueId>
</summary>
<summary>
<taskId>ProvideStaticIP</taskId>
<description>Provision Static IP</description>
<status>PENDING</status>
<statusChanged>2012-02-26T16:25:10</statusChanged>
<UDFs>
<UDF>
<name>itemSpecId</name>
<value>StaticIP</value>
</UDF>
<UDF>
<name>action</name>
<value>PROVIDE</value>
</UDF>
</UDFs>

```

```

    <orderLineIds>
    <orderLineId>B</orderLineId>
    </orderLineIds>
    <executionPlanItems>
    <task>
    <taskId>ProvideStaticIP.1</taskId>
    <description>Task1</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>StaticIP</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>
    <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
    <id>iPROCESS_11.1</id>
    <properties>
    <property name="majorVersion">-1</property>
    <property name="minorVersion">-1</property>
    <property isEditable="true" name="exceptionHandler" />
    <property name="caseNumber">0</property>
    </properties>
    </system>
    </procInfo>
    <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>
    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields>
    <field io="OUT" name="itemSpecId">StaticIP</field>
    <field io="OUT" name="action">PROVIDE</field>
    </fields>
    </startMilestone>
    <endMilestone>

```

```

<description>Task Complete</description>
<id>END</id>
<status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
<dependencies />
<releaseMillis>0</releaseMillis>
</point>
<fields />
</endMilestone>
  <sections>
<section>
<startMilestoneId>START</startMilestoneId>
<finishMilestoneId>END</finishMilestoneId>
  <schedule>
<typicalTiming>
<duration>60000</duration>
  </typicalTiming>
<maxAllowedTiming>
<duration>120000</duration>
  </maxAllowedTiming>
</schedule>
</section>
</sections>
  <schedule>
<typicalTiming>
<duration>60000</duration>
</typicalTiming>
  <maxAllowedTiming>
<duration>120000</duration>
</maxAllowedTiming>
</schedule>
</processVersion>
</task>
  <task>
<taskId>ProvideStaticIP.2</taskId>
<description>Task2</description>
<status>PENDING</status>
<statusChanged>2012-02-07T14:58:53</statusChanged>
  <UDFs>
<UDF>
<name>itemSpecId</name>
<value>StaticIP</value>
  </UDF>
<UDF>
<name>action</name>
<value>PROVIDE</value>
</UDF>
  </UDFs>
  <objectXrefs />
  <itemId />

```



```

    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
  </processVersion>
  <procInfo>
    <procName>TBD</procName>
    <description>TBD</description>
    <system>
      <id>iPROCESS_11.1</id>
      <properties>
        <property name="majorVersion">-1</property>
        <property name="minorVersion">-1</property>
        <property isEditable="true" name="exceptionHandler" />
        <property name="caseNumber">0</property>
      </properties>
    </system>
  </procInfo>
  <startMilestone>
    <description>Start Task</description>
    <id>START</id>
    <status>PENDING</status>
    <properties>
      <property name="stepType" />
      <property name="notifyOnly" />
    </properties>
    <point>
      <dependencies>
        <dependency>
          <point>
            <timeDelta>0</timeDelta>
            <taskId>ProvideStaticIP.1</taskId>
            <type>START_TO_FINISH</type>
            <milestoneId>END</milestoneId>
          </point>
        </dependency>
      </dependencies>
      <point>
        <timeDelta>0</timeDelta>
        <taskId>ProvideStaticIP.3</taskId>
        <type>START_TO_FINISH</type>
        <milestoneId>END</milestoneId>
      </point>
    </point>
    <releaseMillis>0</releaseMillis>
  </point>
  <fields>
    <field io="OUT" name="itemSpecId">StaticIP</field>
    <field io="OUT" name="action">PROVIDE</field>
  </fields>
</startMilestone>
<endMilestone>
  <description>Task Complete</description>
  <id>END</id>
  <status>PENDING</status>

```

```

    <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
    </properties>
    <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
    </point>
    <fields />
    </endMilestone>
    <sections>
    <section>
    <startMilestoneId>START</startMilestoneId>
    <finishMilestoneId>END</finishMilestoneId>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </section>
    </sections>
    <schedule>
    <typicalTiming>
    <duration>60000</duration>
    </typicalTiming>
    <maxAllowedTiming>
    <duration>120000</duration>
    </maxAllowedTiming>
    </schedule>
    </processVersion>
    </task>
    <task>
    <taskId>ProvideStaticIP.3</taskId>
    <description>Task3</description>
    <status>PENDING</status>
    <statusChanged>2012-02-07T14:58:53</statusChanged>
    <UDFs>
    <UDF>
    <name>itemSpecId</name>
    <value>StaticIP</value>
    </UDF>
    <UDF>
    <name>action</name>
    <value>PROVIDE</value>
    </UDF>
    </UDFs>
    <objectXrefs />
    <itemId />
    <action>PROVIDE</action>
    <uniqueCompID>IT3STP</uniqueCompID>
    <processVersion>

```

```

<procInfo>
  <procName>TBD</procName>
  <description>TBD</description>
  <system>
    <id>iPROCESS_11.1</id>
    <properties>
      <property name="majorVersion">-1</property>
      <property name="minorVersion">-1</property>
      <property isEditable="true" name="exceptionHandler" />
      <property name="caseNumber">0</property>
    </properties>
  </system>
</procInfo>
<startMilestone>
  <description>Start Task</description>
  <id>START</id>
  <status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
  </point>
  <fields>
    <field io="OUT" name="itemSpecId">StaticIP</field>
    <field io="OUT" name="action">PROVIDE</field>
  </fields>
</startMilestone>
<endMilestone>
  <description>Task Complete</description>
  <id>END</id>
  <status>PENDING</status>
  <properties>
    <property name="stepType" />
    <property name="notifyOnly" />
  </properties>
  <point>
    <dependencies />
    <releaseMillis>0</releaseMillis>
  </point>
  <fields />
</endMilestone>
<sections>
  <section>
    <startMilestoneId>START</startMilestoneId>
    <finishMilestoneId>END</finishMilestoneId>
    <schedule>
      <typicalTiming>
        <duration>120000</duration>
        <earlyStart>180000</earlyStart>
        <lateStart>180000</lateStart>
        <isOnCriticalPath>true</isOnCriticalPath>
      </typicalTiming>
    </schedule>
  </section>
</sections>

```

```

<maxAllowedTiming>
<duration>180000</duration>
<earlyStart>360000</earlyStart>
<lateStart>360000</lateStart>
<isOnCriticalPath>true</isOnCriticalPath>
  </maxAllowedTiming>
  <actualDuration>0</actualDuration>
</schedule>
</section>
</sections>
  <schedule>
    <typicalTiming>
      <duration>120000</duration>
      <earlyStart>180000</earlyStart>
      <lateStart>180000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>180000</duration>
      <earlyStart>360000</earlyStart>
      <lateStart>360000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </maxAllowedTiming>
    <actualDuration>0</actualDuration>
  </schedule>
  <isScheduleOverride>true</isScheduleOverride>
</processVersion>
</task>
</executionPlanItems>
<startPoint>
  <dependencies>
    <dependency>
      <point>
        <timeDelta>0</timeDelta>
        <taskId>ProvideRouter</taskId>
        <type>FINISH_TO_START</type>
      </point>
    </dependency>
  </dependencies>
</startPoint>
  <schedule>
    <typicalTiming>
      <duration>180000</duration>
      <earlyStart>180000</earlyStart>
      <lateStart>180000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </typicalTiming>
    <maxAllowedTiming>
      <duration>300000</duration>
      <earlyStart>360000</earlyStart>
      <lateStart>360000</lateStart>
      <isOnCriticalPath>true</isOnCriticalPath>
    </maxAllowedTiming>
  </schedule>

```

```
<planFragmentUniqueId>STATICIP_PROVIDE</planFragmentUniqueId>  
  </summary>  
  </executionPlanItems>  
</generatedPlan>  
<amendment>true</amendment>  
</detail>  
</aopdResponse>
```


Appendix F **Sample JNDI Properties File**

This appendix provides an example of a JNDI properties file for AOPD.

Topics

- [JNDI Properties File, page 92](#)

JNDI Properties File

Below is an example of a JNDI properties file for AOPD.

You should remove the comment markers in the sections that apply to the type of application server you are using, and modify the text as necessary.

```
#####
##### Uncomment and Modify for WebLogic App Server #####
#####

# Context factory: Requires weblogic.jar in the classpath
#java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory

# Provider URL: t3://wl_hostname:wl_port/
#java.naming.provider.url=t3://hostname:8001

# Principal: username|ipe_hostname|ipe_nodename|
# Note:
# 1. The trailing slash is required
# 2. The username should belong to the COMUsers group
#java.naming.security.principal=swadmin|ipe_hostname|ipe_node|

# Credentials: The password for the username supplied above
#java.naming.security.credentials=password

#####
##### Uncomment and Modify for JBOSS App Server #####
#####

# Context factory: Requires jbossall-client.jar in the classpath
#java.naming.factory.initial=org.jboss.security.jndi.LoginInitialContextFactory

# Provider URL: jnp://jboss_hostname:jboss_port/
#java.naming.provider.url=jnp://hostname:1099

# Principal: username|ipe_hostname|ipe_nodename|
# Note:
# 1. The trailing slash is required
# 2. The username should belong to the COMUsers group
#java.naming.security.principal=swadmin|ipe_hostname|ipe_node|

# Credentials: The password for the username supplied above
#java.naming.security.credentials=password
```


Index

A

argument, parameter, and option [27, 31, 35, 43, 47, 91](#)

C

changes from the previous release [vi](#)

customer support [xii](#)

I

installation directory

BEA_HOME [ix, ix](#)

iProcessConductorDir [ix](#)

iProcessConductorDomain [ix](#)

JAVA_HOME [ix](#)

JBOSS_HOME [ix](#)

ORACLE_HOME [ix](#)

WL_HOME [x](#)

S

support, contacting [xii](#)

T

technical support [xii](#)

terminology conventions [27, 31, 35, 43, 47, 91](#)