TIBCO iProcess® Conductor

Utility Framework

User's Guide

Software Release 11.2.1
August 2012

two-second advantage™

TIBCO®
The Power of Now®

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

Figures iii

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 TIBCOmmunity xii
- How to Access TIBCO Documentation xii
- How to Contact TIBCO Support xii

Chapter 1 Introduction 1

Product Overview 2

Chapter 2 Configuring the Utility Framework 3

Files 4
- Upgrading from Previous Versions 4

Properties 5
- Server Properties 5
- Database Server Properties 7
- Utility Properties 7

Chapter 3 Export and Import 13

Exporting Plans and Process Components 14

Importing Plans and Process Components 16

Chapter 4 Deletion 23

Deleting Plans and Process Components 24

Chapter 5 Error Codes 27

Error Codes 28

iProcess Conductor Utility General Warnings 28
Server Warnings 29
Import Warnings 30
Export Warnings 31
Delete Warnings 33

Index 35

Figures

Figure 1 Password prompt 14
Figure 2 Exporting plans 15
Figure 3 Password prompt 16
Figure 4 Prompting for invalid start date 17
Figure 5 Entering a start date 18
Figure 6 Renaming a component 21
Figure 7 Renaming an execution plan 21
Figure 8 Two versions of an execution plan 22

Preface

This document describes TIBCO iProcess Conductor Utility Framework. TIBCO iProcess Conductor now supports a framework through which utility applications can be plugged in to provide added functionality.

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
iProcessConductor Dir	The directory where you install TIBCO iProcess Conductor. For example: • on Windows systems, `c:\Program Files\iProcessConductor`. • on UNIX systems, `/opt/comDomain/iProcessConductor`.
iProcessConductor Domain	The target directory where you install the TIBCO iProcess Conductor domain. For example: • on Windows systems, *BEA_HOME*`\user_projects\domains\iPCDomain`; or *JBOSS_HOME*`\server\`*server_type*`\deploy` under JBoss. • on UNIX systems, *BEA_HOME*`/user_projects/domains/iPCDomain`. or *JBOSS_HOME*`/server/`*server_type*`/deploy` under JBoss.
ORACLE_HOME	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.
JAVA_HOME	The pathname to the directory where you install Java. For example: • on Windows systems, `C:\Program Files\Java\jdk1.6.0_21`. • on UNIX systems, `/opt/Java/jdk1.6.0_21`. **Note:** When you install the product on AIX platforms: — set *JAVA_HOME* as the pathname to the directory of Java incorporated by iProcess Engine 11.3.1 if you want to work with iProcess Engine 11.3.1 — install JRE 1.6.0 IBM J9 2.4 AIX SR6, and set *JAVA_HOME* as the pathname to the installation directory of JRE 1.6.0 IBM J9 2.4 AIX SR6 if you want to work with iProcess Engine versions earlier than 11.3.1
If you are using JBoss:	
JBOSS_HOME	The pathname to the home directory of your JBoss installation. For example: • on Windows systems, `c:\jboss\jboss-4.2.1`. • on UNIX systems, `/opt/JBoss/JBoss-4.2.1`.

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 *TIBCO ActiveMatrix BusinessWorks Concepts*. To introduce new terms. For example: A portal page may contain several portlets. *Portlets* 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` *PathName*
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 TIBCOmmunity

TIBCOmmunity 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. TIBCOmmunity 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 Utility Framework.

Topics

- Product Overview, page 2

Product Overview

iProcess Conductor now supports a framework through which utility applications can be plugged in. These utilities can then interact with the iProcess Conductor server and provide added functionality.

The iProcess Conductor Utility Framework must be run on the same server as the iProcess Conductor Server.

This document primarily aims at explaining the Import, Export and Delete utilities and the configuration of the iProcess Conductor Utility Framework. Usage of this framework to create new utility plug-ins is beyond the scope of this document.

Chapter 2 Configuring the Utility Framework

This chapter describes how to configure the iProcess Conductor Utility Framework.

Topics

- Files, page 4
- Properties, page 5

Files

When you install the iProcess Conductor, the files necessary to use the Utility Framework are installed into the directory that you selected during the installation process. See *TIBCO iProcess Conductor Installation* for further details.

Once installation is complete, you should have the directory structure shown below. The example shown assumes that your chosen installation directory is `D:\TIBCO\iProcessConductor\IPCUtilities`:

```
D:\TIBCO\iProcessConductor\IPCUtilities
    IPCUtil.cmd
    IPCUtil.sh
    IPCUtil.xml

    lib
        bootstrap.jar
        lib.jar
```

If you are installing the iProcess Conductor Utility Framework in a cluster:

- In an Oracle WebLogic cluster, install iProcess Conductor Utility Framework on the WebLogic administration server.
- In a JBoss cluster, install iProcess Conductor Utility Framework at every server node.

See *TIBCO iProcess Conductor Installation* for details of installing iProcess Conductor in a cluster.

Upgrading from Previous Versions

If you are upgrading from a previous version of TIBCO iProcess Conductor Utility Framework, the configuration information that you specified in the `iPCUtil.properties` file is lost if you install iProcess Conductor in the same location as the previous version. You must edit the new `iPCUtil.xml` file to specify this information. However, many default values are specified automatically to reduce the editing time required.

TIBCO therefore recommends that, unless you are installing iProcess Conductor in a different location, you take a copy of the `iPCUtil.properties` file, so that you have a record of your configuration data that you can copy to the new `iPCUtil.xml` file.

Properties

The sample property file `IPCUtil.xml` contains the properties necessary to configure the iProcess Conductor Utility Framework.

The properties are divided into two main categories:

- Server properties
- Specific utility properties. The `mode` property, described in Utility Properties on page 7, determines which utility is to be executed, which in turn loads further properties specific to the utility.

Note that:

- The iProcess Conductor Utility Framework needs to connect to either a JBoss or a WebLogic application server. The server needs to be running for it execute successfully.
- Some of the properties set file paths. These should be set in UNIX style, with forward slashes (/) irrespective of the operating system. For example, `e:/dir1/dir2` instead of `e:\dir1\dir2`. Using back slashes will cause the utility to fail.
- Properties starting with (#) are considered as comments and will be ignored.
- Values for many of these properties are automatically filled in from information entered during iProcess Conductor installation.

Server Properties

The table below describes the server properties and their values.

Property	Value
server.target	This property defines the target application server. This can have the value of: • `weblogic9` when WebLogic version 9 is used • `weblogic10` when WebLogic version 10 is used • `jboss` when JBoss is used. **Note:** The property value must appear in lower case.

Property	Value
server.home	The home directory of the application server. This should be set to: • *WL_HOME* in the case of WebLogic (for example, `E:/bea/weblogic92`) • *JBOSS_HOME* in the case of JBoss (for example, `F:/jboss-4.2.1`)
server.domain	The directory where the iProcess Conductor is deployed on the application server. For example, if the application is deployed in WebLogic the property might be set to `E:/bea/user_projects/domains/IPCGA` and for JBoss the property might be set to `F:/jboss-4.2.1/server/default`.

server.url	This is the connection URL for the application server.
	For WebLogic this must be in the format:
	t3://*hostname*:*weblogic_port*
	for example, t3://*hostname*:8001; or, in a WebLogic cluster:
	t3://*host1*:*port1*,*host2*:*port2*...
	For JBoss this must be in the format:
	jnp://*hostname*:*jboss_port*
	for example, jnp:// *hostname*:1099; or, in a JBoss cluster:
	jnp://*host1*:*port1*,jnp://*host2*:*port2*,...

server.principal	The login details for the application server. This should be in the format: *username* | *Computer_Name* | *iProcess_host_name* | *hostname* | *TCP_port_number* | N | 1 **Note:** Ensure that the trailing pipe (|) is present. If it is not, this will cause the utility to fail.
server.credentials	This property is optional. It is the password for the login details supplied in `server.principal`. If security is a concern, this can be commented out (with a leading **#** character). The utility will then prompt you for the password when needed.

Database Server Properties

The table below describes the database server properties and their values. Specifying these properties enables the delete utility to connect to the iProcess Conductor database.

Property	Value
ipc.db.user	The username of the iProcess Conductor database user. For example, `ipc.db.user=ipcuserAA01`
ipc.db.password	This property is optional. It is the password to log into the iProcess Conductor database server. If security is a concern, this can be commented out (with a leading # character). The utility will then prompt you for the password when needed.
ipc.db.host	The hostname of the iProcess Conductor database server. For example, `ipc.db.host=conductordbhostname`.
ipc.db.port	This property is optional. The iProcess Conductor database server port. If this is not specified, it defaults to `1521`. For example, `ipc.db.port=1521`.
ipc.db.sid	The iProcess Conductor database server SID. For example, `ipc.db.sid=orcl2`.

Utility Properties

The `mode` setting determines which utility is to be executed. This in turn loads properties specific to that utility.

Property	Value
mode	The utility that should be executed. Values currently supported are: • export • import • delete

The table below describes the settings for the export utility.

Property	Value

export.plans	The plans to be exported are to be listed here separated by a semicolon (;). For example:
	`plan1;plan2;plan3`
	Plans have unique IDs. These are case-sensitive, so if for example 'Plan1' is listed instead of 'plan1', an error will be generated and export for this plan will be skipped.
	In addition, plan IDs cannot contain spaces. Leading and trailing white spaces may be used for better legibility, but they are automatically trimmed.
	Exported execution plans will be saved with the prefix `EP-`, and exported process components with the prefix `PC-`. For example:
	`EP-plan1.xml`
	`PC-proc1.xml`
	However when importing these plans and process components to another environment, you do not need to specify the prefix.
	Instead of entering a string of plan IDs, if all the plans in the iProcess Conductor database are to be exported, you can use the asterisk (*) wildcard. For example:
	`export.plans=*`
	This wildcard should be used on its own.

export.output	The output directory where the exported files are to be stored. Note that the directory will not be created. You will need to ensure that the directory exists and has write permissions. You can use a period (.) for the current directory.

The table below describes the settings for the import utility.

Property	Value
import.input	The directory from where the exported plans and process components are to be read.

Property	Value

import.plans	The plans to be imported. Multiple execution plans should be separated by a semicolon (;). For example:
	`plan1;plan2;plan3`
	Note that the supporting process components that were exported by the export utility should also be present in this directory. If the process components cannot be imported, the plan import will be skipped.
	Do not modify these exported files or their contents. You can enter the plan names without the `EP-` prefix that was given to them at export. For example, specifying `plan1` will import `EP-plan1.xml`.
	If all plans in the directory specified by `import.input` are to be imported, then you can use the asterisk (*) wildcard. For example:
	`import.plans=*`
	This wildcard should be used on its own.

import.conflict.action	This property is optional. Execution plans or Process components with name conflicts (that is, execution plans or process components with names similar to ones that already exist in the iProcess Conductor database) can be resolved with this property. If this property is included, the action specified will be considered as the default action and the user will not be prompted. To select a different action for each execution plan or process component with name conflicts, comment this property. Valid values can be: • `Skip` • `Rename` • `Auto Rename` • `Abandon`

Property	Value
import.pc. startdate.current	This property is optional. Process components with historic start dates will need to have valid start dates at the time of import. Valid start dates are current date/time or later. If this property is set to **`true`** the user will not be prompted for a date and the current server date and time will be used. If this property is set to `false`, the user will have a choice of either using the current server date and time or defining a custom date when a process component with a historic start date is encountered.

The table below describes the settings for the delete utility.

Property	Value
delete.orderrefs	This property is optional. The order reference of the order to be deleted. Multiple orders should be separated by a semicolon (**;**). For example: `orderref1;orderref2;orderref3` If no value is specified, no orders are deleted. You can use the asterisk (*****) as a wildcard. For example `delete.orderrefs=*` deletes all orders; `delete.orderrefs=order*` deletes all orders whose reference starts with the characters "order".
delete.order.status	You can use this property to limit deletion to orders of a specified status. You can specify more than one status value, separated by semicolons. For example: `delete.order.status=COMPLETE;ORDER SUSPENDED` If this value is blank or the asterisk (*) wildcard is used, all orders are deleted regardless of status.
delete.order.status Changed.before	You can use this property to limit deletion to orders that have changed status before a particular date and time. For example: `delete.order.statusChanged.before=2008-06-20T07:30:00` If no value is specified, deletion is not limited.

Property	Value
delete.plans	This property is optional. The unique ID of the execution plan to be deleted. Multiple plans should be separated by a semicolon (;). For example: `delete.plans=plan1;plan2` If no value is specified, no plans are deleted. You can use the asterisk (*) as a wildcard. For example `delete.plans=*` deletes all plans; `delete.plans=plan*` deletes all plans that start with the characters "plan".
delete.plans.status	You can use this property to limit deletion to plans of a specified status. You can specify more than one status value, separated by semicolons. For example: `delete.plans.status=DRAFT;TEMPLATE` If this value is blank or the asterisk (*) wildcard is used, all plans are deleted regardless of status.
delete.plans.statusChanged.before	You can use this property to limit deletion to execution plans that have changed status before a particular date and time. For example: `delete.plans.statusChanged.before=2008-06-20T07:30:00` If no value is specified, deletion is not limited.
delete.pcs	This property is optional. The unique ID of the process component to be deleted. Multiple components should be separated by a semicolon (;). For example: `delete.pcs=pc1034;pc1035` If no value is specified, no components are deleted. You can use the asterisk (*) as a wildcard. For example `delete.pcs=*` deletes all components; `delete.pcs=pc*` deletes all components that start with the characters "pc".
delete.pcs.createdDate.before	You can use this property to limit deletion to process components that were created before a particular date and time. For example: `delete.pcs.createdDate.before=2008-01-01T05:00:00` If no value is specified, deletion is not limited.

Chapter 3 Export and Import

This chapter describes how to export and import plans and process components.

Topics

- Exporting Plans and Process Components, page 14
- Importing Plans and Process Components, page 16

Exporting Plans and Process Components

Only plans of Draft, Pending or Template status can be exported. If you attempt to export a plan in any other state, a warning will be logged to this effect and the plan will be skipped. For an introduction to plan status, see *TIBCO iProcess Conductor Concepts*.

To export plans and process components, follow the procedure below.

1. Start the iProcess Conductor server. For details of how to do this, see *TIBCO iProcess Conductor User's Guide*.
2. Ensure that the `mode` property in the property file is set to `export`. Also, ensure that the `export.plans` and `export.output` properties have valid values.
3. To run the utility on Windows, open a command prompt and navigate to the directory where you have installed the iProcess Conductor Utility Framework. At the prompt, type the following and press *Enter*:

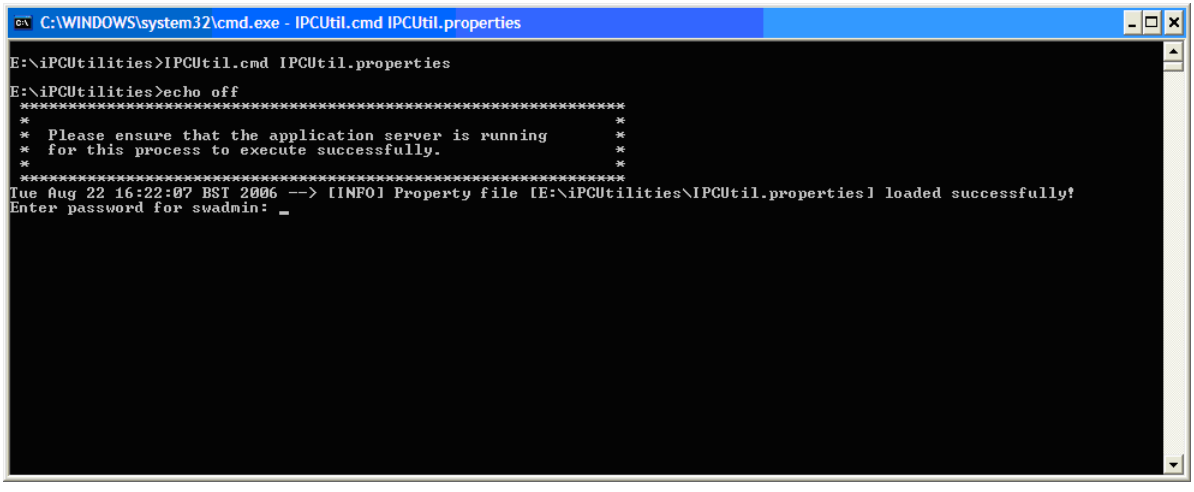
   ```
   E:\iPCUtilities>IPCUtil IPCUtil.xml
   ```

   To run the utility on UNIX, run the following command from the directory where you have installed the iProcess Conductor Utility Framework:

   ```
   ./IPCUtil.sh IPCUtil.xml
   ```
4. If a value has not been specified for the `server.credentials` property in the property file, you will be prompted to enter the password, as follows:

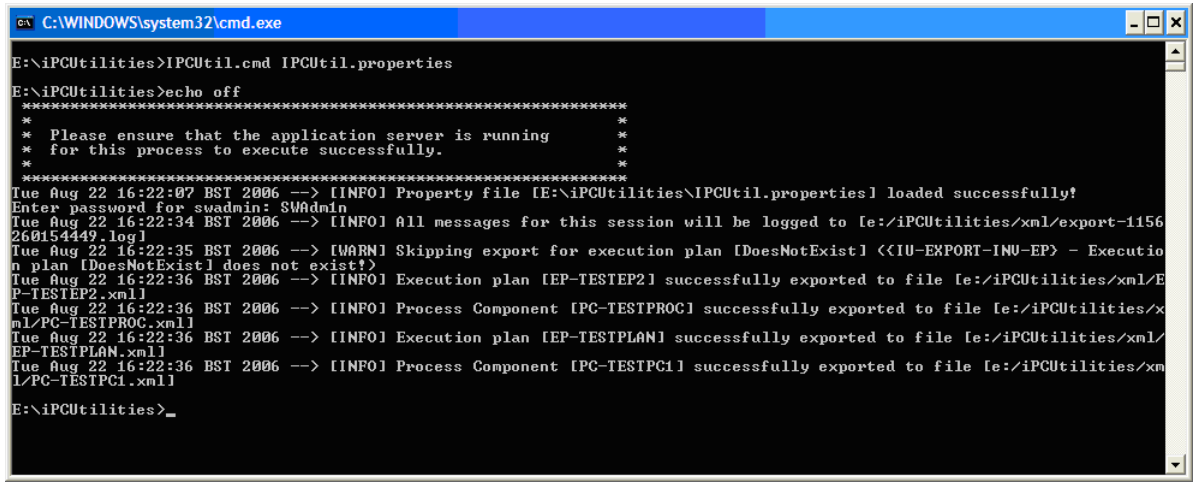
   ```
   Enter password for swadmin:
   ```

Figure 1 Password prompt



5. The utility should now have exported the plans and the supporting process components to the directory specified in the `export.output` property. In the example shown in Figure 2, notice that a warning was generated for plan `[DoesNotExist]`.

Figure 2 Exporting plans



6. All messages and warnings are logged in the directory specified in the `export.output` property with the filename `export-`*unique_number*`.log`. This log file name will be mentioned at the start of the utility. Examine the log files for any warnings. All warnings will be associated with an error code. Check Chapter 5, Error Codes, on page 27 for a detailed explanation.

Importing Plans and Process Components

Only plans of Draft, Pending or Template status can be imported. If you attempt to import a plan in any other state, a warning will be logged to this effect and the plan will be skipped. For an introduction to plan status, see *TIBCO iProcess Conductor Concepts*.

To import plans and process components, follow the procedure below.

1. Start the iProcess Conductor server. For details of how to do this, see *TIBCO iProcess Conductor User's Guide*.
2. Ensure that the `mode` property in the property file is set to `import`. Also, ensure that the `import.plans` and `import.input` properties have valid values.
3. To run the utility on Windows, open a command prompt and navigate to the directory where you have installed the iProcess Conductor Utility Framework. At the prompt, type the following and press *Enter*:

   ```
   E:\iPCUtilities>IPCUtil IPCUtil.xml
   ```

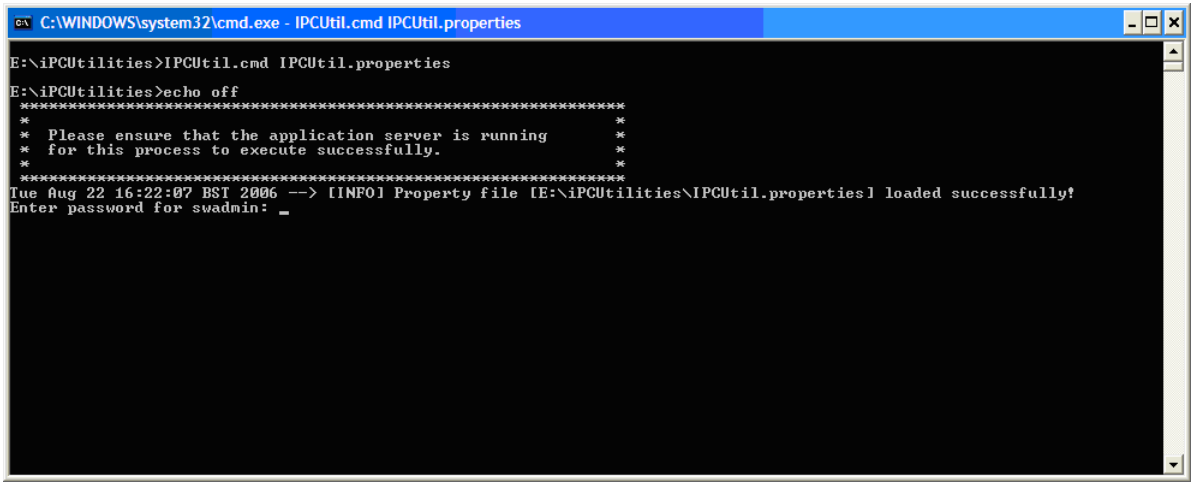
   To run the utility on UNIX, run the following command from the directory where you have installed the iProcess Conductor Utility Framework:

   ```
   ./IPCUtil.sh IPCUtil.xml
   ```

4. If a value has not been specified for the `server.credentials` property in the property file, you will be prompted to enter the password, as follows:

   ```
   Enter password for swadmin:
   ```

Figure 3 Password prompt



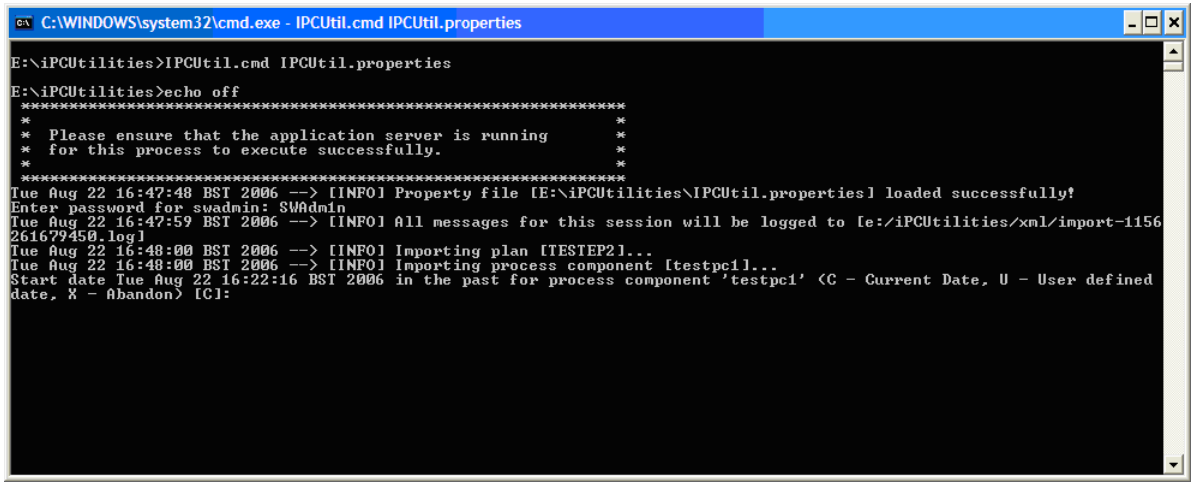
5. The process components for each plan should be present in the same directory as the plans (as specified by the `import.input` property). Failure to find the process component will result in the plan import being skipped.
6. Process component versions should have a valid start date at the time of import. This should be either the current server time or later. If the start date is invalid, the user will be prompted with three choices.

Choice	Shortcut	Description
Current date	C	This will use the current server date and time.

User defined	U	The user will be prompted for a new date and time as follows:
		`Date format yyyy-MM-dd'T'hh:mm:ss`
		`For example 2006-10-13T12:07:22`
		`Enter start date:`
		The date should be entered in the format *yyyy-MM-dd'T'hh:mm:ss*
		For example:
		`2009-12-25T12:07:22`
		Notice the literal '**T**' which is used to separate the date and time. If the date is entered incorrectly the user will be prompted again.

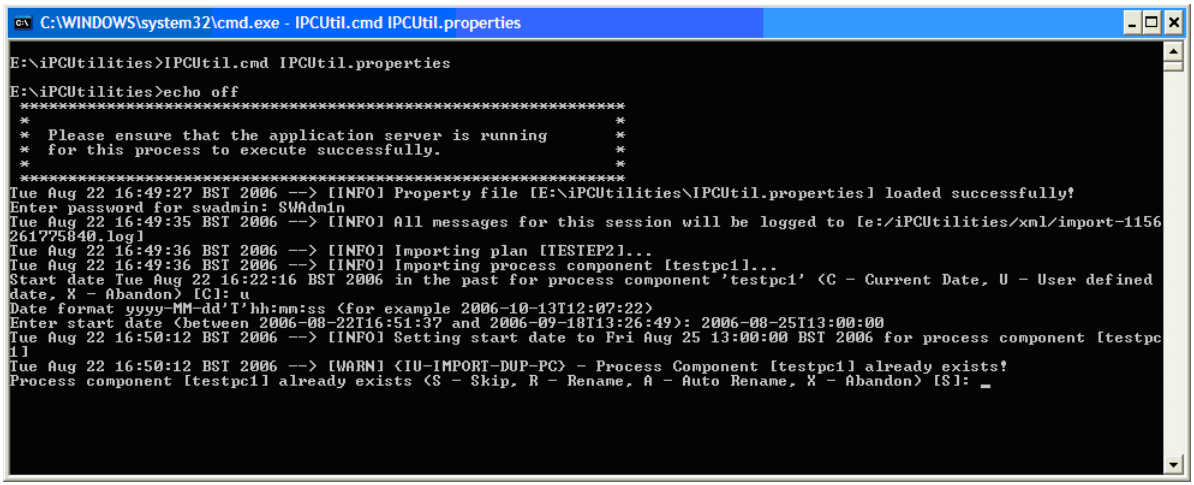
Abandon	X	This means the entire import utility will be abandoned. All plans and process components imported prior to abandoning the process will remain in the iProcess Conductor database.

Figure 4 Prompting for invalid start date



7. In the example shown in Figure 5, the start date is entered as 2006-08-25T13:00:00 which means that the process component version will now be valid from 1:00p.m. on the 25th August, 2006. Since this will be a regular occurrence, the user can set the property `import.pc.startdate.current=true`. This will use the current server date and time every time this problem is encountered. Refer to Chapter 2, Configuring the Utility Framework, on page 3 for more information.

Figure 5 Entering a start date



Note that:

— End dates for process component versions should not be historic and will be skipped. A warning will be logged to this effect.

— If a process component version has an historic start date and a valid end date, the new date should be between the current server time and the end date. Both dates will be displayed to the user as shown below (and as also shown in the example in Figure 5):

```
Date format yyyy-MM-dd'T'hh:mm:ss (for example
2006-10-13T12:07:22)
Enter start date (between 2006-08-11T10:22:07 and
2006-09-09T14:15:41):
```

— If all process component versions have historic end dates, the process component and the execution plan will be skipped.

8. Process components and execution plans should have unique names within the same iProcess Conductor instance. If an execution plan or process component is being imported with a name that already exists in the iProcess

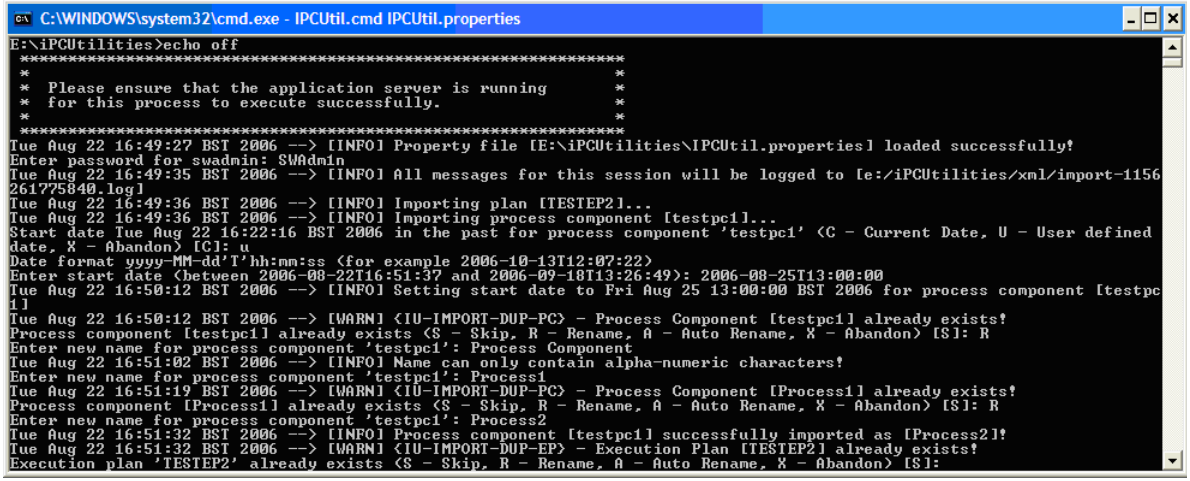
Conductor database, the user is prompted and given four choices. These are explained in the following table.

Choice	Shortcut	Description
Skip	S	Skips importing the current object into the iProcess Conductor database. This is the default option. If a process component is being imported, this means that the component will be skipped and the import utility will proceed to the next process component referenced by the plan. If an execution plan is being imported, this means that the plan will be skipped and the import utility will proceed to the next plan defined in the list specified by the `import.plans` property. Note that: • Using Skip implies that a process component with the same name and signature exists in the iProcess Conductor database. If this is not the case, the plan import will fail since the plan requires all referenced process components to exist in the database before it can be imported. • In the case of plans, any process components that have been imported will remain in the iProcess Conductor database.

Choice	Shortcut	Description
Rename	R	The Rename option gives the user an opportunity to enter a new name for this process component or execution plan. If this option is chosen, the user is presented with a prompt to enter a new name: `Enter new name for process component 'testpc1':` The name entered should be unique. If this is not the case, the user will again be prompted with the four options. Valid names: • Are a maximum of 40 characters in length. • Are case-sensitive. That is, 'Testpc1' is considered different from 'testpc1'. It will be considered a unique name and will be accepted (if it does not already exist). • Should only contain alphanumeric characters (a-z, A-Z, 0-9). For example, the names 'Test Process' or 'Test_Process' are invalid. The user will be warned and prompted to enter a valid name. • Can be entered with leading or trailing spaces; but these will be trimmed before the name is used. For example, ' TestProcess ' will be trimmed and used as 'TestProcess'.
Auto Rename	A	This is similar to the Rename function above. The only difference is that the process component is renamed automatically. This is usually done by appending '_*x*' (where *x* is a number) to the process component name. In the example above, the process component will be renamed automatically as: `testpc1_1, testpc1_2, testpc1_3…` and so on till a name is found that does not exist. The name for the new process component will be logged once the execution plan or process component is imported.
Abandon	X	This means the entire import utility will be abandoned. All plans and process components imported prior to abandoning the process will remain in the iProcess Conductor database.

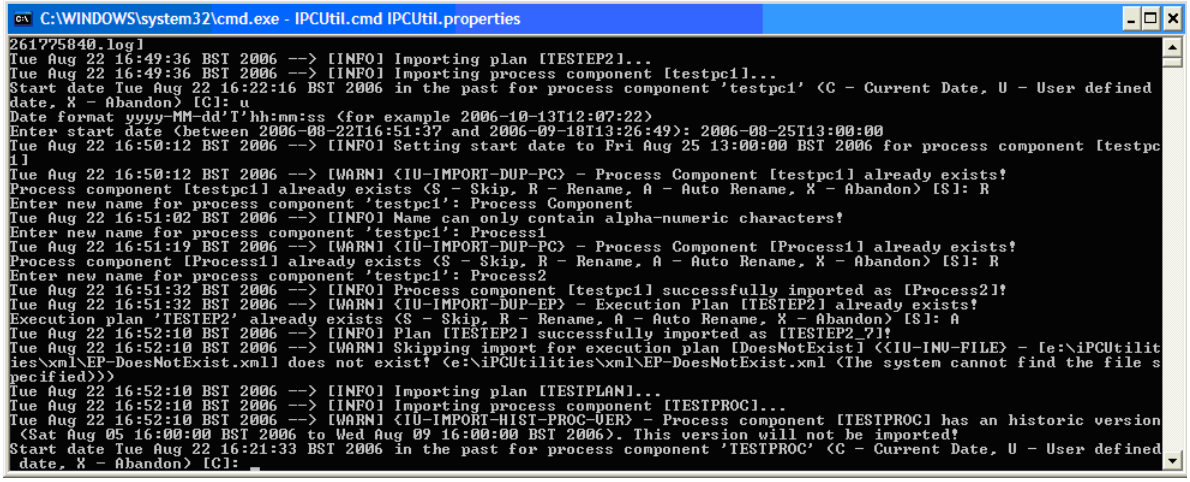
9. In the example shown in Figure 6, we have chosen to rename the process component 'testpc1' to 'Process2'. Notice the name 'Process Component' was rejected since it contained white spaces.

Figure 6 Renaming a component



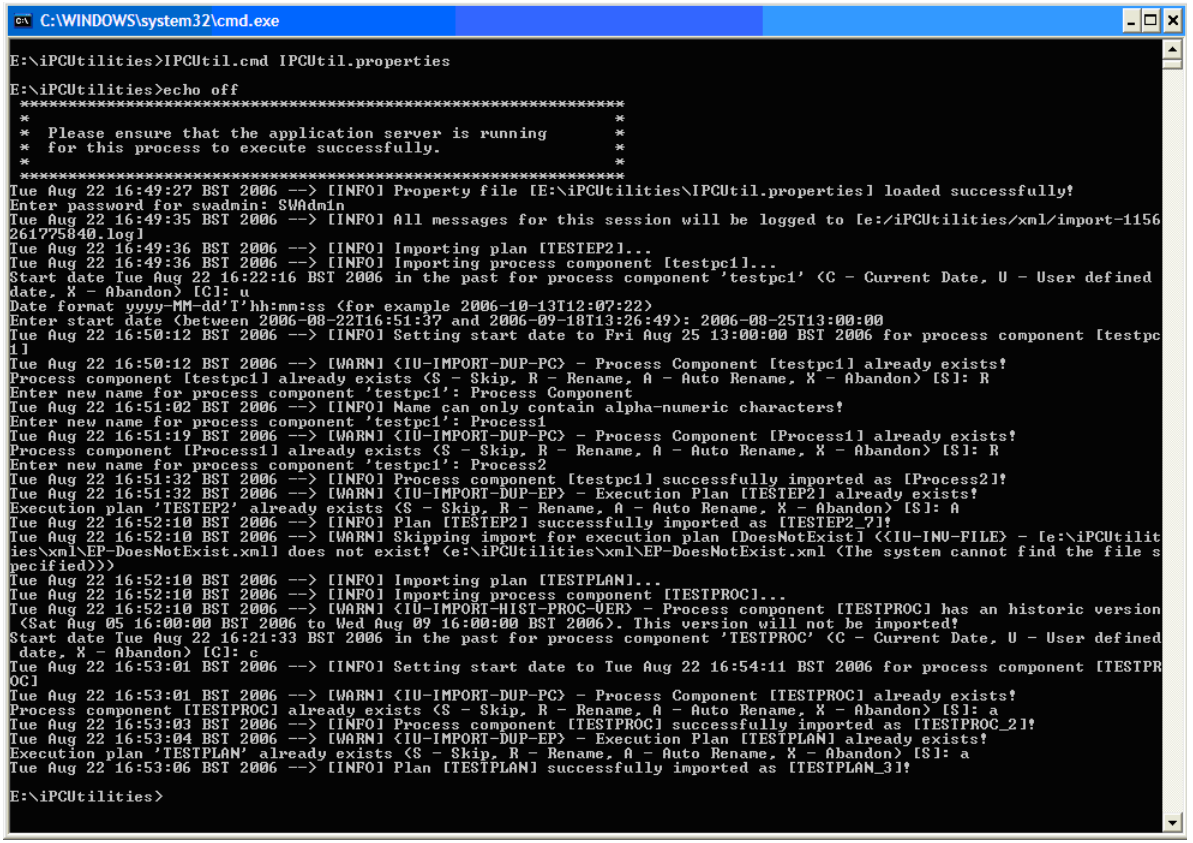
10. Like process components, execution plans also need to have unique names. If an execution plan is being imported with the same name as one that exists in the iProcess Conductor database, the user is presented with the same choices as that for a process component. In the example shown in Figure 7, we have chosen to Auto Rename the execution plan. The message 'Plan [TESTEP2] successfully imported as [TESTEP2_7]!' will be displayed once a unique name is found for the execution plan.

Figure 7 Renaming an execution plan



11. In the example shown in figure 8, [TESPROC] has an historic version and has been skipped. However, since it has two versions, the second version is imported with the current start date.

Figure 8 Two versions of an execution plan



12. All messages and warnings are logged in the directory specified in the `import.input` property with the filename `import-`*unique_number*`.log`. This log file name will be mentioned at the start of the utility. Examine the log files for any warnings. All warnings will be associated with an error code. Check Chapter 5, Error Codes, on page 27 for a detailed explanation.

Chapter 4 Deletion

This chapter describes how to delete plans and process components.

Topics

- Deleting Plans and Process Components, page 24

Deleting Plans and Process Components

You can use the Delete utility to delete orders, execution plans and process components. You can delete orders and plans of any status. For an introduction to plan status, see *TIBCO iProcess Conductor Concepts*.

The types of object that you can delete may have dependencies on each other. The delete utility therefore checks for any dependencies, and handles them as follows:

- Deleting an order automatically deletes any dependent execution plans.
- An execution plan *cannot* be deleted if there are any orders dependent on it.
- A process component *cannot* be deleted if there are any execution plans dependent on it.

To delete plans and process components, follow the procedure below.

1. Start the iProcess Conductor server. For details of how to do this, see *TIBCO iProcess Conductor User's Guide*.
2. Ensure that the `mode` property in the property file is set to `delete`.
3. The three properties that specify whether items will be deleted - `delete.orderrefs`, `delete.plans` and `delete.pcs` - are all optional. See **Utility Properties on page 7**. Ensure that these properties have the values you require.
4. The other properties listed in **Utility Properties on page 7** can be used to specify the range of items that will be deleted, and therefore to limit the scope of the deletion. Ensure that they also have the values you require.

Be careful in specifying and checking these values. Entering the wrong criteria can result in items being deleted from the iProcess Conductor database in error.

5. To run the utility on Windows, open a command prompt and navigate to the directory where you have installed the iProcess Conductor Utility Framework. At the prompt, type the following and press *Enter*:

   ```
   E:\iPCUtilities>IPCUtil IPCUtil.xml
   ```

   To run the utility on UNIX, run the following command from the directory where you have installed the iProcess Conductor Utility Framework:

   ```
   ./IPCUtil.sh IPCUtil.xml
   ```

6. If a value has not been specified for the `ipc.db.password` property in the property file, you will be prompted to enter the database password, as follows:

```
Enter password for ipcuseAA01:
```

7. Sometimes the `IPCUtil` program will access the iProcess Conductor server, and if a value has not been specified for the `server.credentials` property in the property file, you will also be prompted to enter that password (see Figure 3, Password prompt, on page 16,) as follows:

   ```
   Enter password for swadmin:
   ```

   Otherwise, you will not be prompted for this password.

8. The utility now deletes the specified orders, plans, and/or process components.

9. All messages and warnings are logged in the directory from which you run the delete utility, with the filename `delete-`*`unique_number`*`.log`. This log file name will be mentioned at the start of the utility. Examine the log files for any warnings. All warnings will be associated with an error code. Check Chapter 5, Error Codes, on page 27 for a detailed explanation.

Chapter 5 Error Codes

This chapter describes the iProcess Conductor Utility error codes.

Topics

- Error Codes, page 28

Error Codes

The iProcess Conductor utility framework makes it easy to identify warnings. All warnings that are logged contain a unique error code and will appear in curly braces; for example {IU-XXX-XXX}. Where a specific cause for the error can be identified, it will be printed in brackets after the error message text. For example:

```
{IU-FILE-CREATE} - Cannot create file [/xml/test.xml]
(/xml/test.xml (Permission denied))
```

The tables below list the available error codes, the reasons for their occurrence and the actions you could take to correct them.

iProcess Conductor Utility General Warnings

The following table lists general warnings.

Error Code	Description and Reason	Corrective Action
IU-INV-FILE	`[xxx] does not exist!` The file mentioned in square brackets cannot be found, or the user does not have read privileges for it.	Check the path to check for typographical errors. The `server.home` and `server.domain` properties are used by the utilities to load jar files into the classpath, and hence require the paths to point to the existing iProcess Conductor installation. Ensure these paths are valid and that the file is present. Ensure that the directory containing file [*xxx*] has read permissions.
IU-INV-PROP	`Property [xxx] not defined or invalid!` The property mentioned in square brackets is not defined in the property file.	Check the property file for typographical errors. Ensure that the correct property file is used. The full path is printed when the utility executes. Ensure that a valid value is defined. Empty properties are considered as invalid for mandatory properties.

Error Code	Description and Reason	Corrective Action
IU-INV-APP	`Application directory not found or invalid!` The iProcess Conductor application was not found within the domain specified in the `server.domain` property.	Ensure that the path specified in the `server.domain` directory is valid and points to the current active server installation.
IU-FILE-CREATE	`Cannot create file [`*xxx*`]` Cannot create the file in the directory specified.	Log files are created in the directories specified in the `export.output` and `import.input` properties, in addition to the exported files. Ensure that the directory (*xxx*) specified has write permissions. Ensure that the drive has enough space to export data to the directory specified. Check the actual cause specified within brackets for more information.

Server Warnings

The following table lists server warnings.

Error Code	Description and Reason	Corrective Action
IU-SVR	`iProcess Conductor Server cannot be contacted! Please check if the server is running and try again!` The iProcess Conductor Server cannot be contacted.	Ensure that the server defined in the `server.target` is running. Verify that the URL specified in `server.url` is valid. WebLogic servers require the prefix t3:// and JBoss servers require the prefix jnp://. Ensure that the `server.principal` is defined and is valid. If the `server.credentials` property is specified in the property file, ensure it is correct. For the utility to prompt for the password the credentials should be commented out, as an empty property will be considered as a blank password.

Error Code	Description and Reason	Corrective Action
IU-SVR-INVOKE	`iProcess Conductor Server Exception! Cannot xxx` An error has occurred while invoking the iProcess Conductor APIs. The action defined in *xxx* specifies the action that has been invoked on the iProcess Conductor server.	The iProcess Conductor Utility Framework is designed to work with the latest version of the iProcess Conductor. Ensure that the correct versions are installed.

Import Warnings

The following table lists import utility warnings.

Error Code	Description and Reason	Corrective Action
IU-IMPORT-INV-EXPORT FILE	`File [xxx] does not exist or is not a valid Execution plan or process component` The file *xxx* is not in a valid format or does not exist.	Check if the file path is correct. Ensure that the exported files have not been modified manually. If the problem persists, try exporting the plans again. This should overwrite these files.
IU-IMPORT-HIST-PROC-VER	`Process component 'xxx' has an historic version (<startdate> to <enddate>). This version will not be imported!` The process component has an historic version. Historic versions cannot be imported.	Create a new version for this process component, re-export the process component, and then retry the import Ensure that the end date/time is greater than the current date/time.
IU-IMPORT-INV-PROC-VERS	`No valid process versions for process component [xxx]` The process component being imported has no valid process versions.	Create a new version for this process component, re-export the process component, and then retry the import. Ensure that the end date/time is greater than the current date/time.

Error Code	Description and Reason	Corrective Action
IU-IMPORT-DUP-EP	`Execution Plan [xxx] already exists!` The plan named *xxx* already exists in the iProcess Conductor database.	Refer to Importing Plans and Process Components on page 16 for the various choices available.
IU-IMPORT-DUP-PC	`Process Component [xxx] already exists!` The process component named *xxx* already exists in the iProcess Conductor database.	Refer to Importing Plans and Process Components on page 16 for the various choices available.

Export Warnings

The following table lists export utility warnings.

Error Code	Description and Reason	Corrective Action
IU-INV-PLAN-STATE	`Execution plan [xxx] is in the [xxx] state! Plan status should be either Draft, Pending or Template` The execution plan that is currently being imported or exported is in an invalid state.	Ensure that the correct plan is specified in the `export.plans` and `import.plans` properties. The plans should be in Draft, Pending or Template state.
IU-EXPORT-INV-EP	`Execution plan [xxx] does not exist!` The execution plan cannot be found in the iProcess Conductor database.	Check that the `server.url` property is pointing to the correct iProcess Conductor instance (in case of multiple iProcess Conductor instances). Check the plan specified in (*xxx*) for typographical errors.
IU-EXPORT-INV-PC	`Process Component [xxx] does not exist!` The process component cannot be found in the iProcess Conductor database.	This is only likely to happen if the process component has been deleted manually from the iProcess Conductor database and is now in an inconsistent state. Contact the system administrator.

Error Code	Description and Reason	Corrective Action
IU-EXPORT-PC	`Cannot export process component [xxx] to file [xxx]` The process component cannot be exported to the specified file.	A fatal error has occurred while exporting the process component. Check the actual cause specified within brackets for more information.
IU-EXPORT-EP	`Cannot export execution plan [xxx] to file [xxx]` The execution plan cannot be exported to the specified file.	A fatal error has occurred while exporting the execution plan. Check the actual cause specified within brackets for more information.

Delete Warnings

The following table lists delete utility warnings.

Error Code	Description and Reason	Corrective Action
IU-DEL-DRIVER	`Cannot load driver [<driver>]`	Check that the `server.target` and `server.home` properties are correct.
IU-DEL-CON	`Cannot initiate database connection! [<connection url>]`	Check that the `ipc.db.user`, `ipc.db.host`, `ipc.db.port`, and `ipc.db.sid` properties are correct. Make sure you have input the correct database password when prompted by `IPCUtil`. Check the actual cause specified within brackets for more information.
IU-DEL-AUD-FETCH	`Error while fetching audit data for <object unique id> ( <object id> )` A fatal error has occurred while deleting the audit logs.	Check the actual cause specified within brackets for more information.
IU-DEL-AUD	`Error while deleting audit data for <object unique id> ( <object id> )` A fatal error has occurred while deleting the audit logs.	Check the actual cause specified within brackets for more information.
IU-DEL-ORD-FETCH	`Error while fetching order data for [<order ref>] (<order id>)` A fatal error has occurred while deleting the order.	Check the actual cause specified within brackets for more information.
IU-DEL-ORD	`Error while cleaning up order [<order ref>] (<order id>)` A fatal error has occurred while deleting the order.	Check the actual cause specified within brackets for more information.

Error Code	Description and Reason	Corrective Action
IU-DEL-PLAN	`Error while cleaning up plan [<plan unique id>] (<plan id>)` A fatal error has occurred while deleting the plan.	Check the actual cause specified within brackets for more information.
IU-DEL-PC	`Error while cleaning up process component [<pc unique id>](<pc id>)` A fatal error has occurred while deleting the process component.	Check the actual cause specified within brackets for more information.

Index

C

changes from the previous release vi
Configuring the iProcess Conductor Utility Framework 3
customer support xii

D

Database server properties 7
delete.order.status 10
delete.order.statusChanged.befor 10
delete.orderrefs 10
delete.pcs 11
delete.pcs.createdDate.before 11
delete.plans 11
delete.plans.status 11
delete.plans.statusChanged.before 11
Duplicated names 18

E

Error codes 27
  delete 33
  export 31
  format of messages 28
  general 28
  import 30
  server 29
export.output 8
export.plans 8
Exporting Plans and Process Components 14, 24

F

files installed 4

I

import.conflict.action 9
import.input 8
import.pc.startdate.current 10
import.plans 9
Importing Plans and Process Components 16
installation directory
  BEA_HOME ix, ix
  iProcessConductorDir ix
  iProcessConductorDomain ix
  JAVA_HOME ix
  JBOSS_HOME ix
  ORACLE_HOME ix
  WL_HOME x
IPCUtil.xml 5

N

Names 18

P

Password 14, 16, 24, 25
Plan status 14, 16

S

Server properties 5
server.credentials 6
server.domain 6
server.home 6
server.principal 6
server.target 5
server.url 6
Start dates 17
Status 14, 16
support, contacting xii

T

technical support xii

U

Utility Properties 7

V

Valid start date 17
Versions 22