

# TIBCO iProcess<sup>®</sup> swutil and swbatch

## Reference Guide

*Software Release 11.4.1  
April 2014*

## **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, Two-Second Advantage, TIBCO ActiveMatrix BusinessWorks, TIBCO Business Studio, TIBCO Enterprise Message Service, TIBCO Hawk, TIBCO iProcess, TIBCO iProcess Suite, and TIBCO Rendezvous are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation 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 © 1994-2014 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

<b>Tables</b> .....	<b>vii</b>
<b>Preface</b> .....	<b>ix</b>
Related Documentation .....	x
TIBCO iProcess Engine Documentation .....	x
Other TIBCO Product Documentation .....	x
Typographical Conventions .....	xii
Connecting with TIBCO Resources .....	xvii
How to Join TIBCOCommunity .....	xvii
How to Access TIBCO Documentation .....	xvii
How to Contact TIBCO Support .....	xvii
<b>Chapter 1 Using the swutil Utility</b> .....	<b>1</b>
Using swutil .....	2
Running swutil Commands as Different Users .....	2
<b>Chapter 2 Cases and Case Data</b> .....	<b>5</b>
Starting a Case .....	6
Extracting Case Data (Interactive) .....	8
Extracting Case Data (Extended) .....	9
Output Formats .....	10
Closing Cases .....	12
Purging Cases .....	13
Updating Prediction for Cases .....	14
<b>Chapter 3 Work Items</b> .....	<b>15</b>
Resending Work Items .....	16
Clearing Locked Work Queue Items .....	17
Checking Work Queue Items .....	19
Example .....	19
Output of the Command .....	20
Removing Work Queue Items .....	21
Example .....	21
Output of the Command .....	22

<b>Chapter 4 Events</b> .....	<b>23</b>
Issue an Event .....	24
<b>Chapter 5 Activity Monitoring</b> .....	<b>27</b>
Overview .....	28
Exporting Activity Monitoring Configuration Information .....	29
Importing Activity Monitoring Configuration Information .....	31
<b>Chapter 6 Audit Trails</b> .....	<b>33</b>
Create a User-Defined Audit Trail Entry .....	34
Defining Audit Trail Entries .....	34
<b>Chapter 7 Case Data Queue Parameters</b> .....	<b>37</b>
Overview of Case Data Queue Parameters (CDQPs) .....	38
Importing CDQP Configuration Data .....	39
CDQP Control File .....	41
Resetting CDQP Configuration Data .....	41
Defining CDQPs .....	41
Example CDQP Definitions .....	43
Mapping CDQPs to Queues .....	43
Default User and Group Queue Mappings .....	44
Example Queue Mappings .....	45
CDQP Command Errors .....	47
Exporting CDQP Configuration Data .....	50
Export File Format .....	50
Export File Examples .....	51
swutil QINFO Exit Codes and Error Messages .....	54
<b>Chapter 8 Procedures and Procedure Libraries</b> .....	<b>57</b>
Overview of Importing/Exporting Procedures and Libraries .....	58
Import a Procedure .....	59
Import a Procedure Interactively .....	59
Import Procedures Using Command Line Options .....	63
Import a Procedure Library .....	67
Import a Procedure Library Interactively .....	67
Import a Procedure Library Using Command Line Options .....	69
Display Commit/Rollback Messages When Importing Procedures or Procedure Libraries .....	72
Export a Procedure .....	73
Export a Procedure Interactively .....	73

Export Procedures Using Command Line Options . . . . .	76
Export a Procedure Library . . . . .	79
Export a Procedure Library Interactively . . . . .	79
Export a Procedure Library Using Command Line Options . . . . .	80
<b>Chapter 9 Procedure Versions . . . . .</b>	<b>83</b>
Release a Procedure Version . . . . .	84
Withdraw a Procedure Version . . . . .	86
Delete a Procedure Version . . . . .	87
swutil PROC command errors . . . . .	88
<b>Chapter 10 Users and Groups . . . . .</b>	<b>89</b>
Command Line Invocation . . . . .	90
Control File Format . . . . .	91
Control Instructions . . . . .	92
Reset . . . . .	92
Change Attribute . . . . .	92
Change User . . . . .	93
Change Group . . . . .	94
Change Attribute Value . . . . .	94
Change Membership . . . . .	95
Export Format . . . . .	96
Security . . . . .	98
<b>Chapter 11 Roles . . . . .</b>	<b>99</b>
Command Line Invocation . . . . .	100
Control File Format . . . . .	101
Control Instructions . . . . .	102
Example Control File . . . . .	103
<b>Chapter 12 Queue Participation . . . . .</b>	<b>105</b>
Command Line Invocation . . . . .	106
Control File Format . . . . .	107
Control Instructions . . . . .	108
Reset . . . . .	108
Reset Queue Participants . . . . .	108
Add Participants . . . . .	108
Example Control File . . . . .	110

<b>Chapter 13 Queue Redirection</b>	<b>111</b>
Command Line Invocation	112
Control File Format	113
Control Instructions	114
Reset	114
Reset Queue Redirection	114
Redirect Queue	114
Example Control File	115
<b>Chapter 14 Lists</b>	<b>117</b>
Import a List	118
Export a List	119
<b>Chapter 15 iProcess Tables</b>	<b>121</b>
Extract a Table	122
Delete a Table	123
Import an iProcess Table	124
Export a Table	125
<b>Chapter 16 Moving System Information</b>	<b>127</b>
Move System Information	128
<b>Chapter 17 Batch Processing Cases</b>	<b>129</b>
Overview	130
Running swbatch	131
Command File Format	131
swbatch Commands	132
Start a Case	133
Issue an Event	136
Close a Case	138
Purge a Case	139
<b>Index</b>	<b>141</b>

# Tables

Table 1	General Typographical Conventions . . . . .	xii
Table 2	Syntax Typographical Conventions . . . . .	xiii
Table 2	Syntax Typographical Conventions . . . . .	xiv
Table 2	Syntax Typographical Conventions . . . . .	xv
Table 3	swutil Commands. . . . .	2
Table 4	CDQP Command Errors. . . . .	47
Table 5	swutil QINFO Exit Codes and Error Messages . . . . .	54
Table 6	Procedure Release Status in .xfr and Available Import Options . . . . .	62
Table 7	Procedure Release Status on Target System and Required Conditions for Import. . . . .	62
Table 8	swutil IMPORT Options . . . . .	63
Table 9	swutil IMPORTLIB options . . . . .	69
Table 10	Release Status Options of the Procedure Export Command. . . . .	74
Table 11	swutil EXPORT Options. . . . .	76
Table 12	swutil EXPORTLIB Options . . . . .	80
Table 13	swutil PROC Error Codes. . . . .	88
Table 14	swutil ROLEINFO Control Instructions. . . . .	102



# Preface

This guide provides a complete reference to the *SWDIR/bin/swutil* and *SWDIR/bin/swbatch* utilities. It should be used in conjunction with *TIBCO iProcess Engine Administrator's Guide*.



A different utility, also called *swutil*, is available on TIBCO iProcess<sup>®</sup> Workspace (Windows). See *TIBCO iProcess Workspace (Windows) User's Guide* for more information.

## Topics

---

- [Related Documentation, page x](#)
- [Typographical Conventions, page xii](#)
- [Connecting with TIBCO Resources, page xvii](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO iProcess Engine Documentation

The following documents form the TIBCO iProcess Engine documentation set:

- *TIBCO iProcess Engine Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO iProcess Engine 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 Suite Documentation This documentation set contains all the manuals for TIBCO iProcess Engine and other TIBCO products in TIBCO iProcess<sup>®</sup> Suite. The manuals for TIBCO iProcess Engine are as follows:
  - *TIBCO iProcess Engine Architecture Guide*
  - TIBCO iProcess Engine Administrator's Guides:
    - TIBCO iProcess Engine Administrator's Guide*
    - TIBCO iProcess Objects Director Administrator's Guide*
    - TIBCO iProcess Objects Server Administrator's Guide*
  - TIBCO iProcess Engine Database Administrator's Guides:
    - TIBCO iProcess Engine (DB2) Administrator's Guide*
    - TIBCO iProcess Engine (Oracle) Administrator's Guide*
    - TIBCO iProcess Engine (SQL) Administrator's Guide*
  - *TIBCO iProcess swutil and swbatch Reference Guide*
  - *TIBCO iProcess Engine System Messages Guide*
  - *TIBCO iProcess User Validation API User's Guide*
  - *LDAPCONF Utility User's Guide*

### Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO ActiveMatrix BusinessWorks<sup>™</sup>

- TIBCO Business Studio™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous®

## Typographical Conventions

---

TIBCO iProcess Engine can be run on both Microsoft Windows and UNIX/Linux platforms. In this manual, the Windows convention of a backslash (\) is used. The equivalent pathname on a UNIX or Linux system is the same, but using the forward slash (/) as a separator character.



UNIX or Linux pathnames are occasionally shown explicitly, using forward slashes as separators, where a UNIX or Linux-specific example or syntax is required.

Any references to UNIX in this manual also apply to Linux unless explicitly stated otherwise.

The following typographical conventions are used in this manual

Table 1 *General Typographical Conventions*

Convention	Use
<i>SWDIR</i>	<p>TIBCO iProcess Engine installs into a directory. This directory is referenced in documentation as <i>SWDIR</i>. The value of <i>SWDIR</i> depends on the operating system. For example,</p> <ul style="list-style-type: none"> <li>on a Windows server (on the C: drive)           <p>if <i>SWDIR</i> is set to the C:\swserver\staffw_nod1 directory, then the full path to the <code>swutil</code> command is in the C:\swserver\staffw_nod1\bin\swutil directory.</p> </li> <li>on a UNIX or Linux server           <p>if <i>SWDIR</i> is set to the /swserver/staffw_nod1 directory, then the full path to the <code>swutil</code> command is in the /swserver/staffw_nod1/bin/swutil directory or the <code>\$SWDIR/bin/swutil</code> directory.</p> <p><b>Note:</b> On a UNIX or Linux system, the environment variable <code>\$SWDIR</code> should be set to point to the iProcess system directory for the <i>root</i> and <i>swadmin</i> users.</p> </li> </ul>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use <code>MyCommand</code> to start the foo process.</p>

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<b>bold code font</b>	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none"> <li>• In procedures, to indicate what a user types. For example: Type <b>admin</b>.</li> <li>• In large code samples, to indicate the parts of the sample that are of particular interest.</li> <li>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [<b>enable</b>   disable]</li> </ul>
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> <li>• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.</li> <li>• To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.</li> <li>• To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i></li> </ul>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	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
[ ]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <p>MyCommand [optional_parameter] required_parameter</p>

Table 2 *Syntax Typographical Conventions (Cont'd)*

Convention	Use
	<p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <pre>MyCommand param1   param2   param3</pre>
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2}   {param3 param4}</pre> <p>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:</p> <pre>MyCommand {param1   param2} {param3   param4}</pre> <p>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.</p> <pre>MyCommand param1 [param2] {param3   param4}</pre>

## 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 **Using the swutil Utility**

This chapter explains how to use the `swutil` utility. This utility provides command line access to some of the common administration tasks that you might perform such as extracting case data, exporting a procedure or starting a procedure.

### Topics

---

- [Using swutil, page 2](#)

## Using swutil

You can use `swutil` either:

- interactively by using the `swutil` menu or by running the command at the command line and following the prompts, or
- (for import and export commands only) manually issue a command with command line options to automatically set import or export options and suppress the interactive prompts and questions. This method can be much quicker when working with many procedures. See [Procedures and Procedure Libraries on page 57](#).

If you invoke the `swutil` utility without any options, you are presented with the `swutil` menu that enables you to use `swutil` interactively.



Only the most common `swutil` commands are shown on the `swutil` menu. To see a complete listing of the utilities you can access directly, use the following syntax:

```
swutil ?
```

### Running swutil Commands as Different Users

Most functions require you to be logged in as either:

- On UNIX, the `IPEBACKGROUND` user or `IPEADMIN` user. The `root` user can run only a limited subset of commands.
- On Windows, the `IPEADMIN` user.

The following table also summarizes the commands you can use as each user.

Table 3 *swutil* Commands

Administrative area	swutil Command(s)	Can be run as...		
		<i>IPEADMIN</i> user	<i>IPEBACKGROUND</i> user	root user
Cases and case data	CSTART, DCSTART, EXTCD, CLOSE, PURGE, PREDICT	Yes	Yes, if the <i>IPEBACKGROUND</i> user account is also an iProcess user.	No
Work items	RESEND, UNLOCKMAIL, CHECKITEM, CLEANITEM	Yes	No	No

Table 3 *swutil Commands (Cont'd)*

Administrative area	swutil Command(s)	Can be run as...		
		<i>IPEADMIN</i> user	<i>IPEBACKGROUND</i> user	root user
Events	EVENT	Yes	No	No
Activity monitoring	EXPMONITOR, IMPMONITOR	Yes	No	No
Audit trails	AUDIT	Yes	No	No
CDQPs	QINFO	Yes	Yes	No
Procedures and libraries	IMPORT, IMPORTLIB, EXPORT, EXPORTLIB	Yes	Yes	Yes
Procedure versions	PROC	Yes	No	No
Users, groups and roles	USERINFO, ROLEINFO, MOVESYSINFO	Yes	Yes	Yes
Participation and redirection	PARTICIPATION, REDIRECTION	Yes	Yes	Yes
Lists	IMPORT, EXPLST	Yes	Yes	Yes
iProcess tables	EXTTAB, DELTAB, IMPORT, EXPTAB	Yes	Yes	Yes
Batch processing	swbatch	Yes	No	No



## Chapter 2 Cases and Case Data

This chapter describes the following `swutil` commands.

Command	Description
<code>swutil CSTART</code> <code>swutil DCSTART</code>	Starts a case
<code>swutil EXTCD</code>	Extracts case data
<code>swutil CLOSE</code>	Closes a case
<code>swutil PURGE</code>	Purges a case
<code>swutil PREDICT</code>	Updates cases for Prediction



To run any of these commands you must be logged in either as the *IPEADMIN* user or *IPEBACKGROUND* user; the *IPEBACKGROUND* user account must also be an *iProcess* user.

## Starting a Case

---

You can start a case of a procedure from the `swutil` menu (see [page 2](#)) using either of the following options:

- 8 - Start a Case (without supplying any data)
- 9 - Start a Case with Data

Alternatively, you can use the following commands:

- to start a case without data:

```
swutil CSTART [version] [-sN] [-c] [-V] procname casedesc [startstep] [-P
priority]
```

- to start a case with data:

```
swutil DCSTART [version] [-sN] [-c] [-V] procname casedesc datafile
[startstep] [-P priority]
```

where:

- *version* (optional) specifies the version of the procedure that you want to start a case of. Specify *one* of the following flags:

Flag	Description
-r	Start a (live) case of the Released version.
-u	Start a (test) case of the Unreleased version.
-m	Start a (test) case of the Model version.
-vX.Y	Start a case of Version X.Y, where X is the major version number and Y is the minor version number.

If *version* is not specified, the following default precedence order is used to select the version to use:

Released > Unreleased > Model

Using a Released version starts a live case. Using an Unreleased or Model version starts a test case.

- `-sN` (optional) specifies the precedence order to be used to select the version of any sub-procedure started by the case.

Value	Precedence
1	Released version only
2	Unreleased > Released version
3	Model > Released version
4	Unreleased > Model > Released version
5	Model > Unreleased > Released version

If the `-sN` flag is not specified, only Released versions are used.

Using a Released version starts a live sub-case. Using an Unreleased or Model version starts a test sub-case.



For more information about procedure versions, see "Using Version Control" in *TIBCO iProcess Modeler Procedure Management*.

- `-c` (optional) specifies that the command returns the case number of the started case.
- `-v` (optional) specifies verbose operation.
- `procname` is the name of the procedure that you want to start a case of.
- `casedesc` is the case description to use for the started case.
- `datafile` is the name of a file in abox format (See "Using iProcess Abox Files" in *TIBCO iProcess Modeler Integration Techniques*), containing data for field initialization.
- `startstep` (optional) specifies the stepname at which to start the case; if omitted the start step in the procedure definition is used.
- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.

## Extracting Case Data (Interactive)

---

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil EXTCD [-m]
```

where `-m` is an optional flag you can specify to extract memo data. If `-m` is specified, the path to the output file for the memo field is displayed. If not, the message `MEMO FIELD: NO OUTPUT` is displayed.

The utility prompts you for the host nodename, procedure name, case number and output choice (screen or file).

Here is an example of the output:

---

```
Case data for Procedure TESTPROC, Case #1 (First case)
```

```
Field nameField value
```

```
AFIELDHere's a text value  
DATE23/10/1997
```

---

## Extracting Case Data (Extended)

---

If you want to specify the output format of the case data, use the following command:

```
swutil EXTCD [flags] procname [casenum [fields]]
```

where:

- *flags* indicates one or more of the following optional parameters, preceded by a hyphen character (-), which can appear anywhere on the command line.

Parameter	Description
format	<p>One of the following flags, which specifies the output format for the extracted data. (The default is <b>-v</b> if no format flag is included.)</p> <ul style="list-style-type: none"> <li>-v verbose</li> <li>-a abox</li> <li>-c comma delimited</li> <li>-t tab delimited</li> <li>-t<i>char</i> delimited by character <i>char</i></li> </ul> <p>These formats are explained in <a href="#">Output Formats on page 10</a>.</p>
textmodifier	<p>One of the following flags, which specifies whether the text is written with DOS or UNIX newline conventions.</p> <ul style="list-style-type: none"> <li>-u UNIX: LF</li> <li>-d DOS: CR + LF</li> </ul>
-f	<p>Causes a list of field names to be included as the first line of the output. It only applies if the <b>-c</b>, <b>-t</b> or <b>-t<i>char</i></b> format options are specified. The field names are written in upper-case letters, separated as specified by the format flag.</p>

Examples of flag options:

```
-vu
-a -d
-dt~
-t
```

- *procname* is the name of the procedure (case insensitive).
- *casenum* is optional, and specifies the number of the case to be extracted, or one of the following may be used:

Value	Description
ALL_CASES	All cases for the procedure will be extracted.
ALL_ACTIVE	All active cases for the procedure.
ALL_TERM	All terminated cases for the procedure.

If this parameter is not specified, the default is ALL\_CASES.

- *fields* is optional (but requires that *casenum* is included), and provides a list of fields to be extracted, separated by commas, for example:

---

FIELD1 , FIELD2 , FIELD3

---

(Field names are case insensitive.)

As well as fields defined in the procedure, the following predefined fields may be included:

Value	Description
SW_CASENUM	specifies the case number
SW_CASEREF	specifies the case reference, <i>pp-nn</i>
SW_CASEDESC	specifies the case description

Alternatively, ALL\_FIELDS may be used to specify that all fields of the procedure should be extracted (including SW\_CASENUM, and SW\_CASEDESC if it is used as a marked field). This is the default if the parameter is omitted.

## Output Formats

### Verbose: -v

This gives the same output as `swutil EXTCD` without any extra parameters. See [Extracting Case Data \(Interactive\) on page 8](#).

**abox: -a**

This is formatted one field per line, each line consisting of the field name, followed by a comma, followed by the field data.

Cases are written out in numeric order; the order of fields is as specified on the command line. There is no break between different cases.

Examples of output:

---

```
SW_CASENUM,1
AFIELD,Here's a text value
DATE,23/03/1999
SW_CASENUM,2
AFIELD,Here's a name: "Peter"
DATE,24/03/1999
```

---

**Comma delimited: -c**

This is formatted one case per line, with each field value separated from the next by a comma.

Field names are written out on the first line if the `-f` flag is included. Cases are written out in numeric order; the order of fields is as specified on the command line.

If the field contains a comma as part of its value, the entire field contents is enclosed in double quotes; if a field contains double quotes, each double quotes character is represented by two double quotes characters, `" "`.

Examples of output for `-cf`:

---

```
SW_CASENUM,AFIELD,DATE
1,Here's a text value,23/03/1999
2,Here's a name: ""Peter"",24/03/1999
```

---

**Tab (or other character) delimited: -t**

This works the same as the comma-delimited format, except:

- The tab (or specified separator) is used between fields
- No double-quoting of fields is performed.

## Closing Cases

---

The following command instructs the iProcess background process to perform an instruction. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

If you want to close a large number of cases you should consider using the `swbatch` utility instead. See [Chapter 17, Batch Processing Cases, on page 129](#).

To close one or more cases of a procedure, enter the following command:

```
swutil CLOSE procname caselist [-y] [-d] [-P priority]
```

where:

- *procname* is the procedure name (including sub-procedures).
- *caselist* is a list of case numbers to be closed separated by commas and with dashes denoting ranges of case numbers, for example: 2, 5-10 denotes case 2 and cases 5 to 10 inclusive.
- `-y` suppresses the request for confirmation from the user.
- `-d` closes cases without triggering events that are set for the `OnBeforeClose` event or the `OnAfterClose` event. If you are running the `swutil CLOSE` command without the `-d` option, these events are triggered before or after closing the case. For more information about the `OnBeforeClose` event and the `OnAfterClose` event, see *TIBCO iProcess Modeler Procedure Management*.
- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.

## Purging Cases

---

The following command instructs the iProcess background process to purge one or more cases of a procedure. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

If you want to purge a large number of cases you should consider using the `swbatch` utility instead. See [Chapter 17, Batch Processing Cases, on page 129](#).

To purge one or more cases of a procedure, enter one of the following commands:

```
swutil PURGE procname caselist [-y] [-d] [-P priority]
swutil PURGE procname ALL_CASES [-y] [-d] [-P priority]
swutil PURGE procname dd_mm_yyyy [-y] [-d] [-P priority]
```

where:

- *procname* is the procedure name. If the named procedure has sub-procedures, the cases for the sub-procedures will also be purged.
- *caselist* is a list of case numbers to be purged separated by commas and with hyphens denoting ranges of case numbers, for example: 2, 5-10 denotes case 2 and cases 5 to 10 inclusive. All cases in the range are purged whether they are open or closed.  
  
ALL\_CASES denotes all cases of the procedure whether they are open or closed.
- *dd\_mm\_yyyy* purges all cases terminated up to and including a particular date, which is specified as the day, month and year numbers (in that order) separated by underscores.
- `-y` suppresses the request for confirmation from the user.
- `-d` purges cases without triggering events that are set for the `OnBeforePurge` event. If you are running the `swutil PURGE` command without the `-d` option, these events are triggered before purging the case. For more information about the `OnBeforePurge` event, see *TIBCO iProcess Modeler Procedure Management*.
- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.

## Updating Prediction for Cases

---

The following command instructs the iProcess background process to perform an instruction. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

If you enable prediction on a procedure and there are live cases for that procedure, you can choose whether to update some or all of your live cases for prediction. If you do not choose to update some or all of the live cases for prediction, the live cases are updated for prediction when there is activity on the cases. For example, when a step is released.

To update one or more cases of a procedure for prediction, enter one of the following commands:

```
swutil PREDICT procname case_list [-y]
swutil PREDICT procname ALL_CASES [-y]
```

where:

- *procname* is the procedure name (including sub-procedures).
- *case\_list* is a list of case numbers to be updated, separated by commas and with hyphens denoting ranges of case numbers, for example: 2, 5-10 denotes case 2 and cases 5 to 10 inclusive.

`ALL_CASES` denotes all cases of the procedure.

- `-y` is optional and, if used, suppresses the request for confirmation from the user.

## Chapter 3 Work Items

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil RESEND</code>	Resend outstanding work items.
<code>swutil UNLOCKMAIL</code>	Unlock work items.
<code>swutil CHECKITEM</code>	Check whether the work queue items in the WIS cache have matching records in the <code>outstanding_addr</code> table or not.
<code>swutil CLEANITEM</code>	Remove work queue items from the WIS cache, if the work items do not have matching records in the <code>outstanding_addr</code> table.



To run any of these commands you must be logged in as the *IPEADMIN* user.

## Resending Work Items

---

The following command instructs the iProcess background process to perform an instruction. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

To resend outstanding work items, enter one of the following commands:

```
swutil RESEND procname queueName [casenum |ALL_CASES]
[stepname|ALL_STEPS] [-y] [-P priority]
```

```
swutil RESEND procname ALL_QUEUES [casenum |ALL_CASES]
[stepname|ALL_STEPS] [-y] [-P priority]
```

where:

- *procname* is the procedure name (including sub-procedures).
- *queueName* is the name of the queue (user or group). ALL\_QUEUES means resend all queues.
- *casenum* is the case number you want to resend or use ALL\_CASES to resend all cases of this procedure.
- *stepname* is the step name in the procedure that you want to resend or use ALL\_STEPS to resend all steps in the procedure.
- `-y` is optional and, if used, suppresses the request for confirmation from the user.
- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.



If a procedure is resent that contains sub-procedures, they will not be resent automatically. If the sub-procedures need to be resent, you need to perform an explicit resend on the sub-procedure.

## Clearing Locked Work Queue Items

---

The following command instructs the iProcess background process to perform an instruction. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

You can unlock work queue items that have been marked as locked by a user and not unlocked after use. A work item is locked when it is opened. The iProcess Suite uses two types of lock:

- **Short lock** This is the standard lock normally set by TIBCO iProcess Workspace. A short lock is removed when a user keeps or releases a work item. In the event that short locks are left behind because, for example, the iProcess Workspace crashed, the only way to remove the short lock is to re-start the iProcess Engine or use the following command.
- **Long lock** A lock that is persistent and can only be removed using `sal_llock_frm_init()` by the user holding the lock. Long locks are not normally used by TIBCO iProcess Workspace, but may be set by SAL SDK programs or the TIBCO iProcess Objects Client. Re-starting the iProcess Engine has no effect on long locks, so a work item locked in this way remains inaccessible to every iProcess user until it is manually unlocked.

To unlock a work item, use the following command. You *must* have a *MENUNAME* of *ADMIN* to be able to use this command. See "Managing iProcess Users" in *TIBCO iProcess Workspace (Windows) Manager's Guide* for more information about iProcess attributes.

```
swutil UNLOCKMAIL queuename reqid:hostname [/t]
```

where:

- *queuename* is the work queue that contains the work item you want to unlock. Either specify a queue name or use:
  - `ALL` to specify all work queues.
  - `$UNDELIV` to specify the Undelivered Work Items queue.



As well as being logged in as the *IPEADMIN* user, to use this option you must have access to the specified work queue. (To access a group queue, you must be either a member of that queue or a participant. To access another user's queue, you must be a participant of that queue.)

- *reqid:hostname* is the unique identifier (for example, 10245:node1) of the work item that you want to unlock.



You can find the *reqid* and *hostname* values by using the command `plis -mv queueName`. The value for *reqid* is displayed under the `RQID` heading and the value for *hostname* is displayed under the `Host` heading.

- `/t` is an optional parameter which should be used if *queueName* is a test queue.

The `swutil UNLOCKMAIL` command returns the following error codes:

Error Code	Description
0	No error item unlocked successfully
4	Invalid parameter(s) supplied
6	Internal system error
7	Specified work item was not locked
11	One of the following happens: <ul style="list-style-type: none"> <li>• Specified work item could not be found in <i>queueName</i>.</li> <li>• <i>queueName</i> is invalid.</li> <li>• You do not have access to <i>queueName</i>.</li> </ul>
64	You do not have <i>ADMIN</i> as your <i>MENUNAME</i>

## Checking Work Queue Items

---

Run the following command to check whether the work items in a work queue that has been cached by the WIS process, have matching records in the `outstanding_addr` table or not.



This command is used in very the rare circumstance that there is a discrepancy between the foreground and background records for a work item. The discrepancy is usually caused by some system failure.

```
swutil CHECKITEM queuename workitemtag workitemtag...
```

where:

- *queuename* is the name of the queue that contains the work items you want to check.
- *workitemtag workitemtag...* is the case number and the request ID in the following format:

*case number, request id*

For example: 101,1001.

You can check more than one work item in a queue by using spaces to separate them. For example: 101,1001 102,1002 103,1003.

### Example

#### Checking a Work Item

To check a work item with the case number, 101, and the request ID, 1001, in the `queue1` queue, enter the following command on a command line:

```
swutil CHECKITEM queue1 101,1001
```

#### Checking Work Items in a Queue

To check three work items in the `queue1` queue, enter the following command at a command line:

```
swutil CHECKITEM queue1 101,1001 102,1002 103,1003
```

where each work item with the case number and the request ID is as follows:

- 101,1001
- 102,1002
- 103,1003

## Output of the Command

The output of the command and their descriptions are shown in the following table.

Command Output	Description
Unmatched workitem(s)	Lists the case number and the request ID of the work items that have no matching records in the <code>outstanding_addr</code> table.
Matched workitem(s)	Lists the case number and the request ID of the work items that have matching records in the <code>outstanding_addr</code> table.
Not sure because internal error	Lists the case number and the request ID of work items that are not sure whether have matching records in the <code>outstanding_addr</code> table due to iProcess Engine internal errors.
Unable to find in WIS cache	Lists the case number and the request ID of work items that cannot be found from the WIS cache.

## Removing Work Queue Items

---

Run the following command to remove work items in a work queue that is cached by the WIS process, if the work items do not have matching records in the `outstanding_addr` table.

```
swutil CLEANITEM queuename workitemtag workitemtag...
```

where:

- *queuename* is the name of the queue that contains the work items you want to remove.
- *workitemtag workitemtag...* is the case number and the request ID in the following format:

*case number, request id*

For example: 101,1001.

You can remove more than one work item in a queue by using spaces to separate them. For example: 101,1001 102,1002 103,1003.



This command must be used with the `swutil CHECKITEM` command. You must run the `swutil CHECKITEM` command to confirm that unmatched records exist in the `outstanding_addr` table before running this command to remove the work items.

TIBCO recommends you use this command with guidance from TIBCO Support.

The command checks whether the work items have matching records in the `outstanding_addr` table before removing them from the WIS cache. If the work items have matching records in the `staffo`, `pack_data`, and `pack_memo` tables, these records of work items are also removed from the tables.

### Example

#### Removing a Work Item

To remove a work item with the case number, 101 and the request ID, 1001 in the `queue1` queue, enter the following command at a command line:

```
swutil CLEANITEM queue1 101,1001
```

## Removing Work Items in a Queue

To clean three work items in the `queue1` queue, enter the following command at a command line:

```
swutil CLEANITEM queue1 101,1001 102,1002 103,1003
```

where each work item with the case number and the request ID is as follows:

- 101,1001
- 102,1002
- 103,1003

## Output of the Command

The output of the command and their descriptions are shown in the following table.

Command Output	Description
Cleaned workitem(s)	Lists the case number and the request ID of the removed work items that have no matching records in the <code>outstanding_addr</code> table.
Workitem(s) not cleaned	Lists the case number and the request ID of the work items that are not removed, since they have matching records in the <code>outstanding_addr</code> table.
Not cleaned because internal error	Lists the case number and the request ID of work items that are not removed due to iProcess Engine internal errors.
Unable to find in WIS cache	Lists the case number and the request ID of work items that cannot be found from the WIS cache.

## Chapter 4 Events

This chapter describes the following `swutil` command.

Command	Description
<code>swutil EVENT</code>	Issue an event.



To run this command, you must be logged in as the *IPEADMIN* user.

## Issue an Event

---

Use this command to issue an event from the command line. There is no equivalent menu option. Only users with administration access to a procedure can issue an EVENT on an iProcess case.



If you want to issue a large number of events you should consider using the `swbatch` utility instead. see [Chapter 17, Batch Processing Cases, on page 129](#).

```
swutil EVENT [-hostnode] procname [R]casenum eventname [-P priority]
[datafile [-[r|R][p]]]
```

where:

- *hostnode* is optional and, if included, should be the nodename of the node where the procedure is hosted. If the parameter is omitted the current node is assumed.
- *procname* is the short name of the procedure (including sub-procedures), for example: QUOTA.
- *casenum* is a single case number, for example: 23. You can only issue an event to a single case.

An R preceding the case number can be used to restart a closed case. For example, R23 restarts case number 23.



After a case is closed, all the deadlines of the case are removed. If the case is reopened, you can reset the deadlines by running the `CreateCaseDeadline` function. For more information about the `CreateCaseDeadline` function, see *TIBCO iProcess Expressions and Functions Reference Guide*.

- *eventname* is the short name of the step you are issuing the event for, for example: TREV. (This may be an event step, or an ordinary step.)



If you issue an event for an ordinary step only the actions which follow the step are triggered; the ordinary step itself is not re-run.

- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.
- *datafile* is optional and, if included, should be the full pathname of a file in abox format containing the new field values for the case. See "Using iProcess Abox Files" in *TIBCO iProcess Modeler Integration Techniques* for more

information about abox format.) These values will be used to update the central case data (in the `case_data` database table).

The following options can be used when a *datafile* is specified:

Option	Description
-r	<p>Any deadlines for the case and case steps will be recalculated using the new field values supplied in the <i>datafile</i>. If the case is being restarted (using the R option before the case number) then deadlines for the case and any sub-cases will be recalculated.</p> <p>See "Dynamically Recalculating Deadlines" in <i>TIBCO iProcess Modeler Basic Design</i> for more information about setting deadlines on a step. See "Case Deadline Setting" in <i>TIBCO iProcess Modeler Procedure Management</i> for more information about setting deadlines on a case.</p>
-R	<p>Any deadlines for the case, sub-cases, case steps, and sub-case steps will be recalculated using the new field values supplied in the <i>datafile</i>.</p> <p><b>Note:</b> Only the sub-case data is updated with -p option in the <code>swutil EVENT</code> command, the deadlines of sub-case and sub-case steps are recalculated.</p> <p>See "Dynamically Recalculating Deadlines" in <i>TIBCO iProcess Modeler Basic Design</i> for more information about setting deadlines on a step. See "Case Deadline Setting" in <i>TIBCO iProcess Modeler Procedure Management</i> for more information about setting deadlines on a case.</p>
-p	<p>The new field values supplied in the <i>datafile</i> will also be copied to all outstanding work items (via the <code>pack_data</code> database table).</p> <p>You can propagate these <code>pack_data</code> changes into sub-cases as described in "Propagation of New Field Values" in <i>TIBCO iProcess Modeler Integration Techniques</i>.</p> <p>For more information about updating field values, see "Externally Updating Field Data in a Case" in <i>TIBCO iProcess Modeler Integration Techniques</i>.</p>



## Chapter 5 Activity Monitoring

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil EXPMONITOR</code>	Export activity monitoring configurations
<code>swutil IMPMONITOR</code>	Import activity monitoring configurations



To run any of these commands, you must be logged in as the *IPEADMIN* user.

## Overview

---

TIBCO iProcess Engine can be enabled to publish iProcess Engine activity information to external applications. An activity is any instruction in iProcess Engine that creates an audit trail entry, for example, Case Started or Event Issued. You can configure any combination of step, activity, or both to be monitored. This enables an external application to monitor important business events during the processing of cases.

Activity monitoring import and export configuration information is stored in the iProcess database. You can import and export activity monitoring configuration (in the form of XML) to or from the iProcess database, using the `swutil EXPMONITOR` and `IMPMONITOR` commands.

The format of the XML file must conform to the schemas defined in `SWMonitorList.xsd`. When you install TIBCO iProcess Engine, three schemas are installed which are used to configure the activity monitoring configuration information. The schemas are installed in the `SWDIR\schemas` directory. The schemas are:

- `SWMonitorList.xsd` The format of the Monitor Event Request (MER) messages must conform to this schema. MER messages request the activities to monitor and are sent to the iProcess database to update the activity monitoring configuration information. See "How to Configure Activity Monitoring Information" in *TIBCO iProcess Engine Administrator's Guide* for more information.
- `SWAuditMessage.xsd` The format of the Monitor Event Detail (MED) messages must conform to this schema. MED messages contain the information about the activities being monitored and are sent from the IAP JMS library to the external application. See "Administering Activity Monitoring" in *TIBCO iProcess Engine Administrator's Guide* for more information.
- `SWType.xsd` This defines the iProcess field types that are used in `SWMonitorList.xsd` and `SWAuditMessage.xsd`.

The following sections explain in detail how to run the `swutil EXPMONITOR` and `IMPMONITOR` commands.

## Exporting Activity Monitoring Configuration Information

You can export activity monitoring configuration information from your iProcess Engine to an XML file, which you can subsequently import to the same or another iProcess Engine if required. You can export the configuration information for all procedures or a specific procedure, depending on your requirements.

To export your activity monitoring configuration information, enter the following command:

```
swutil EXPMONITOR [procname]
```

where *procname* is the procedure name. You can specify a main procedure or a sub-procedure.



You must run this command for each procedure or sub-procedure that has activity monitoring configuration information set. For example, if you have a main procedure that calls sub-procedures and you run this command for the main procedure only, the sub-procedures are not included in the export file.

The information is written in XML format. By default, it is written to the console or you can redirect it to a file.



You can locate the file wherever you want. TIBCO iProcess Engine does not require that it is located in a specific place.

The XML format conforms to the `SWMonitorList.xsd` schema (see [Overview on page 28](#) for more information). This enables it to be used in a `swutil IMPMONITOR` command to amend or overwrite activity monitoring configuration information on another iProcess Engine.

If no procedure name is specified in the `swutil EXPMONITOR` command, then all the activity monitoring configuration information is exported. This causes the `FullImport` element to be set to `true` in the XML file. For example,

```
<FullImport>true</FullImport>
```

This means that when you import the activity monitoring configuration information, the import will overwrite the existing activity monitoring configuration on iProcess Engine with the configuration information in the XML file. If you specify a procedure name, the `FullImport` element is set to `false`. This means the imported activity monitoring configuration information is added to the existing activity monitoring configuration information.



If this does not meet your requirements, you can amend the `FullImport` element directly in the XML file.

The server processes the activity monitoring configuration information, and reports and logs any errors it encounters:

- Progress is reported to the console.
- Serious errors are logged either to the `sw_warn.log` file or to the `sw_error.log` file located in the `SWDIR\logs` directory. See "Error Handling" in *TIBCO iProcess Engine Administrator's Guide* for more information.



If you have changed the log files directory in the `staffpms` file, errors will be logged to the `sw_error.log` file and the `sw_warn.log` file located in the directory you have specified.

## Importing Activity Monitoring Configuration Information

---

You can import the activity monitoring configuration information from one iProcess Engine to another. You can import the activity monitoring configuration information for a specific procedure or for all procedures, depending on your requirements. To import your activity monitoring configuration information, enter the following command:

```
swutil IMPMONITOR filename
```

where *filename* is the full path to the XML file that stores the activity monitoring configuration information.

- On Windows, *filename* can be a full pathname or a simple filename in the current directory.
- On UNIX, *filename* can be a full pathname or a simple filename in the `SWDIR\util` directory.

The XML format conforms to the `SWMonitorList.xsd` schema. See [Overview on page 28](#) for more information.

The server processes the activity monitoring configuration information, and reports and logs any errors it encounters:

- Progress is reported to the console.
- Serious errors are logged either to the `sw_warn` file or to the `sw_error` file located in the `SWDIR\logs` directory. See "Error Handling" in *TIBCO iProcess Engine Administrator's Guide* for more information about errors.



If you have changed the log files directory in the `staffpms` file, errors will be logged to the `sw_error` file and the `sw_warn` file located in the directory you have specified.

The new activity monitoring configuration data will be available when the server is next restarted.



## Chapter 6

# Audit Trails

This chapter describes the following `swutil` command:

Command	Description
<code>swutil AUDIT</code>	Creates a user-defined entry in the audit trail



To run this command, you must be logged in as the *IPEADMIN* user.

## Create a User-Defined Audit Trail Entry

---

To create a user-defined entry in the audit trail for a case, enter the following command:

```
swutil AUDIT [-hostname] procname casenum eventnum desc user [-P priority]
```

where:

- *hostname* (optional) is the name of the node where the procedure is hosted. If the parameter is omitted, the current node is assumed.
- *procname* is the procedure name, including sub-procedures.
- *casenum* is the case number you want to create the audit trail entry for.
- *eventnum* is the number of the user-defined audit trail entry, as defined in the `SWDIR\etc\language.lng\auditusr.mes` file. See [Defining Audit Trail Entries on page 34](#).
- *desc* and *user* are text strings which will be displayed in the audit trail entry for the case, in place of the `%desc` and `%user` strings, which are defined in the `event_description` for the entry in the `SWDIR\etc\language.lng\auditusr.mes` file. See [Defining Audit Trail Entries on page 34](#).

The *desc* and *user* strings can be used to supply case-specific information, such as a description of the event, the name of the user affected, and any other desired information. The *desc* string can be a maximum of 24 characters, while *user* a maximum of 255 characters. If either string includes spaces, it should be enclosed in quotation marks.

- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.

### Defining Audit Trail Entries

User-defined audit trail entries must be pre-defined in all of the `SWDIR/etc/language.lng/auditusr.mes` files. You must create these files if you want to define your own audit trail messages.



System audit trail messages are defined in the `SWDIR/etc/language.lng/audit.mes` file. You cannot use system audit trail messages with `swutil`.

User-defined audit trail entries must be defined using the format:

```
eventnum : event_description
```

where:

- *eventnum* is the decimal number which identifies the entry, as referenced by the *eventnum* parameter in the `swutil AUDIT` command. See [Create a User-Defined Audit Trail Entry on page 34](#).

*eventnum* must be 256 or higher. (Numbers 0 to 255 are reserved for use by `iProcess`, and are defined in the `SWDIR/etc/language.lng/audit.mes` file.)

- *event\_description* is a message string which describes the event. It can contain the strings `%user` and `%desc`, which are replaced by the *desc* and *user* strings supplied in the `swutil AUDIT` command. See [Create a User-Defined Audit Trail Entry on page 34](#).



## Chapter 7 **Case Data Queue Parameters**

This chapter describes the `swutil QINFO` command.

### Topics

---

- [Overview of Case Data Queue Parameters \(CDQPs\), page 38](#)
- [Importing CDQP Configuration Data, page 39](#)
- [CDQP Control File, page 41](#)
- [CDQP Command Errors, page 47](#)
- [Exporting CDQP Configuration Data, page 50](#)
- [swutil QINFO Exit Codes and Error Messages, page 54](#)

## Overview of Case Data Queue Parameters (CDQPs)

---

Case Data Queue Parameters allow values from work item case data to be used to sort, display and filter work items lists, and to find specific work items, in Work Queue Manager.

Any field in an iProcess procedure can be used as a CDQP. CDQPs must first be *defined* (globally), and can then be *mapped* to specific queues for use in Work Queue Manager.

Once a CDQP has been mapped to a queue, users with access to that queue can use the CDQP to sort, display and filter the work items list, or to find specific work items. See "Organizing Your Work Items Lists" in *TIBCO iProcess Workspace (Windows) User's Guide* for more information.

The iProcess administrator can:

- import and export CDQP configuration data to or from the server, using the `swutil QINFO` command. The following sections explain in detail how to do this.
- set parameters in the `staffcfg` file to enable or disable the use of CDQPs, and to optimize the effect of their use on system performance. See "Tuning the iProcess Engine Using `staffcfg` Parameters" in *TIBCO iProcess Engine Administrator's Guide* for more information.

## Importing CDQP Configuration Data

---

To import new or updated CDQP configuration data to TIBCO iProcess Engine, use the following command:

```
swutil QINFO -|filename [PARSEONLY|PUBLISH]
```

where:

- - indicates that CDQP commands will be entered from the command line. Commands must be entered using the same syntax and structure as if they were in a control file. Use the appropriate end-of-file key sequence to terminate the command input, for example, CTRL + D on many UNIX systems.
- *filename* is the full path of the CDQP control file, containing CDQP commands. See [CDQP Control File on page 41](#) for more information about the content and format of this file.
- PARSEONLY indicates that the CDQP commands should be validated for correctness, without changing current CDQP configuration data. See [CDQP Command Errors on page 47](#) for more information.
- PUBLISH indicates that affected work queues should be updated immediately with the new data, without having to restart the iProcess Engine. (An event is published to the WIS processes as soon as the import operation has successfully completed, informing them that new CDQP configuration data is available.) See "Configuring CDQP Updates" in *TIBCO iProcess Engine Administrator's Guide* for more information.

If the PUBLISH parameter is omitted, you must restart the iProcess Engine to make the new CDQP configuration data available to affected work queues.

Note that you can use this command only when:

- you are logged in as the *IPEADMIN* user or the *IPEBACKGROUND* user.
- CDQP configuration is enabled in the *SWDIR/etc/staffcfg* file. See "Tuning the iProcess Engine Using *staffcfg* Parameters" in *TIBCO iProcess Engine Administrator's Guide* for more information.

The server processes the CDQP commands, and reports and logs any errors it encounters:

- Progress is reported to the console and logged to the *qinfoimp.log* file located in the *SWDIR\logs* directory.
- Serious errors, that is, those that are fatal either for the entire import process or for a particular queue, are logged either to the *sw\_warn* file or to the *sw\_error* file located in the *SWDIR\logs* directory. See "Error Handling" in

*TIBCO iProcess Engine Administrator's Guide* for more information about errors.



If you have changed the log files directory in the `staffpms` file, errors will be logged to the `qinfoimp.log`, `sw_error`, and `sw_warn` file located in the directory you have specified.

See [swutil QINFO Exit Codes and Error Messages on page 54](#) for a list of exit codes and error messages that the command can return.

**Examples** To enter CDQP commands, use the following command:

---

```
swutil QINFO -
```

---

To validate a set of new or updated CDQP configuration data before importing it, use the following command:

---

```
swutil QINFO C:\TEMP\MYPROC.XFR PARSEONLY
```

---

To import a set of new or updated CDQP configuration data to the server, use the following command:

---

```
swutil QINFO C:\TEMP\MYPROC.XFR
```

---

To import a set of new or updated CDQP configuration data to the server and publish it immediately, use the following command:

---

```
swutil QINFO C:\TEMP\MYPROC.XFR PUBLISH
```

---

## CDQP Control File

---

The CDQP control file is a text file consisting of one or more commands, one command per line:

- A line starting with a semicolon (;) is treated as a comment line and ignored.
- Blank lines are ignored.
- Each command starts with a single character code (for example: F), followed by a modifier (for example: +).
- The code character (plus modifier) can be followed by one or more fields separated by commas.
- (optional) The P command enables prediction on the CDQP. See "Using Case Prediction to Forecast Outstanding Work Items" in *TIBCO iProcess Modeler Advanced Design* for more information about case prediction.
- The maximum line length is 512 characters.

The control file contains the following sections, which must be specified in this order:

- (optional) Reset command - see [Resetting CDQP Configuration Data on page 41](#).
- CDQP definitions - see [Defining CDQPs on page 41](#).
- CDQP queue mappings (optionally containing default user and group queue mappings) - see [Mapping CDQPs to Queues on page 43](#).

### Resetting CDQP Configuration Data

To delete all existing CDQP definitions and queue assignments, including the default user and group queue mappings, use the command:

R



The R command must be the first character of the first non-comment line of the control file, otherwise it is ignored.

### Defining CDQPs

You can define any field in a procedure as a CDQP. Once defined, a CDQP is available system-wide, and can then be mapped to iProcess queues as required.

## About CDQP Definitions

You should note the following when defining CDQPs:

- Fields that are defined as CDQPs are assumed to exist in all iProcess procedures, and to have consistent type and size definitions across all procedures. *No validation is performed to check this. (Any fields that do not exist will be treated as null or blank fields in Work Queue Manager.)*
- CDQPs are stored as text fields. Non-text fields should therefore be sized as if they were text fields, for example, 10 characters for a Date field (*dd/mm/yyyy*).
- Memo fields should not be defined as CDQPs.
- If a CDQP is used in a queue, for example, as a display column, but the corresponding field does not exist in a given procedure, or is a memo field, the value returned will correspond to a zero-length text field.

## Add or Change a CDQP

To add a field as a CDQP, or to change the definition of an existing CDQP, use the command:

*F+FieldName, FieldSize, WQMDescription [ , P]*

where:

- *FieldName* is the name of the iProcess field to be defined as a CDQP. *FieldName* must be less than or equal to 15 characters.

If the field is a composite field of the form *TagName:FieldName*, *TagName* must be less than 15 characters and *FieldName* must be less than 15 characters.

- *FieldSize* is the size, in characters, of the CDQP. *FieldSize* must be less than or equal to 255.
- *WQMDescription* is the name that will be used to represent this field in Work Queue Manager dialogs. *WQMDescription* must be unique, and must be less than or equal to 39 characters.
- *P* is the command used to enable prediction on the CDQP.

## Delete a CDQP

To delete an existing CDQP, use the command:

*F-FieldName*

where *FieldName* is the name of the iProcess field to be deleted as a CDQP.

## Example CDQP Definitions

The following control file fragment illustrates how to define CDQPs.

---

```

; Add the fields TOWN and AGE as CDQPs.
;
F+TOWN,20,Town/City
F+AGE,3,Age

; Change the definition of an existing CDQP,
; DRIVER_NAME.
;
F+DRIVER_NAME,30, Driver Name

; Delete an existing CDQP, DISTRICT, that is
; no longer required.
;
F-DISTRICT

```

---

## Mapping CDQPs to Queues

When you have defined fields as CDQPs, you can map them to particular iProcess queues for use in Work Queue Manager.

### Add or Change a Queue Mapping

To add or change a CDQP mapping for a queue, use the following command:

```
Q+QueueName[ , QueueName , . . . ]
```

where *QueueName* is the name of an iProcess queue, which must be already defined in iProcess. Multiple *QueueNames* can be specified, separated by commas.



Q commands must appear after the CDQP definition section (F commands) in the control file. Any F commands that follow a Q command are ignored.

### Add or Delete Field Mappings

After the Q command, you can add or delete field mappings for this queue, as follows:

- To add any *FieldName* already defined as a CDQP (see [Add or Change a CDQP on page 42](#)) to the specified queue's mapping, use the command:

```
C+FieldName
```

The maximum number of CDQPs that can be mapped to a queue depends upon the *FieldSize* of each CDQP, and upon the number and size of any other

standard selectable fields that are mapped to the queue. The larger the *FieldSize*, the fewer CDQPs you can map to the queue. For example, if each CDQP has a *FieldSize* of 255, and the default selectable fields are used, *approximately* 14 CDQPs can be mapped to a single queue, although this is not an exact calculation.

If you attempt to map too many fields to a queue a validation error will occur See [C+ on page 49](#).



Only the first 50 CDQPs that are mapped to a queue can be selected for display in the Display Criteria dialog in Work Queue Manager.

- To delete any *FieldName* from the queue's existing mapping, use the command:  
*C-FieldName*

## Delete a Queue Mapping

To delete a queue's existing CDQP mapping, use the command:

```
Q-QueueName[ , QueueName , . . . ]
```

where *QueueName* is the name of an iProcess queue, which must be already defined in iProcess. Multiple *QueueNames* can be specified, separated by commas.

All fields currently mapped to the specified queue are removed. The queue reverts to the default user or group queue mapping, if one has been configured.

## Default User and Group Queue Mappings

You can use the Q and C commands described above to define default user and group queue mappings. The default mappings will be used for all queues on the system that do not have their own mappings defined.

To define default mappings:

1. Use the following identifiers as the *QueueName* in a Q command:
  - SW\_DEFAULT\_CASEDATAQPARAM\_USER defines the default user queue mapping.
  - SW\_DEFAULT\_CASEDATAQPARAM\_GROUP defines the default group queue mapping.
2. Add or delete field mappings as required using C commands. See [Add or Delete Field Mappings on page 43](#).

## Exclude a Queue from Using Default Mappings

You can exclude a particular queue from using default mappings if required. To do this:

- If the queue has no mappings already, use the Q+ command to set up a queue mapping, but do not use any C commands to map fields.

```
Q+QueueName
```

- If the queue already has field mappings, delete them using C- commands. For example:

---

```
Q+QueueName
C-FieldName
...
```

---

## Example Queue Mappings

The following control file fragment continues the example shown in [Example CDQP Definitions on page 43](#), and illustrates the use of the different queue mapping commands:

---

```
; Define DRIVER_NAME as the default mapping for user and group queues.
;
Q+SW_DEFAULT_CASEDATAQPARAM_USER
C+DRIVER_NAME

Q+SW_DEFAULT_CASEDATAQPARAM_GROUP
C+DRIVER_NAME

; Add AGE to user queues usr30 and usr31.
;
Q+usr30,usr31
C+AGE

; The following command is intended to add VEHICLE as a CDQP, but will be ignored
; because it is specified AFTER a Q command.
;
F+VEHICLE,20,Vehicle Type

; Remove TOWN from user queue usr30.
;
Q+usr30
C-TOWN

; The following command is intended to add VEHICLE to user queue usr31, but will be
; ignored because VEHICLE is not a CDQP.
;
```

```
Q+usr31  
C+VEHICLE
```

```
; Delete existing mappings for user queue usr24, which will revert to the default  
; user queue  
; mapping.  
;  
Q-usr24
```

```
; Remove current mapping (TOWN) from group queue group10, and exclude it from using  
; the default group queue mapping.  
;  
Q+group10  
C-TOWN
```

---

## CDQP Command Errors

CDQP command errors are reported using messages in the following format:

Line *LineNum*: *Error*

where:

- *LineNum* is the line number of the command that failed validation, including blank lines.
- *Error* is one of the error messages shown in the following table. The command that each error message relates to is also shown in the table.

Table 4 CDQP Command Errors (Sheet 1 of 3)

Command	Error Message	Description
All	Invalid line length; line ignored	Maximum line length is 512 characters.
	Invalid control file command; line ignored	Command character must be R, F, Q or C.
	Invalid control file command context; line ignored	Command character is out of context. For example, a C command must follow a Q command.
	Invalid control file modifier; line ignored	The modifier character following a Q or C command must be either a plus (+) or minus (-) sign.
R	Invalid reset command; line ignored	R must be the only non-whitespace character on the line See <a href="#">Resetting CDQP Configuration Data on page 41</a> .
F+	Invalid number of parameters; line ignored	The F+ command must be followed by three parameters. See <a href="#">Add or Change a CDQP on page 42</a> .
	Invalid iProcess field name; line ignored	The specified <i>FieldName</i> (or each component in a composite <i>FieldName</i> ) must be less than or equal to 15 characters. See <a href="#">Add or Change a CDQP on page 42</a> .
	Invalid size parameter; line ignored	The specified <i>FieldSize</i> must be less than or equal to 255 characters or bytes. See <a href="#">Add or Change a CDQP on page 42</a> .
	Invalid description parameter; line ignored	The specified <i>WQMDescription</i> must be less than or equal to 39 characters. See <a href="#">Add or Change a CDQP on page 42</a> .
	Description already in use; line ignored	The specified <i>WQMDescription</i> must be unique. See <a href="#">Add or Change a CDQP on page 42</a> .

Table 4 CDQP Command Errors (Sheet 2 of 3)

Command	Error Message	Description
F-	Invalid number of parameters; line ignored	The F- command must be followed by a single <i>FieldName</i> parameter. See <a href="#">Delete a CDQP on page 42</a> .
	iProcess field name does not exist; line ignored	The specified <i>FieldName</i> is not defined as a CDQP, so it cannot be deleted. See <a href="#">Delete a CDQP on page 42</a> .
Q+	Invalid number of parameters; line ignored	The Q+ command must be followed by one or more <i>QueueName</i> parameters. See <a href="#">Add or Change a Queue Mapping on page 43</a> .
	Invalid iProcess Queue Name; queue <i>queuename</i> ignored	The specified <i>queuename</i> is not one of the following: <ul style="list-style-type: none"> <li>• an iProcess work queue</li> <li>• SW_DEFAULT_CASEDATAQPARAM_USER</li> <li>• SW_DEFAULT_CASEDATAQPARAM_GROUP</li> </ul> See <a href="#">Mapping CDQPs to Queues on page 43</a> .
Q-	Invalid number of parameters; line ignored	The Q- command must be followed by one or more <i>QueueName</i> parameters. See <a href="#">Add or Change a Queue Mapping on page 43</a> .
	CDQP data does not exist for queue <i>queuename</i> ; queue ignored	The specified <i>queuename</i> has no CDQP mappings, so they cannot be deleted. See <a href="#">Add or Change a Queue Mapping on page 43</a> .
	Invalid iProcess Queue Name; queue <i>queuename</i> ignored	The specified <i>queuename</i> is not one of the following: <ul style="list-style-type: none"> <li>• an iProcess work queue</li> <li>• SW_DEFAULT_CASEDATAQPARAM_USER</li> <li>• SW_DEFAULT_CASEDATAQPARAM_GROUP</li> </ul> See <a href="#">Mapping CDQPs to Queues on page 43</a> .

Table 4 CDQP Command Errors (Sheet 3 of 3)

Command	Error Message	Description
C+	Invalid number of parameters; line ignored	The C+ command must be followed by a single <i>FieldName</i> parameter. See <a href="#">Add or Delete Field Mappings on page 43</a> .
	Global CDQP data does not exist for specified iProcess field name; line ignored	The specified <i>FieldName</i> has not been previously defined as a CDQP. See <a href="#">Add or Delete Field Mappings on page 43</a> .
	iProcess field name already exists; line ignored	The specified <i>FieldName</i> is already mapped to this queue. See <a href="#">Add or Delete Field Mappings on page 43</a> .
	Size of iProcess field causes RPC packet limit to be exceeded; line ignored	You cannot map the specified <i>FieldName</i> to this queue (see <a href="#">Add or Delete Field Mappings on page 43</a> ), because each work item's data record size will then exceed 4k bytes, which is the maximum size of the RPC data packets used to retrieve work item data by Work Queue Manager. You will need to reduce the <i>FieldSize</i> of one or more of the CDQPs to be mapped to this queue.
C-	Invalid number of parameters; line ignored	The C- command must be followed by a single <i>FieldName</i> parameter. See <a href="#">Add or Delete Field Mappings on page 43</a> .
	iProcess field name does not exist for this queue; line ignored	The specified <i>FieldName</i> is not mapped to this queue, so it cannot be deleted. See <a href="#">Add or Delete Field Mappings on page 43</a> .

## Exporting CDQP Configuration Data

---

You can export CDQP definitions and mappings - either for the entire system, or for specific queues - using the following command:

```
swutil QINFO EXPORT [queuename [GLOBAL]]
```

where:

- *queue*name is the name of the iProcess queue for which CDQP mappings are to be exported. If *queue*name is omitted, all CDQP data is exported - CDQP definitions, default user/group mappings and all defined user/group queue mappings.

You can also use any of the following keywords in place of a *queue*name:

- SW\_DEFAULT\_CASEDATAQPARAM\_USER indicates that the default user queue mappings should be exported.
- SW\_DEFAULT\_CASEDATAQPARAM\_GROUP indicates that the default group queue mappings should be exported.
- SW\_DEFAULT\_CASEDATAQPARAM\_GLOBAL indicates that only CDQP definitions should be exported (without any queue mappings).
- GLOBAL indicates that CDQP definitions are to be exported, along with whatever queue mappings are defined by the *queue*name parameter.



Using the command `swutil QINFO EXPORT SW_DEFAULT_CASEDATAQPARAM_GLOBAL GLOBAL` will generate only one copy of CDQP definitions.

You must be logged in as the *IPEADMIN* user to use this command. You can use the command whether or not the iProcess Engine is running.

See [swutil QINFO Exit Codes and Error Messages on page 54](#) for a list of exit codes and error messages that the command can return.

Configuration data is written to the standard output device, but can be redirected to a file - either for reference or for editing and subsequent re-importing. See [CDQP Control File on page 41](#).

### Export File Format

The exported file will contain some or all of the following sections, in this order:

1. A Reset command (if a specific queue is not being exported).
2. CDQP definitions.

3. Default group queue mappings (if they exist).
4. Default user queue mappings (if they exist).
5. User and group queue mappings (if they exist).

## CDQP Filter Identifiers

The queue mappings in the export file contain (as comments) unique *filter identifiers* for each mapped field in a queue, in the format:

`dn`

where *n* is a unique integer indicating the order that the field was mapped to the queue. For example, `d3` is the 3rd field mapped to this queue.



The numbers used are unique to each queue.

You can use these filter identifiers to create additional filter parameters when starting TIBCO iProcess Workspace (Windows) from command line, to:

- filter the work queue (`staffw OPENWQS FILTER`)
- find and open a work item in the work queue (`staffw OPENWQS OPENITEM`)

See *TIBCO iProcess Workspace (Windows) User's Guide* for more information about how to use these filter identifiers.

## Export File Examples

### Example 1

This example shows output from the command:

```
swutil QINFO EXPORT
```

which exports all CDQP configuration data on the system.

---

```
R
; CDQP definitions
F+DRIVER_NAME,30,Driver Name
F+TOWN,20,Town/City
F+AGE,3,Age

; Default Group CDQP mappings
Q+SW_DEFAULT_CASEDATAQPARAM_GROUP
; d1
C+DRIVER_NAME
```

```

; Default User CDQP mappings
Q+SW_DEFAULT_CASEDATAQPARAM_USER
; d1
C+DRIVER_NAME

; Queue CDQP mappings
Q+usr30
; d1
C+AGE
; d2
C+DRIVER_NAME

Q+usr31
; d1
C+AGE
; d2
C+DRIVER_NAME

Q+grp40
; d1
C+DRIVER_NAME
; d2
C+TOWN
; d3
C+AGE

```

---

**Example 2**

This example shows output from the command:

```
swutil QINFO EXPORT grp40
```

which exports just the queue mappings for the group queue grp40.

---

```

; Queue CDQP mappings
Q+grp40
; d1
C+DRIVER_NAME
; d2
C+TOWN
; d3
C+AGE

```

---

**Example 3**

This example shows output from the command:

```
swutil QINFO EXPORT usr30 GLOBAL
```

which exports both CDQP definitions, and the queue mappings for the user queue `usr30`.

---

```

; CDQP definitions
F+DRIVER_NAME,30,Driver Name
F+TOWN,20,Town/City
F+AGE,3,Age

; Queue CDQP mappings
Q+usr30
; d1
C+AGE
; d2
C+DRIVER_NAME

```

---

#### Example 4

This example shows output from the command:

```
swutil QINFO EXPORT SW_DEFAULT_CASEDATAQPARAM_USER GLOBAL
```

which exports the CDQP definitions and the default user queue mappings.

---

```

; CDQP definitions
F+DRIVER_NAME,30,Driver Name
F+TOWN,20,Town/City
F+AGE,3,Age

; Queue CDQP mappings
Q+SW_DEFAULT_CASEDATAQPARAM_USER
; d1
C+DRIVER_NAME

```

---

## swutil QINFO Exit Codes and Error Messages

The following table lists the error codes and messages that may be returned by `swutil QINFO`.

Table 5 *swutil QINFO Exit Codes and Error Messages (Sheet 1 of 3)*

Exit Code	Error Message	Description
0	n/a	Command succeeded.
1	Invalid user	You must be logged in as: <ul style="list-style-type: none"> <li>the <code>IPEADMIN</code> user or <code>IPEBACKGROUND</code> user, if you want to import CDQP configuration data (see <a href="#">Importing CDQP Configuration Data on page 39</a>.)</li> <li>an iProcess administrator, that is, with a <code>MENUNAME</code> of <code>ADMIN</code>, if you want to export CDQP configuration data (see <a href="#">Exporting CDQP Configuration Data on page 50</a>.)</li> </ul>
2	n/a	This code is currently not used.
3	Filename not found	The specified CDQP control file <i>filename</i> does not exist . See <a href="#">Importing CDQP Configuration Data on page 39</a> .
4	Invalid control file	The specified CDQP control file is invalid. See <a href="#">CDQP Control File on page 41</a> .
5	No CDQP data defined	You have tried to export CDQP configuration data, but no CDQPs are defined on the server.
6	No queue CDQPs defined	You have tried to export CDQP configuration data for a queue that has no queue mappings defined.
7	swutil usage message	You have tried to use a <code>swutil QINFO</code> command with either: <ul style="list-style-type: none"> <li>invalid keywords, or</li> <li>an incorrect number of parameters.</li> </ul>
8	Invalid queue name	You have tried to export CDQP configuration data for a queue, but the specified <i>queuename</i> either does not exist or is not a valid keyword. See <a href="#">Exporting CDQP Configuration Data on page 50</a> .

Table 5 swutil QINFO Exit Codes and Error Messages (Sheet 2 of 3)

Exit Code	Error Message	Description
9	System not configured for CDQP configuration	<p>You cannot import CDQP configuration data because CDQP configuration is disabled on this server (by CDQP parameters in the <code>SWDIR\etc\staffcfg</code> file).</p> <p>See "Tuning the iProcess Engine Using <code>staffcfg</code> Parameters" in <i>TIBCO iProcess Engine Administrator's Guide</i> for more information.</p>
10	CDQP Configuration exceeds system limit <i>global_message</i> <i>queue_message</i>	<p>You have attempted to import CDQP configuration data, but have exceeded current CDQP configuration limits for this server. <i>global_message</i> appears only if the global CDQP limit is exceeded. The message displayed is:</p> <p>Max Global CDQP limit is <i>maxglobal</i>, <i>numCDQPs</i> are configured</p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>maxglobal</i> is the maximum number of CDQPs that can be defined on the server, as defined by the <code>CDQPMAXGLOBAL</code> parameter in the <code>SWDIR\etc\staffcfg</code> file. See "Tuning the iProcess Engine Using <code>staffcfg</code> Parameters" in <i>TIBCO iProcess Engine Administrator's Guide</i> for more information.</li> <li>• <i>numCDQPs</i> is the number of CDQPs defined on the server after processing the CDQP control file.</li> </ul> <p>A <i>queue_message</i> appears for each queue for which the CDQP limit is exceeded. The message displayed is:</p> <p>Max Queue CDQP limit is <i>maxqueue</i>, <i>mappings</i> are configured for queue <i>queueName</i></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>maxqueue</i> is the maximum number of CDQPs that can be mapped to a specific queue, as defined by the <code>CDQPMAXQUEUE</code> parameter in the <code>SWDIR\etc\staffcfg</code> file. See "Tuning the iProcess Engine Using <code>staffcfg</code> Parameters" in <i>TIBCO iProcess Engine Administrator's Guide</i> for more information.</li> <li>• <i>mappings</i> is the number of CDQPs mapped to the queue after processing the CDQP control file.</li> <li>• <i>queueName</i> is the name of the affected iProcess queue.</li> </ul>
11-14	n/a	These codes are currently not used.

Table 5 *swutil QINFO Exit Codes and Error Messages (Sheet 3 of 3)*

Exit Code	Error Message	Description
15	Non-Fatal error(s) have occurred while processing Queue CDQP configuration. See the <code>sw_warn</code> file.	Errors affecting one or more queues have occurred during the import process. See <i>TIBCO iProcess Engine System Messages Guide</i> for more information.
16	A Fatal error has occurred. See the <code>sw_warn</code> or <code>sw_error</code> file.	The import process has failed with a fatal error. The status of the server's CDQP configuration data depends on the phase at which the error occurred. See <i>TIBCO iProcess Engine System Messages Guide</i> for more information.

## Procedures and Procedure Libraries

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil IMPORT</code>	Import a procedure definition (XFR) file.
<code>swutil IMPORTLIB</code>	Import a procedure library definition (XFR or XPDL) file.
<code>swutil EXPORT</code>	Export a procedure to a procedure definition (XFR) file.
<code>swutil EXPORTLIB</code>	Export a procedure library to a procedure definition (XFR) file.



Many of the prompts used with these commands involve version control, for example, the version number and release status to be used. See "Using Version Control" in *TIBCO iProcess Modeler Procedure Management* for more information.

TIBCO does not recommend that you import procedures, which are exported from a later version of TIBCO iProcess Engine, to an earlier version of TIBCO iProcess Engine.

To run any of these commands you must be logged in either as the `IPEADMIN` user or, on a UNIX system, as the `IPEBACKGROUND` user or `root` user.

## Overview of Importing/Exporting Procedures and Libraries

---

Procedures and/or procedure libraries can easily be moved around between iProcess engines so that a procedure developed in one office can be exported, sent to another office and imported on their system to use.

Each command can be used:

- interactively, following the prompts and questions, or
- manually using command line options to automatically answer questions and/or suppress questions.

When you use the `swutil` utility to export or import procedures and/or libraries, you are prompted to answer a number of questions. If you are exporting or importing:

- a library that contains multiple procedures/libraries, or
- a procedure that calls sub-procedures,

you have to answer the same questions multiple times, although your answer may be the same for all the procedures/libraries you are processing.

To make the process faster and reduce the level of interaction required, there are:

- prompts that allow you to specify that the answer applies to all procedures (or sub-procedures), so that the prompt is not displayed again.
- command line options that allow you to suppress specific prompts and change specific values for all the procedures to be processed by the command.

## Import a Procedure

---

This section describes how to import a procedure into Procedure Manager. For example, you might want to do this if you have received a procedure definition file (.xfr) from another office that you need to use.

Note that:

- Single procedures can only be imported to the root Procedure Management library. You can move them to your target libraries after the import using Procedure Manager.
- If you import a .xfr file that was exported from iProcess Version 8 or Version 9, any Autosteps or Open Client steps will be ignored. After you have imported the procedure, you can replace or remove these step types in TIBCO iProcess Modeler.
- By default, the procedure keeps the version number contained in the .xfr file. A pre-iProcess Version 10.0 procedure uses the default version number (because it has no version numbering information in the .xfr file).

### Import a Procedure Interactively

The interactive method uses the `swutil` menu and displays prompts for each stage of the import. If you want to import a procedure without displaying all of the prompts, use the command line method. See [Import Procedures Using Command Line Options on page 63](#).

To import a procedure (the following example uses a procedure called `APPLY1`) using the `swutil` menu:

1. Select the following option from the `swutil` menu (see [Using swutil on page 2](#)):

7 - Import an 'xfr' file

The following prompt is displayed:

---

```
Working, Please Wait .....
Importing a Procedure
Name of xfr file (<CR> to quit )?
```

---

2. Enter the name of the .xfr file that contains the procedure you want to import.

For example, if the procedure definition file is called `apply1.xfr`, enter `apply1`.

- If the procedure already exists on the target system, you cannot overwrite it. The following prompt is displayed:

---

```
Checking Procedure 'APPLY1':
Procedure name APPLY1 already exists
'R'ename, 'N'ew version or 'A'lways new version?
```

---

Instead, you must choose one of the following options:

- **R** Rename the procedure.  
(You cannot rename an OEM locked procedure. If you do, the import attempt will fail.)  
Name of procedure (<CR> to quit).
  - **N** Create a new version of the procedure, in which case the minor version number is automatically incremented.
  - **A** Always create a new version. Select this option, if you are importing more than one procedure and want to create a new version for each one. This prompt is then suppressed for any remaining procedures during this command session.
- If you chose to rename the procedure, you are prompted to enter a new username for the owner of the procedure. If you chose to create a new version, go to step 5.

---

```
Do you want to change the default owner - owner?
Yes, No, A yes and apply to all, Skip all
```

---

You can choose one of the following:

- **Yes** - and then enter a new username.  
iProcess checks that this username is a valid iProcess user, but does not verify that the user has the appropriate permissions on the procedure. You must ensure that the specified username has appropriate access to the procedure.
  - **No** - and use the existing username.
  - **A** - and then enter a new username and this will apply to all remaining procedures being imported.
  - **S** - responds No to all and sets the owner to the user performing the import.
- If you are importing a procedure to an iProcess Engine that has case data normalization enabled and the procedure you are importing does not (for example, it may have come from an earlier iProcess Engine version that does

not support case data normalization), then you are prompted to import the procedure with case data normalisation enabled.

---

Enable Case Data Normalization for the current procedure?  
'Y'es, 'N'o, 'A'(Yes and apply to all), 'S'kip (No to all ):

---

You can choose one of the following:

- **Y**es - to enable case data normalization.
- **N**o - to import the procedure without case data normalization enabled.
- **A** - yes to enable case data normalization and apply this to all remaining procedures being imported.
- **S** - no to enabling case data normalization and apply this to all remaining procedures being imported.

6. You are prompted to overwrite the procedure description. Select **Y** to overwrite the procedure description, or **N** to use the existing procedure description.

7. Enter the new release status for the procedure. The available options are determined by the following factors:
  - a. The release status of the procedure in the `.xfr` file, as shown in the following table:

Table 6 Procedure Release Status in `.xfr` and Available Import Options

Status in <code>.xfr</code> file	Available import options
Incomplete	Incomplete, Withdrawn
Unreleased	Unreleased, Withdrawn
Released	Model, Released, Withdrawn
Model	Model, Released, Withdrawn
Withdrawn	Withdrawn

- b. The current release status of the procedure versions on the target system (if you have chosen to create a new version of a procedure that already exists).

The status that a procedure can be imported as depends upon the status of any existing versions of the procedure on the system, as shown in the following table.

Table 7 Procedure Release Status on Target System and Required Conditions for Import

Status to be imported	Required conditions for import
<i>Unreleased or Incomplete</i>	If there is not already an Unreleased or Incomplete version of the procedure on the system.
Model	If there is not already a Model version of the procedure on the system.
Released	If there is not already a Released version of the procedure on the system. If there is, then you will only be able to import the procedure as Withdrawn.

8. (Optional) Specify whether or not you want to import the table data for any tables used by the procedure. This only appears if the procedure contains any tables.



TIBCO recommend that you import table data, otherwise you may experience problems when you edit the procedure definition or try to start cases of the procedure.

9. (Optional) If the procedure uses sub-procedures, steps 3 to 5 are repeated for each sub-procedure.
10. You are prompted if you want to import another procedure. Enter **Y** or **N**. If you enter **Y**, continue from step 1.

The procedure is imported to the root Procedure Management library. If you want to move the procedure to another library, use Procedure Manager to do so. See "Moving and Copying Procedures, Libraries and Shortcuts" in *TIBCO iProcess Modeler Procedure Management*.

## Import Procedures Using Command Line Options

Use the `swutil IMPORT` command line options to quickly import one or more procedures. By setting the required command line options, you can define the required import options and suppress the prompts/questions.

To import a procedure, use the following command:

```
swutil IMPORT [procedure] [options]
```

where:

- *procedure* is the name of the `.xfr` file to import.  
To import a procedure (for example, `carpool.xfr`) from a directory outside of `SWDIR/util` (for example, `/Dept/Purchasing`), run the following command:  

```
swutil IMPORT /Dept/Purchasing
```
- *options* can be one or more of the following to suppress or change a specific option:

Table 8 *swutil IMPORT Options (Sheet 1 of 3)*

Option	Description
+c	Overwrite the existing procedure description when you import a procedure as a new version.
-c	Do not overwrite the existing procedure description when you import a procedure as a new version.
-d	Do not prompt for the procedure's <i>Owner</i> (the current user is used).
+d <i>owner</i>	Do not prompt for the <i>Owner</i> but set the value of <i>Owner</i> to the value supplied in <i>owner</i> .
-e	Use existing use files if there is a clash of names with existing files.
-g	Suppress confirmation when generating a new GUID for a renamed procedure.
-h	Display the command help (showing these options.)

Table 8 *swutil IMPORT Options (Sheet 2 of 3)*

Option	Description
-ix	<p>Import an Incomplete procedure with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>i</i> for Incomplete</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-ii</code> imports an Incomplete procedure as Incomplete.</p> <p><b>Note:</b> If you choose <code>-ii</code> but an Incomplete or Unreleased version of the procedure already exists, the option is ignored and the procedure is imported as Withdrawn instead.</p>
-l	Overwrite lists that already exist.
-mx	<p>Import a Model procedure with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>m</i> for Model</li> <li>• <i>r</i> for Released</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-mr</code> imports a Model procedure as Released.</p> <p><b>Note:</b> If you choose <code>-mm</code> but a Model version of the procedure already exists, or <code>-mr</code> and a Released version already exists, the option is ignored and the procedure is imported as Withdrawn instead.</p>
-n	Choose to create a new version of the procedure for the prompt “New version or New name”.
+n <i>name</i>	Import the procedure as a new procedure with the given name instead of prompting for “New version or New name.”
-oa	Do not overwrite access control properties
+oa	Overwrite access control properties
-q	<p>Import a procedure using the same procedure name (as a new version), owner and release status (except for an Incomplete procedure) with which it was exported, without having to complete the associated prompts.</p> <p><b>Note:</b> This option is equivalent to using the <code>-d</code>, <code>-n</code>, <code>-mm</code>, <code>-rr</code> and <code>-uu</code> options.</p>
-rx	<p>Import a Released procedure with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>m</i> for Model</li> <li>• <i>r</i> for Released</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-rr</code> imports a Released procedure as Released.</p> <p><b>Note:</b> If you choose <code>-rm</code> but a Model version of the procedure already exists, or <code>-rr</code> and a Released version already exists, the option is ignored and the procedure is imported as Withdrawn instead.</p>

Table 8 *swutil IMPORT Options (Sheet 3 of 3)*

Option	Description
-t	Respond No to Import/Overwrite Table Data? prompt. Do not Import/Overwrite iProcess Table Data.
+t	Respond Yes to Import/Overwrite Table Data? prompt. Import/Overwrite iProcess Table Data.
-ux	<p>Import an Unreleased procedure with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• u for Unreleased</li> <li>• w for Withdrawn</li> </ul> <p>For example, -uu imports an Unreleased procedure as Unreleased.</p> <p><b>Note:</b> If you choose -uu but an Incomplete or Unreleased version of the procedure already exists, the option is ignored and the procedure is imported as Withdrawn instead.</p>
-v	Display messages showing if a procedure has been committed to the database or rolled back. See <a href="#">Display Commit/Rollback Messages When Importing Procedures or Procedure Libraries on page 72</a> .
+z	Import the procedure with case data normalization enabled. See <i>TIBCO iProcess Engine Administrator's Guide</i> for more information about case data normalization.
-z	Import the procedure with case data normalization disabled. See <i>TIBCO iProcess Engine Administrator's Guide</i> for more information about case data normalization.



- For any options that you do not set or suppress, you will be prompted in the same way as using the interactive method of importing a procedure.
- If you do *not* specify a procedure name or any options with the command, a "Do you want to import another xfr file?" prompt appears at the end of the current import process.  
  
If you specify a procedure name or any options, the prompt is suppressed and the command terminates at the end of the current import process.

The procedure is imported to the root Procedure Management library. If you want to move the procedure to another library, use Procedure Manager to do so. See "Moving and Copying Procedures, Libraries and Shortcuts" in *TIBCO iProcess Modeler Procedure Management*.

## Example

To import a released procedure called `quota` as a Model procedure suppressing all the other questions/prompts, use the following command:

---

```
swutil IMPORT quota -d -e -rm +t
```

---



You can abbreviate the options as shown in the above example. Therefore, instead of entering `-d -e- -rm +t`, you just need to enter `-derm +t`. You need to leave a space between a `-` option and a `+` option.

## Import a Procedure Library

---

This section describes how to import a procedure library into Procedure Manager. Each procedure and sub-procedure contained in the library is also imported.

### Note:

- This option is not available from the `swutil` menu. You can only import procedure libraries using the command line option either interactively or by using the options to suppress the prompts.
- If your library pathnames use spaces you need to quote the path, for example:

---

```
swutil IMPORTLIB newlib '/dev/dept 1'
```

---

- The hierarchical structure of the library is maintained when you import it.
- For each procedure and sub-procedure in the library, the command displays the same prompts that you would see if you were importing the procedure individually. See [Import a Procedure on page 59](#) for more information.
- Libraries and sub-libraries do not have a release status or version number. This means for each library and/or sub-library you import, you are not required to set a status or version number.
- An exported root library can only be imported as the root library on the target system.

## Import a Procedure Library Interactively

The interactive method displays prompts for each stage of the import. If you want to import a procedure library without displaying all of the prompts, use the command line method. See [Import a Procedure Library Using Command Line Options on page 69](#).



The `IMPORTLIB` option is not available from the `swutil` menu.

1. To import a procedure library, use the following command:

```
swutil IMPORTLIB library target
```

where:

- *library* is the name of the library you want to import.

To import a procedure library (for example, `carpool.xpd1`) from a directory outside of `SWDIR/util` (for example, `/Dept/Purchasing`), run the following command:

```
swutil IMPORTLIB /Dept/Purchasing/carpool.xpd1 target
```

- *target* is the full path to the library (as shown in Procedure Manager).



If your library names use spaces, you must quote them in the command line. See [Import a Procedure Library on page 67](#).

The library can be either an `.xfr` or `.xpd1` library. If there is only one type in the folder, then iProcess imports that library automatically. If however there are both types such as `carpool.xfr` and `carpool.xpd1`, you will be prompted to choose which you want to import, as follows:

```
'F' carpool.xfr, 'P' carpool.xpd1, 'Q'uit?
```

Enter **F** to import the `.xfr` or enter **P** to import the `.xpd1`.

- For each procedure and sub-procedure in the library, the command displays the same prompts that you would see if you were importing the procedure individually. See [Import a Procedure on page 59](#) for more information.

The library and all of its procedures are imported to Procedure Manager in the target you specified. If you want to move the library or procedures, use Procedure Manager to do so. See "Moving and Copying Procedures, Libraries and Shortcuts" in *TIBCO iProcess Modeler Procedure Management*.



When you import an `.xpd1` library, it already has a default destination library, which is the user input package name used when you exported the procedures as `.xpd1` from TIBCO Business Studio. However, you still can define the destination library, which will contain the default destination library.

## Example

To import a procedure library from the `admin.xfr` file into the `/Dept/Purchasing` library following the interactive prompts, you would use the following command:

---

```
swutil IMPORTLIB admin /dept/purchasing
```

---

To import a procedure library from the `admin.xfr` file into the `/Dept/Purchasing 2` library following the interactive prompts, you would use the following command (note the quoted pathname):

```
swutil IMPORTLIB admin '/dept/purchasing 2'
```

## Import a Procedure Library Using Command Line Options

To import a procedure library using the command line options, use the following command:

```
swutil IMPORTLIB library target [options]
```

where:

- *library* is the name of the library you want to import.
- *target* is the full path to the library (as shown in Procedure Manager).



If your library names use spaces, you must quote them in the command line. See [Import a Procedure Library on page 67](#).

- *options* can be one or more of the following to suppress or change a specific option:

Table 9 *swutil IMPORTLIB options (Sheet 1 of 3)*

Option	Description
+c	Overwrite the existing procedure description when you import a procedure as a new version.
-c	Do not overwrite the existing procedure description when you import a procedure as a new version.
-d	Do not prompt for the procedure's <i>Owner</i> (the current user is used).
+d <i>owner</i>	Do not prompt for the <i>Owner</i> but set the value of <i>Owner</i> to the value supplied in <i>owner</i> .
-e	Use existing use files if filenames are the same.
-g	Suppress confirmation when generating a new GUID for a renamed procedure.
-h	Displays the help (shows these options).

Table 9 *swutil IMPORTLIB options (Sheet 2 of 3)*

Option	Description
-ix	<p>Import Incomplete procedures with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>i</i> for Incomplete</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-ii</code> imports Incomplete procedures as Incomplete.</p> <p><b>Note:</b> If you choose <code>-ii</code> but an Incomplete or Unreleased version of a procedure already exists, the option is ignored and that procedure is imported as Withdrawn instead.</p>
-l	Overwrite lists that already exist.
-mx	<p>Import Model procedures with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>m</i> for Model</li> <li>• <i>r</i> for Released</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-mr</code> imports Model procedures as Released.</p> <p><b>Note:</b> If you choose <code>-mm</code> but a Model version of a procedure already exists, or <code>-mr</code> and a Released version already exists, the option is ignored and that procedure is imported as Withdrawn instead.</p>
-n	Opt for new version at the "New version or New name" prompt.
+n <i>name</i>	Opt for new name at the "New version or New name" prompt and use the name supplied.
-q	<p>Import procedures using the same procedure names (as new versions), owners and release status (except for Incomplete procedures) with which they were exported, without having to complete the associated prompts.</p> <p><b>Note:</b> This option is equivalent to using the <code>-d</code>, <code>-n</code>, <code>-mm</code>, <code>-rr</code> and <code>-uu</code> options.</p>
-rx	<p>Import Released procedures with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• <i>m</i> for Model</li> <li>• <i>r</i> for Released</li> <li>• <i>w</i> for Withdrawn</li> </ul> <p>For example, <code>-rr</code> imports Released procedures as Released.</p> <p><b>Note:</b> If you choose <code>-rm</code> but a Model version of a procedure already exists, or <code>-rr</code> and a Released version already exists, the option is ignored and that procedure is imported as Withdrawn instead.</p>
-t	Do not import/overwrite iProcess table data.
+t	Import/overwrite iProcess table data.

Table 9 *swutil IMPORTLIB options (Sheet 3 of 3)*

Option	Description
-ux	<p>Import Unreleased procedures with status <i>x</i>, where <i>x</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• u for Unreleased</li> <li>• w for Withdrawn</li> </ul> <p>For example, -uu imports Unreleased procedures as Unreleased.</p> <p><b>Note:</b> If you choose -uu but an Incomplete or Unreleased version of a procedure already exists, the option is ignored and that procedure is imported as Withdrawn instead.</p>
-v	<p>Displays messages showing if a procedure has been committed to the database or rolled back. See <a href="#">Display Commit/Rollback Messages When Importing Procedures or Procedure Libraries on page 72</a>.</p>



For any options that you do not set or suppress, you will be prompted in the same way as when you use `swutil IMPORT` interactively.

The procedure library is imported to the target library in Procedure Manager.

### Example

To import a procedure library from the `admin.xfr` file into the `Dept\Purchasing` library where you want all released procedures to be imported as model procedures and you want to suppress all prompts, you would use the following command:

```
swutil IMPORTLIB admin /dept/purchasing -deglrm
```

## Display Commit/Rollback Messages When Importing Procedures or Procedure Libraries

`swutil IMPORT -v` and `IMPORTLIB -v` can be used so that messages showing whether procedures have been committed to the database or rolled back are displayed. For example, if you import a procedure called `quota` that already exists:

---

```
swutil IMPORT quota -vrwtden
Importing a Procedure
Checking Procedure 'QUOTA':
A 'model' procedure already exists; so importing as 'withdrawn'.
**** COMMIT(&publish): Procedure:QUOTA
Unresolved issues from import of procedure quota, source file
D:\swserver\swnod730\util\quota.imp
```

---

Use the following commands to display commit/rollback messages when importing a procedure or library of procedures:

```
swutil IMPORTLIB library target [-v]
swutil IMPORT procedure [-v]
```

where:

- *library* is the name of the library to import.
- *target* is the full path to the library (as shown in Procedure Manager).
- *procedure* is the name of the `.xfr` or `.xpd1` file to import.
- `-v` (optional) can be used to display messages about whether a procedure is rolled back or committed to the iProcess database.

## Export a Procedure

---

Follow this procedure to export a procedure from Procedure Manager .

To export a procedure:

- you must have edit permissions to export a procedure (or sub-procedure).
- the *IPEADMIN* user or *IPEBACKGROUND* user can export all procedures and sub-procedures.

### Export a Procedure Interactively

The interactive method uses the `swutil` menu and displays prompts for each stage of the export. If you want to export a procedure without displaying all of the prompts, use the command line method. See [Export Procedures Using Command Line Options on page 76](#).

To export a procedure:

1. Select the following option from the `swutil` menu (see [Using swutil on page 2](#)):

4 - Export a Procedure

The following prompt is displayed:

---

```
Working, Please Wait .....
Exporting a Procedure
Name of Procedure (<CR> to quit?)
```

---

2. Enter the name of the procedure that you want to export (as displayed in Procedure Manager).
3. If multiple versions of the procedure exist, you need to specify which version you want to export. You can:
  - Choose - choose from a list of versions. See step 4.
  - Always choose - if you are exporting multiple procedures, select this to choose from a list for each procedure without displaying the question.
  - Same version for all - if you are exporting multiple procedures, you can specify a version number that will be used for all remaining procedures you export.
  - Default - select this option to use the default version.

4. Choose the version from a list that displays the available versions and their release status. For example:

---

There are 5 versions of this procedure:

0) 1.4 is the default version

1) 1.1 'm'odel /

2) 1.2

3) 1.3 'w'ithdrawn /

4) 1.4 'r'eleased / 'n'o preference /

5) 1.5 'u'nreleased /

Select r,u,m,w,n or index of desired procedure version 0 - 5 :0

---

Select the version of the procedure to export from the list shown. You can select a version either explicitly, by its version number, or based on its release status:

To select a specific version number, enter the version's index number - 0, 1, 2 or 3 in the example above.

To select a version by release status, enter the appropriate letter. In the example above:

— entering **r** selects the current Released version, which is Version 1.4.

— entering **u** selects the current Unreleased version, which is Version 1.5.

— entering **w** selects the *latest* Withdrawn version, which is Version 1.3.

The following table shows the full list of release status options that can be shown by the command.

*Table 10 Release Status Options of the Procedure Export Command*

Option	Description
<b>r</b>	Current Released version
<b>u</b>	Current Unreleased version
<b>m</b>	Current Model version
<b>i</b>	Current Incomplete version
<b>w</b>	Latest Withdrawn version. (Earlier Withdrawn versions cannot be selected by status. You must use an explicit version number to select them.)
<b>x</b>	Latest Incomplete/Withdrawn version
<b>n</b>	No preference

5. If the procedure being exported calls sub-procedures, you must specify the precedence order that will be used to determine the version of each sub-procedure that is exported. This is based on release status, using the same letter codes shown in the table above. For example:

---

```
Change default sub-procedure precedence order (rumiwxn)?
'Y'es change ('A'lways use new), 'N'o - default ('S'kip - always
default)
```

---

If you choose **Y**, you are prompted to change the precedence order:

---

```
What precedence order should be used for
exporting any sub-procedures found ?
```

```
'r'eleased|'u'nreleased|'m'odel|
'i'ncomplete|'w'ithdrawn|'x'|'n'
```

```
Enter order using key above (default is rumiwxn):
```

---



If you select an invalid status option, then the sub-procedure will not be exported. To overcome this, it is better to enter a range of status options and include 'n' for no preference. This means that even if the previous status options you have chosen are invalid, the sub-procedure will always be exported. See *TIBCO iProcess Modeler Procedure Management* for more information about precedence.

6. Choose whether or not to reset the procedure's extended properties. You can supply new values for each of the following (or leave them unchanged):
- `object author` is the name (or other identifier) of the iProcess user who created the object.
  - `release identification` is free text that you can use to identify the procedure as you wish.
  - `date object created` is the date and time that the procedure was created.
  - `date object modified` is the date and time that the procedure was last modified.

For each option, you can either respond with **Yes**, **No**, **A** (Yes and apply to all), **Skip** or **Quit**



You are not prompted to reset extended properties if the procedure is already OEM locked or is locked by another user.

7. Choose whether to change the procedure's major and/or minor version number. You can change the major version number, the minor version, or both.
8. Choose whether or not you want to OEM lock this procedure. (An OEM locked procedure cannot be renamed when it is imported.)



If the procedure you are exporting is already OEM locked you do not get this option.

9. If the procedure being exported calls sub-procedures, choose whether you want to export all, some or none of them. For each sub-procedure you export, steps 5 to 7 are repeated.

The procedure and any sub-procedures are exported to the file `SWDIR\util\procname.xfr`. If any use files are referenced, a copy of the file (in the iProcess `SWDIR\nodename.n\use` directory) is included in the exported file. Also, any VBA projects used in the procedure are exported.

## Export Procedures Using Command Line Options

To export one or more procedures using the `swutil EXPORT` command line options, use the following command:

```
swutil EXPORT [procedure] [directory] [options]
```

where:

- *procedure* is the name of the `.xfr` file to export.
- *directory* is the directory to export the procedure.

To export a procedure (for example, `carpool.xfr`) to a directory outside of `SWDIR/util` (for example, `/Dept/Purchasing`), run the following command:

```
swutil EXPORT carpool /Dept/Purchasing
```

If this is unspecified, the procedure is exported as a `.xfr` file to the `SWDIR/util` directory by default.

- *options* can be one or more of the following to suppress or change a specific option:

Table 11 *swutil EXPORT Options*

Option	Description
-a	Do not prompt for the Author (Author value remains the same).
+a <i>author</i>	Changes the value of Author to the supplied <i>author</i> value.

Table 11 *swutil* EXPORT Options (Cont'd)

Option	Description
-c	Do not prompt for the Creation Date.
+c "YY:MM:DD HH:MM"	Changes the value of the Creation Date to the supplied date.
-e	Do not prompt for the version of the procedure to export (use the current default version).
+e <i>version/precedence</i>	Do not prompt for the version of the procedure to export but use the specified version or precedence value, for example, +e 3.0 or +e rumiwxn. For more information about precedence, see <i>TIBCO iProcess Modeler Procedure Management</i> .
-l	Do not prompt for OEM Lock (and do not set it).
+l	Do not prompt for OEM Lock (and set OEM Lock).
-m	Do not prompt for the Modification Date (does not change).
+m "YY:MM:DD HH:MM"	Changes the value of the Modification Date to the supplied date.
-r	Do not prompt for the Release ID (does not change).
+r <i>relid</i>	Do not prompt for Release ID but change it to the value provided.
-s	Do not export referenced sub-procedures (default).
+s	Export referenced sub-procedures.
-t	Do not export iProcess tables.
+t	Export all iProcess tables used in the procedure being exported.
-v	Do not prompt to reset the version number (does not change).
+v <i>major.minor</i>	Do not prompt to reset the version number but set it to the supplied <i>major.minor</i> value.
-x	Do not prompt for the version of the sub-procedures to export.
+x <i>precedence</i>	Do not prompt for the version of the sub-procedure to export and use the supplied precedence value.



For any options that you do not set or suppress, you will be prompted in the same way as when you use `swutil EXPORT` interactively.

**Example**

To export a procedure called `APPLY1` where you want to define a new Author called `Fred` and a Release ID of `Release1` and want to suppress all other prompts by `swutil`, you can enter the following command:

---

```
swutil EXPORT apply1 +ar Fred Release1 -celmstvx
```

---

## Export a Procedure Library

---

Follow this procedure to export a procedure library listed in your Procedure Manager as an `.xfr` file. The hierarchical structure of the library is maintained when you export it.

This option is not available from the `swutil` menu. You can only export procedure libraries using the command line option either interactively or by using the options to suppress the prompts.

To export a procedure library:

- you must have edit permissions to export each object in the procedure library. Any objects that you do not have edit permissions for are not included in the `.xfr` file.
- the `IPEADMIN` user or `IPEBACKGROUND` user can export all procedure objects.

### Export a Procedure Library Interactively

To export a procedure library following the on screen prompts:

1. Use the following command:

```
swutil EXPORTLIB /pathname [directory]
```

where:

- *pathname* is the full path name of the library you want to export (as displayed in the Procedure Manager.)

For example, to export the Dept/Purchasing library, you would use the command:

```
swutil EXPORTLIB /dept/purchasing
```

- *directory* is the directory to export the procedure.

If this is unspecified, the procedure library is exported as a `.xfr` file to the `SWDIR\util` directory by default.

2. Choose whether or not to globally reset the extended properties (author, release ID, date created and date amended - see [step 6 of Export a Procedure Interactively on page 73](#)) for the procedure library and all of its sub-libraries.

If you do this, the new values are applied to each object in the library.

3. For each procedure and sub-procedure in the library, the command displays the same prompts that you would see if you were exporting the procedure individually. See [Export a Procedure on page 73](#) for more information.

The one difference is that if you chose to globally reset the extended properties (in [step 2](#)), you cannot reset the extended properties for individual procedures and sub-procedures. Those prompts are not displayed.

If any `use` files are referenced, a copy of the file (in the `iProcess SWDIR\nodename.n\use` directory) is included in the exported file. Also, any VBA projects used in the procedure are exported.

## Export a Procedure Library Using Command Line Options

Use the `swutil EXPORTLIB` command line options to quickly export a procedure library. By setting the required command line options, you can define the required export options and suppress the prompts/questions.

To export a procedure library:

1. Use the following command:

```
swutil EXPORTLIB pathname [directory] [options]
```

where:

- *pathname* is the full path name of the library you want to export (as displayed in Procedure Manager.) Use / to import the library to the root of Procedure Manager.
- *directory* is the directory to export the procedure.  
If this is unspecified, the procedure library is exported as a `.xfr` file to the `SWDIR\util` directory by default.
- *options* can be one or more of the options described below to suppress or change a specific option:

Table 12 *swutil EXPORTLIB Options*

Option	Description
-a	Do not prompt for the Author (Author value remains the same).
+a <i>author</i>	Changes the value of Author to the supplied <i>author</i> value.
-c	Do not prompt for the Creation Date.
+c "YY:MM:DD HH:MM"	Changes the value of the Creation Date to the supplied date.
-e	Do not prompt for the version of the procedure to export (use the current default version).
+e <i>version/precedence</i>	Do not prompt for the version of the procedure to export and use the specified version or precedence, for example, +e 3.0 or +e rumiwxn.

Table 12 *swutil EXPORTLIB Options (Cont'd)*

Option	Description
-l	Do not prompt for OEM Lock and do not set it.
+l	Do not prompt for OEM Lock and set the OEM Lock.
-m	Do not prompt for the Modification Date (does not change).
+m "YY:MM:DD HH:MM"	Changes the value of the Modification Date to the supplied date.
-p	Do not prompt for the reset properties and assume No.
+p	Do not prompt for the reset properties but assume Yes.
-r	Do not prompt for the Release ID and the current Release ID is not changed.
+r <i>relid</i>	Do not prompt for the Release ID but change it to the value provided.
-s	Do not export the referenced sub-procedures (default).
+s	Exports the referenced sub-procedures.
-t	Do not export iProcess tables.
+t	Export all iProcess tables used in the procedure being exported.
-v	Do not prompt to reset the version number of the library and do not change the current value.
+v <i>major.minor</i>	Do not prompt to reset the version number of the library but set it to the supplied <i>major.minor</i> version. <b>Note:</b> If the referred template is included in the exported library, the template version will also be set to the supplied <i>major.minor</i> version. If the template is not included, this command will not change the template's version number.
-x	Do not prompt for the version of the sub-procedure to export (use the current default).
+x <i>precedence</i>	Do not prompt for the version of the sub-procedure to export (use the supplied precedence setting).



For any options that you do not set or suppress, you will be prompted in the same way as when you use `swutil EXPORTLIB` interactively.

2. The procedure library is exported as a `.xfr` file in the `SWDIR\util` directory by default or the directory specified in the command.

**Example**

To export the `/Dept/Purchasing` library where you do not want any prompts to display but you need to set OEM lock for every procedure in the library, you would enter the following command:

---

```
swutil EXPORTLIB /dept/purchasing -acemprstvx +l
```

---

## Chapter 9 Procedure Versions

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil PROC RELEASE</code>	Release a given version of a procedure.
<code>swutil PROC WITHDRAW</code>	Withdraw a given version of a procedure.
<code>swutil PROC DELETE</code>	Delete a given version of a procedure.



For more information about procedure versions, see "Using Version Control" in *TIBCO iProcess Modeler Procedure Management*.

To run any of these commands you must be logged in as the `IPEADMIN` user.

## Release a Procedure Version

---

Use this command to release a given version of a procedure.

```
swutil PROC RELEASE name version comment [-m|-M [ALL|LIVE]]
```

where:

- *name* is the procedure (or sub-procedure) name.
- *version* is the version number of the procedure.
- *comment* is a text string which is displayed in the version control history for the procedure.

The *comment* string can be used to supply version-specific information, such as a description of the version, or any other desired information. It can be a maximum of 24 characters. If the string includes spaces, it should be enclosed in quotation marks.

- - m|-M [ALL|LIVE] (optional) migrates active cases associated with previously released versions of the procedure to the current version that you are releasing. (This option is case sensitive).



Before using the -m or -M options, TIBCO recommends that you perform a full system backup. This is because there is no undo feature after running this command.

If you use:

- the -m or -M parameter on its own, only cases that use the currently released version, that is, the version that will be automatically withdrawn by the Release operation, are migrated to the new version.
- the -m or -M parameter with the ALL option, all cases are migrated to the new version, no matter what previous version of the procedure they are using.
- the -m or -M parameter with the LIVE option, only the live cases are migrated to the new version.

The -M option operates in the same way as -m but does not interact with the iProcess background process (BG). Therefore, only use -M when iProcess Engine is not running.

Make sure that all existing cases are compatible with the new version of the procedure before you migrate them.

## Examples

This example releases version 1.1 of a procedure called `myproc` and adds the "This is released" comment.

---

```
swutil PROC RELEASE myproc 1.1 "This is released"
```

---

This example releases version 1.2 of a procedure called `myproc` and migrates all existing cases from previous versions to this new version:

---

```
swutil PROC RELEASE myproc 1.2 "Migrated cases" -m ALL
```

---

## Withdraw a Procedure Version

---

Use this command to withdraw the currently Released (or Model or Unreleased) version of a given procedure.

```
swutil PROC WITHDRAW name version comment
```

where:

- *name* is the procedure (or sub-procedure name).
- *version* is the version number of the procedure.
- *comment* is a text string which is displayed in the version control history for the procedure.

The *comment* string can be used to supply version-specific information, such as a description of the version, or any other desired information. It can be a maximum of 24 characters. If the string includes spaces, it should be enclosed in quotation marks.

### Example

This example withdraws version 1.1 of a procedure called `myproc` and adds the "This is withdrawn" comment.

---

```
swutil PROC WITHDRAW myproc 1.1 "This is withdrawn"
```

---

## Delete a Procedure Version

---



You must be logged in as the *IPEADMIN* user or *IPEBACKGROUND* user to run this command.

Use this command to delete a given version of a procedure. It uses the following syntax:

```
swutil PROC DELETE name version [-y]
```

where:

- *name* is the procedure (or sub-procedure name).
- *version* is the version number of the procedure. ALL means delete all versions of the procedure.
- *-y* suppresses the request for confirmation from the user.

### Example

The following example deletes version 1.1 of a procedure called *myproc*.

---

```
swutil PROC DELETE myproc 1.1
```

---

The following example deletes all versions of a procedure called *myproc*.

---

```
swutil PROC DELETE myproc ALL
```

---

## swutil PROC command errors

---

The following table lists the error codes that may be returned from the `swutil PROC` command:

*Table 13 swutil PROC Error Codes*

Error Code	Description
10	Invalid option (not RELEASE, WITHDRAW or DELETE)
11	Invalid procedure name
12	Invalid procedure version
13	Invalid procedure instance
14	Comment too large
15	Invalid number of parameters

## Chapter 10 **Users and Groups**

This chapter describes the `swutil USERINFO` command.

### Topics

---

- [Command Line Invocation, page 90](#)
- [Control File Format, page 91](#)
- [Control Instructions, page 92](#)
- [Export Format, page 96](#)
- [Security, page 98](#)

## Command Line Invocation



To run this command you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or *root* user.

```
swutil USERINFO EXPORT [username] [nolocation]
swutil USERINFO { - | filename } [nourschk]
```

where:

- `EXPORT [username]` instructs `swutil` to write user information to standard output, either for all users, or optionally for a single *username*.

The information is written in control file format (see [Control File Format on page 91](#)), enabling it to be directed to a file that may be read back in (with the above commands). This enables user information to be transferred to another computer, restored to a former state or updated after editing the file.

By default, the value of a user's `SW_DOMAIN` user attribute is output for an "add user" (U+) instruction, unless the *nolocation* parameter is used.

- *filename* is the name of the control file to be used to update user information from. (On Windows, *filename* can be a full pathname or a simple filename in the current directory. On UNIX, *filename* can be a full pathname or a simple filename in the `SWDIR\util` directory.)

By default, the list of users in *filename* is checked against the list of operating system users. Any user that does not exist in the list is ignored. This validation is not performed if the optional *nourschk* parameter is used. (*nourschk* has no effect if user validation is being performed with UVAPI).

- `-` (a hyphen character) instructs `swutil` to read control instructions from standard input and update user information. Control instructions must use control file format (see [Control File Format on page 91](#)).

During processing of a control file, error messages are written to standard output. On completion of processing, a message is written to standard output giving the total number of lines of instructions successfully processed.

The UNIX exit status is set to 0 for no errors, 2 if there are errors but some data is updated, or 3 if no data is updated.

Input and output redirection may be used as normal.

## Control File Format

---

The control file is a text file consisting of one or more control instructions, one to a line:

- A line starting with a semicolon (;) is treated as a comment line and ignored.
- Blank lines are ignored.
- Each instruction line starts with a single character code (for example: U), possibly followed by a modifier (for example: +).
- The code (plus optional modifier) may be followed by one or more fields separated by commas.
- White space following the code character (plus optional modifier), or following a field delimiter comma is ignored.
- Commas may be included in attribute values (since these are always the last field on the line).

The maximum line length is 512 characters.

## Control Instructions

---

The following control instructions can be used:

### Reset

This instruction (which should only appear as the first character in a control file) resets all user information: all users, attributes and groups are removed. It consists of the single character R.

### Change Attribute

This instruction adds or deletes an attribute applicable to all users. It consists of the single character A followed by either + to add, or - to delete:

*A-attribute*

Delete *attribute*.

*A+attribute, type [ , length ] , changer , [ default ]*

Add an attribute, where:

- *attribute* is the name of the new attribute.
- *type* is T for Text, N for Numeric, D for Date or M for Time.
- *length* is the length of the field for Text or Numeric fields; decimal numerics should have the number of decimal places after a period.
- *changer* specifies who can change the values for this attribute: U for the user the attribute applies to, or S for the user or group's supervisor, (that is, the username in the user or group's SUPERVISOR attribute), or blank for neither. The IPEADMIN user or IPEBACKGROUND user can change any attribute.
- *default* specifies an initial value of this attribute for all users.

For example:

---

```
A+MENUNAME, T, 10, , USER
A+WEIGHT, N, 5.2, U, 0.00
A+DEPT, T, 10, S, DEPT
```

---

## Change User

This instruction adds or deletes a user. It consists of the single character U followed by either + to add, or - to delete the user:

U-*name*

Delete the *name* user.

U+*name*[@*location*]

Add the *name* user.

If iProcess Engine is running on a Windows system, you can optionally include a *location* (a single valid machine name or domain name) which will be used to check if the user is a valid Windows operating system user. See [Validating New Users on a Windows iProcess Engine](#).

*location* is ignored on UNIX platforms or when using UVAPI to perform user validation.

The new user is given default attribute values, which may subsequently be changed with the change attribute value instruction.

For example:

---

```
U-keith
U+peter
U+fred@EMEA
```

---

## Validating New Users on a Windows iProcess Engine

Unless an alternative user validation system has been implemented using the UVAPI, an iProcess Engine user must also exist as an operating system user. On Windows, when iProcess Engine processes a U+ instruction to add a new user, it will only create the user if it can find the named *user* on the following search path:

1. the location specified in the *location* parameter in the U+ instruction.
2. the location defined by the value of the LOGON\_OS\_LOCATION process attribute.
3. the search path provided by the Windows LookupAccountName function.

If the *user* cannot be found in any of these locations, the add user instruction fails.

For more information about:

- UVAPI, see *TIBCO iProcess User Validation API User's Guide*.
- LOGON\_OS\_LOCATION, see "Administering Process Attributes" in *TIBCO iProcess Engine Administrator's Guide*.

## Change Group

This instruction adds or deletes a group. It consists of the single character G followed by either + to add, or - to delete:

*G-group*

Delete *group*.

*G+group*

Add *group*.

The new group is given default attribute values, which may subsequently be changed with the change attribute value instruction.

For example:

---

```
G+c1erks
G-admin
```

---

## Change Attribute Value

This instruction enables the attributes of an existing user or group to be changed. It consists of the single character v followed by either + to set an attribute to a new value, or - to reset an attribute to its default value. Both the user/group and the attribute must already exist.

*V+name, attribute, value*

Change the *attribute* belonging to user or group *name* to *value*.

*V-name, attribute*

Reset the *attribute* belonging to user or group *name* to its default.

For example:

---

```
V+fred,DESCRIPTION,Fred Bloggs
```

---



As well as the *IPEADMIN* user or *IPEBACKGROUND* user, a user can change their own attributes if the attribute has the U flag set. Also the supervisor of a user or group can change attributes with the S flag set.

## Change Membership

This instruction adds or removes a user from a group.

*M+user ,group*

Add *user* to *group*.

*M-user ,group*

Remove *user* from *group*.

For example:

---

*M+keith ,clerks*

*M-peter ,admin*

---



An iProcess user can have access to a maximum of 32,767 work queues. If this number is exceeded, users cannot log in. Users can access personal, group and test queues, either directly or by supervision or participation.

For more information, see "How Many Work Queues Can an iProcess User Access?" in *TIBCO iProcess Workspace (Windows) Manager's Guide*.

## Export Format

---

The output of the `swutil USERINFO EXPORT` command is in a format that can be re-imported. It starts with an `R` command to reset the user information, followed by sections for each type of data, for example:

---

```
R

; START OF ATTRIBUTES -----
A+DESCRIPTION,T,24,,
A+LANGUAGE,T,24,,
A+MENUNAME,T,24,,
A+QSUPERVISORS,T,48,,
A+SORTMAIL,T,24,,
A+SW_DOMAIN,T,24,,
A+USERFLAGS,T,24,,
; END OF ATTRIBUTES -----

; START OF GROUPS -----
G+clerks
V+clerks,DESCRIPTION,Clerical Workers
V+clerks,MENUNAME,sw_group

; END OF GROUPS -----

; START OF USERS -----
U+fred@EMEA
V+fred,DESCRIPTION,System Administrator
V+fred,LANGUAGE,english
V+fred,MENUNAME,ADMIN
V+fred,QSUPERVISORS,
V+fred,SORTMAIL,PROCEDURE
V+fred,SW_DOMAIN,EMEA
V+fred,USERFLAGS,

; END OF USERS -----

; START OF GROUP MEMBERSHIP -----
M+fred,clerks

; END OF GROUP MEMBERSHIP -----
```

---

The export format for a single user export command starts with a `U-` instruction to delete the user, followed by instructions to add it back again together with its other data, enabling a single user's data to be updated. For example, `swutil USERINFO EXPORT fred` could give the following:

---

```
U-fred
U+fred@EMEA
V+fred,DESCRIPTION,System Administrator
```

```
V+fred,LANGUAGE,english
V+fred,MENUNAME,ADMIN
V+fred,QSUPERVISORS,
V+fred,SORTMAIL,PROCEDURE
V+fred,SW_DOMAIN,EMEA
V+fred,USERFLAGS,
V+fred,WISCACHE,
```

---



By default, the value of a user's `SW_DOMAIN` attribute is included in an "add user" `U+` instruction. For example:

```
U+fred@EMEA
```

This value will be used to validate the user as a Windows operating system user if the instruction is used to re-import the user to iProcess Engine. See [Validating New Users on a Windows iProcess Engine on page 93](#).

However, if the `EXPORT` command included the `nolocation` option, the `SW_DOMAIN` attribute value is not included. For example:

```
U+fred
```

The `SW_DOMAIN` attribute value is always included in the "change attribute value" `V+` instruction, even if the `EXPORT` command included the `nolocation` option. For example:

```
V+fred,SW_DOMAIN,EMEA
```

For more information about the `SW_DOMAIN` user attribute, see "Defining Where a User's Password Should be Validated" in *TIBCO iProcess Workspace (Windows) Manager's Guide*.

## Security

---

Each time the utility is run, a log entry is created in the `SWDIR\logs` directory in the `userinfo.log` file, which includes the date and time that it happened.



If you have changed the log files directory in the `staffpms` file, all log entries will be created in the `userinfo.log` file located in the directory you have specified.

## Chapter 11 **Roles**

This chapter describes the `swutil ROLEINFO` command.

### Topics

---

- [Command Line Invocation, page 100](#)
- [Control File Format, page 101](#)
- [Control Instructions, page 102](#)
- [Example Control File, page 103](#)

## Command Line Invocation

---



To run any of these commands you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or *root* user.

`swutil ROLEINFO filename`

Update role information from control file *filename*. (On Windows, *filename* can be a full pathname or a simple filename in the current directory. On UNIX, *filename* can be a full pathname or a simple filename in the *SWDIR\util* directory.)

`swutil ROLEINFO -`

Read control instructions from standard input and update role information.

`swutil ROLEINFO EXPORT [rolename]`

Write role information to standard output; either for all roles, or just for optional *rolename*.

The information is written in control file format (see below), enabling it to be directed to a file which may be read back in (with the above commands). This enables role information to be transferred to another computer, restored to a former state or updated after editing the file.

`swutil MOVESYSINFO`

Update data from the temporary area. This is necessary to make changes to role data available to *iProcess*.

During processing of a control file, error messages are written to standard output. On completion of processing, a message is written to standard output giving the total number of lines of instructions successfully processed.

The UNIX exit status is set to 0 for no errors, 2 if there are errors but some data is updated, or 3 if no data is updated. Input and output redirection may be used as normal.

## Control File Format

---

The control file is a text file consisting of one or more control instructions, one to a line:

- A line starting with a semicolon (;) is treated as a comment line and ignored.
- Blank lines are ignored.
- Each instruction line starts with a single character code (for example: 0), possibly followed by a modifier (for example: +).
- The code (plus optional modifier) may be followed by one or more fields separated by commas.
- White space following the code character (plus optional modifier), or following a field delimiter comma is ignored.
- The maximum line length is 512 characters.

## Control Instructions

---

The following control instructions can be used:

Table 14 *swutil ROLEINFO Control Instructions*

Instruction	Description
R	(Reset) All roles are deleted. (This instruction should only appear as the first in a control file.)
<i>O+role , user</i>	Add new <i>role</i> with associated <i>user</i> . (If <i>role</i> already exists a warning message is issued.) <i>user</i> can be an iProcess user or group name.
<i>O=role , user</i>	Existing <i>role</i> is updated with <i>user</i> .
<i>O-role</i>	<i>role</i> is deleted.

## Example Control File

---

---

```
O+manager,paul
; New role 'manager' added with user 'paul
; allocated to it
O=director,justin
; Existing role 'director' updated to user
; 'justin'
O-executive
; Role 'executive' deleted
```

---



## Chapter 12 **Queue Participation**

This chapter describes the `swutil PARTICIPATION` command.

### Topics

---

- [Command Line Invocation on page 106](#)
- [Control File Format on page 107](#)
- [Control Instructions on page 108](#)
- [Example Control File on page 110](#)

## Command Line Invocation

---



To run any of these commands you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or root user.

`swutil PARTICIPATION filename`

Update queue participant records from control file *filename*. (On Windows, *filename* can be a full pathname or a simple filename in the current directory. On UNIX, *filename* can be a full pathname or a simple filename in the *SWDIR\util* directory.)

`swutil PARTICIPATION -`

Read control instructions from standard input and update queue participant records.

`swutil PARTICIPATION EXPORT`

Write queue participant records to standard output.

The information is written in control file format (see [Control File Format on page 107](#)), enabling it to be directed to a file which may be read back in (with the above commands). This enables queue participant records to be transferred to another computer, restored to a former state or updated after editing the file.

During processing of a control file, error messages are written to standard output. On completion of processing, a message is written to standard output giving the total number of lines of instructions successfully processed.

The UNIX exit status is set to 0 for no errors, 2 if there are errors but some data is updated, or 3 if no data is updated. Input and output redirection may be used as normal.

## Control File Format

---

The control file is a text file consisting of one or more control instructions, one to a line:

- A line starting with a semicolon (;) is treated as a comment line and ignored.
- Blank lines are ignored.
- Each instruction line starts with a single character code (for example: P), possibly followed by a modifier (for example: +).
- The code (plus optional modifier) can be followed by one or more fields separated by commas.
- White space following the code character (plus optional modifier), or following a field delimiter comma is ignored.
- The maximum line length is 256 characters.

## Control Instructions

---

The following control instructions can be used:

### Reset

This instruction (which should only appear as the first character in a control file) removes all queue participant records. It consists of the single character R.

### Reset Queue Participants

This instruction removes all queue participant records for a specific queue *queuename*.

P-*queuename*

### Add Participants

This instruction adds a participant record for a specific queue. It is a multi-line instruction which takes the format:

```
P+queuename , startdate , enddate , starttime , endtime , days
U+username
...
END
```

where:

- *queuename* is the name of the queue to add a participant record to.
- *startdate* is the date from which the participants have access to the queue, in the format *DD\_MM\_YYYY*. A hyphen character (-) is used, if no date is specified.
- *enddate* is the date from which the participants do not have access to the queue, in the format *DD\_MM\_YYYY*. A hyphen character (-) is used, if no date is specified.
- *starttime* is the time from which the participants have access to the queue on each day, in the format *HH:MM*. A hyphen character (-) is used, if no time is specified.

- *endtime* is the time from which the participants do not have access to the queue on each day, in the format *HH:MM*. A hyphen character (-) is used, if no time is specified.



If a start date/time is not specified, participants are granted access to the queue with immediate effect. If an end date/time is not specified, participants are granted access to the queue permanently.

- *days* lists the days of the week on which the participants have access to the queue, in the *MTWTFSS* format. A hyphen character (-) means that the participants do not have access to the queue on that day.

For example, the *M-W--S* entry indicates that the participants can access the queue only on Mondays, Wednesdays and Sundays.

- *U+username* adds user *username* to the participant list for this record. There should be one entry per participant, with each entry on a separate line.



An iProcess user can have access to a maximum of 32,767 work queues. If this number is exceeded, users cannot log in. Users can access personal, group and test queues, either directly or by supervision or participation.

For more information, see "How Many Work Queues Can an iProcess User Access?" in *TIBCO iProcess Workspace (Windows) Manager's Guide*.

## Example Control File

---

```
; Delete all current participant records.
```

```
R
```

```
; Give auser1 access to swadmin queue every day starting from 01/12/1998.
```

```
P+swadmin,1_12_1998,-,-,-,MTWTFSS  
U+auser1  
END
```

```
; Give auser1 access to queue fred on Mondays only for a specific time period.
```

```
P+fred,1_1_1999,12_1_1999,09:00,17:00,M-----  
U+auser1  
END
```

```
; Give auser2 access to queue fred on Mondays to Fridays.
```

```
P+fred,-,-,-,-,MTWTF--  
U+auser2  
END
```

---

## Chapter 13 **Queue Redirection**

This chapter describes the `swutil REDIRECTION` command.

### Topics

---

- [Command Line Invocation, page 112](#)
- [Control File Format, page 113](#)
- [Control Instructions, page 114](#)
- [Example Control File, page 115](#)

## Command Line Invocation

---



To run any of these commands you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or root user.

`swutil REDIRECTION filename`

Update queue redirection information from control file *filename*. (On Windows, *filename* can be a full pathname or a simple filename in the current directory.

On UNIX, *filename* can be a full pathname or a simple filename in the *SWDIR\util* directory.)

`swutil REDIRECTION -`

Read control instructions from standard input and update queue redirection information.

`swutil REDIRECTION EXPORT`

Write queue redirection information to standard output.

The information is written in control file format (see [Control File Format on page 113](#)), enabling it to be directed to a file which may be read back in (with the above commands). This enables queue redirection information to be transferred to another computer, restored to a former state or updated after editing the file.

During processing of a control file, error messages are written to standard output. On completion of processing, a message is written to standard output giving the total number of lines of instructions successfully processed.

The UNIX exit status is set to 0 for no errors, 2 if there are errors but some data is updated, or 3 if no data is updated. Input and output redirection may be used as normal.

## Control File Format

---

The control file is a text file consisting of one or more control instructions, one to a line:

- A line starting with a semicolon (;) is treated as a comment line and ignored.
- Blank lines are ignored.
- Each instruction line starts with a single character code (for example: F), possibly followed by a modifier (for example: +).
- The code (plus optional modifier) can be followed by one or more fields separated by commas.
- White space following the code character (plus optional modifier), or following a field delimiter comma is ignored.
- The maximum line length is 256 characters.

## Control Instructions

---

The following control instructions can be used:

### Reset

This instruction (which should only appear as the first character in a control file) removes all current queue redirection information. It consists of the single character R.

### Reset Queue Redirection

This instruction removes all queue redirection information for a specific queue *queuename*.

*F-queuename*

### Redirect Queue

This instruction redirects a specific queue to another queue:

*F+queuename , destname , startdate , enddate*

where:

- *queuename* is the name of the queue from which work items are to be redirected.
- *destname* is the name of the queue to which work items for *queuename* should be redirected. Queues on remote nodes can be specified as *name@nodename*. (You should make sure that any procedure which may send work items to the redirected queue is multi-noded to the *nodename* node. If it is not, work items sent by that procedure will not be able to reach the destination queue, and so will be delivered to the undelivered work items queue instead.)
- *startdate* is the date and time from which redirection should take place, in the format *DD\_MM\_YYYY HH:MM*. If no date is specified, the current date and time is used. A hyphen character (-) is used if no date/time is specified.
- *enddate* is the date and time at which redirection should stop, in the format *DD\_MM\_YYYY HH:MM*. If no date is specified, a date and time of 31/12/3000 23:59 is used. A hyphen character (-) is used if no date/time is specified.

## Example Control File

---

---

```
; Delete all current redirection information.
```

```
R
```

```
; Redirect queue swadmin to queue adrian for specific date range
```

```
F+swadmin,adrian,20_12_1998 09:00,31_12_1998 12:00
```

```
; Redirect queue auser1 to queue fred
```

```
F+auser1,fred,23_12_1998 00:00,4_1_1999 23:59
```

---



## Chapter 14 Lists

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil IMPORT</code>	Import a <code>.xfr</code> file containing an iProcess List.
<code>swutil EXPLIST</code>	Create a <code>.xfr</code> file containing an iProcess List.



To run any of these commands you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or *root* user.

## Import a List

---

Use this command to import a list definition file (`.xfr`) file from the `SWDIR/util` directory. A list definition file can be created from a procedure using `swutil EXPLST`.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil IMPORT
```

The utility prompts you for the name of the `.xfr` file to be imported.

You need to log out and log back in again to TIBCO iProcess Workspace before the list you have imported will be visible.

## Export a List

---

Use this command to create a file containing an iProcess List in a format suitable for installation on another system.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil EXPLST
```

The utility prompts you for the name of the list to be exported.

The list is exported to the file `SWDIR/util/listname.xfr` by default.



## Chapter 15 iProcess Tables

This chapter describes the following `swutil` commands:

Command	Description
<code>swutil EXTTAB</code>	Print a list of the fields for a selected iProcess table and/or iProcess table data.
<code>swutil DELTAB</code>	Delete iProcess table data and/or the iProcess table structure.
<code>swutil IMPORT</code>	Import a <code>.xfr</code> file containing an iProcess table
<code>swutil EXPTAB</code>	Create a file containing an iProcess table.



To run any of these commands you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or *root* user.

## Extract a Table

---

You can use this command to print a list of the fields for a selected iProcess table, optionally including the data associated with those fields. You can also choose to export the fields to a file.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil EXTTAB
```

The utility prompts you for the name of the table to print a list of fields for, optionally including the data associated with those fields.

The following example illustrates the output if you elect to disregard the table data or if no data is present in the table.

---

```
Table structures
```

```
Table: SOFTWARE (Dataspace 30 bytes)
```

```
Field nameTypeLengthDP
```

```
SITEM Text18
SPRICE Numeric80
```

---

This example illustrates the output with data.

---

```
Table structures and data
```

```
Table: HARDWARE (Dataspace 30 bytes)
```

```
Field nameTypeLengthDPData
```

```
HITEM Text18 Tiny machine
HPRICE Numeric80 995
```

```
HITEM Text18 Medium machine
HPRICE Numeric80 1995
```

```
HITEM Text18 Large machine
HPRICE Numeric80 5995
```

---

The Dataspace bytes indicates the total amount of space that will be allocated to any one case to hold data from the selected table.

The Length of a field indicates the maximum length of that field; the table data for that field may not exceed this amount.

## Delete a Table

---

Use this command to delete all data associated with a selected iProcess table. Optionally, the structure may also be deleted.



This only deletes iProcess tables and does not have any effect on tables in the iProcess database instance.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil DELTAB
```

The utility prompts you for the name of the table to be deleted.

## Import an iProcess Table

---

Use this command to import a `.xfr` file containing an iProcess table from the `SWDIR/util` directory. A `.xfr` file can be created from a table using `swutil EXPTAB`.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil IMPORT
```

The utility prompts you for the name of the `.xfr` file to be imported. If the table already exists on the target server, the utility prompts you to overwrite the data in the existing table with the data from the table you are importing. Select `y` or `n` depending on your requirements.

You need to log out and log back in again to TIBCO iProcess Workspace before the table you have imported will be visible.

## Export a Table

---

Use this command to create a file containing an iProcess table in a format suitable for installation on another system. For example, you may use a table at one location and other offices around the world need to use it.

You can either select this utility from the `swutil` menu (see [Using swutil on page 2](#)), or use the following command:

```
swutil EXPTAB
```

The utility prompts you for the name of the table to be exported, and whether the table data (as well as the structure) is to be exported.

The table is exported to the file `SWDIR/util/tablename.xfr`.



# Moving System Information

This chapter describes the following `swutil` command:

Command	Description
<code>swutil MOVESYSINFO</code>	Initiates a data update.



To run this command you must be logged in either as the *IPEADMIN* user or, on a UNIX system, as the *IPEBACKGROUND* user or *root* user.

## Move System Information

---

The following command instructs the iProcess background process to perform an instruction. This is a command line option only.



This command requests confirmation from the user (with an “Are you sure you want to do this” message), but this may be suppressed with a `-y` flag on the end of the command line enabling it to be run from a batch file.

This command line initiates a data update from the temporary iProcess storage area, making changes in user data available to iProcess Suite:

```
swutil MOVESYSINFO
```

## Chapter 17 **Batch Processing Cases**

This chapter describes the `SWDIR\util\swbatch` utility command which enables you to process large numbers of cases as a single batch operation.

### Topics

---

- [Overview, page 130](#)
- [Running swbatch, page 131](#)
- [swbatch Commands, page 132](#)

## Overview

---

The `swbatch` utility enables you to process any number and combination of the following functions in a single batch operation:

- Start a case, with or without associated case data.
- Issue an event for a case, with or without associated case data.
- Close one or more cases.
- Purge one or more cases.



The `swbatch` utility is optimized to perform these operations on a large number of cases as a single batch. To perform these operations on an individual basis you may find it easier to use the `swutil` utility instead. See [Chapter 1, Using the swutil Utility, on page 1](#).

The `swbatch` utility is a command-line utility:

- Commands can be presented to `swbatch` either from a command file or from standard input, for example, as redirected output from another process.
- Case data can be embedded in the command file or input stream, in `abox` format.



Each of the requests processed by `swbatch` is performed as a separate transaction. If multiple cases are to be processed, such as when using `PURGE` or `CLOSE`, then these are split up into a number of transactions.

## Running swbatch

---

To run `SWDIR\util\swbatch`, enter the following command from a command prompt:

```
SWDIR\util\swbatch [-v] {command_file|-}
```

where:

- `-v` indicates verbose operation.
- `command_file` is the name of a file containing the swbatch commands to be processed. Enter a hyphen (-) if the commands are to be read from standard input instead.



To run this command, you must log in as the `IPEADMIN` user.

### Command File Format

The command file (or redirected input) must contain one or more iterations of the following:

```
command  
[case_data]  
[STOP]
```

where:

- `command` is an appropriately formatted swbatch command. The available commands are described in [swbatch Commands on page 132](#).
- `case_data` is embedded case data for a `START` or `EVENT` command, in `abox` format. For more information about `abox` format, see "Using iProcess Abox Files" in *TIBCO iProcess Modeler Integration Techniques*.
- `STOP` terminates an embedded `case_data` list.

## swbatch Commands

---

This section describes the available `swbatch` commands:

- [Start a Case, page 133](#)
- [Issue an Event, page 136](#)
- [Close a Case, page 138](#)
- [Purge a Case, page 139](#)

## Start a Case

To start a case, with or without associated case data, use the following command:

**Syntax** `START [version] [-sN] [-c] -e|-a[D]datafile] [-P priority] <node|-> procname  
<stepname|-> casedesc`

where:

- *version* specifies the version of the procedure that you want to start a case of. Specify *one* of the following flags:

Flag	Description
-r	Start a (live) case of the Released version.
-u	Start a (test) case of the Unreleased version.
-m	Start a (test) case of the Model version.
-vX.Y	Start a case of Version X.Y, where X is the major version number and Y is the minor version number.

If *version* is not specified, the following default precedence order is used to select the version to use:

Released > Unreleased > Model

Using a Released version starts a live case. Using an Unreleased or Model version starts a test case.

- *-sN* specifies the precedence order to be used to select the version of any sub-procedure started by the case. Specify one of the following values:

Value	Description
1	Released version only
2	Unreleased > Released version
3	Model > Released version
4	Unreleased > Model > Released version
5	Model > Unreleased > Released version

If the `-sN` flag is not specified, the following default precedence order is used to select the version to use:

Released > Unreleased > Model

Using a Released version starts a live sub-case. Using an Unreleased or Model version starts a test sub-case.



Note: For more information about procedure versions see "Using Version Control" in *TIBCO iProcess Modeler Procedure Management*.

- `-c` specifies that the command returns the case number of the started case
- `-e` indicates that case data is embedded in the command file or input stream. See [Command File Format on page 131](#).
- `-a[D] datafile` indicates that case data is supplied in abox format in the *datafile*. For more information about *abox* format, see "Using iProcess Abox Files" in *TIBCO iProcess Modeler Integration Techniques*.

If the `D` flag is specified, the *datafile* is deleted after it has been processed.

- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.
- *node* is the name of the host node, if the procedure is hosted remotely. Enter a hyphen (-) if the procedure is hosted locally.
- *procname* is the name of the procedure.
- *stepname* is the step at which to start the case. Enter a hyphen (-) to specify the default start step.
- *casedesc* is the case description of the case to be started.

**Notes** When all the `START` commands succeed, the return code from `SWBATCH` is `SW_SUCCESS`. If a `SWBATCH` command fails, `SWBATCH` will continue to execute the subsequent commands, and the return code will be the same as that of the first failure command.

If you use a syntactically incorrect `START` command, the utility may not return an error message (in verbose mode) indicating that it could not process the command.

## Examples

1. This example starts a case of procedure `HIRING` on the local node, with case description `test1`. No case data is supplied.

---

```
START - hiring - test1
```

---

2. This example starts a case of procedure `HIRING`. Case data is supplied in the `test1.txt` file, which is deleted when the command has been processed.

---

```
START -aD test1.txt - hiring - test1
```

---

3. This example starts a case of procedure `HIRING` on the local node, with case description `app123`. Case data for the `FIRST` and `LAST` (name) fields is embedded after the command, and terminated by the use of the `STOP` line. See [Command File Format on page 131](#).

---

```
START -e - hiring - app123  
FIRST,George  
LAST,Smith  
STOP
```

---

## Issue an Event

---

To issue an event, with or without associated case data, use the following command:

**Syntax** `EVENT [e[p][r|R]|-a[D][p][r|R] datafile] [-P priority] <node|-> procname casenum eventname`

where:

- `-e` indicates that case data is embedded in the command file or input stream. See [Command File Format on page 131](#). The following options can also be used when `-e` is specified:

Option	Description
r	Any deadlines for the case and case steps will be recalculated using the new field values supplied. See "Dynamically Recalculating Deadlines" in <i>TIBCO iProcess Modeler Basic Design</i> for more information about using this option on a step. See "Case Deadline Setting" in <i>TIBCO iProcess Modeler Procedure Management</i> for more information about using this option on a case.
R	Any deadlines for the case, sub-cases, case steps, and sub-case steps will be recalculated using the new field values supplied. See "Dynamically Recalculating Deadlines" in <i>TIBCO iProcess Modeler Basic Design</i> for more information about using this option on a step. See "Case Deadline Setting" in <i>TIBCO iProcess Modeler Procedure Management</i> for more information about using this option on a case.
p	The new field values supplied will also be copied to all outstanding work items. This is analogous to using the <code>-p datafile</code> flag with the <code>swutil EVENT</code> command. See <a href="#">Issue an Event on page 24</a> for more information about the effect of this flag.

- `-a` indicates that case data is supplied in abox format in *datafile*. For more information about abox format, see "Using iProcess Abox Files" in *TIBCO iProcess Modeler Integration Techniques*. The following options can also be used when `-a` is specified:

Option	Description
D	The <i>datafile</i> is deleted after it has been processed.

Option	Description
p	The new field values supplied in the <i>datafile</i> will also be copied to all outstanding work items. This is analogous to using the <code>-p datafile</code> flag with the <code>swutil EVENT</code> command. See <a href="#">Issue an Event on page 24</a> for more information about the effect of this flag.
r	Any deadlines for the case and case steps will be recalculated using the new field values supplied in the <i>datafile</i> .
R	Any deadlines for the case, sub-cases, case steps, and sub-case steps will be recalculated using the new field values supplied in the <i>datafile</i> . <b>Note:</b> Only the sub-case data is updated with <code>-p</code> option in the <code>swutil EVENT</code> command, the deadlines of sub-case and sub-case steps are recalculated.

- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.
- `node` is the name of the host node, if the procedure is hosted remotely. Enter a hyphen (-) if the procedure is hosted locally.
- `procname` is the name of the procedure, including sub-procedures.
- `casenum` is the case number.
- `eventname` is the name of the event.

**Example** This example issues an event called `trigger` for case 1 of procedure `HIRING`, hosted on the local node.

```
EVENT - hiring 1 trigger
```

This example similarly issues an event called `trigger` for case 1 of procedure `HIRING`, but it also:

- copies the new field values supplied in file `NewData1` to all outstanding work items.
- forces the recalculation of deadlines for the case and any sub-cases based on the new field values supplied in file `NewData1`.

```
EVENT -apR NewData1 - hiring 1 trigger
```

## Close a Case

---

To close one or more cases of a specific procedure, use the following command:

**Syntax** `CLOSE [-P priority] [-d] <node|-> procname caselist`

where:

- `-P priority` specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.
- `-d` closes cases without triggering events that are set for the `OnBeforeClose` event or the `OnAfterClose` event. The events are triggered before or after closing the case if running the `CLOSE` command without the `-d` option. For more information about the `OnBeforeClose` event and the `OnAfterClose` event, see *TIBCO iProcess Modeler Procedure Management*.
- `node` is the name of the host node, if the procedure is hosted remotely. Enter a hyphen (-) if the procedure is hosted locally.
- `procname` is the name of the procedure, including sub-procedures.
- `caselist` closes all cases in a given range. Use a comma (,) to separate case numbers and a hyphen (-) to specify a range. For example, the entry `2, 5-10` means case 2 and cases 5 to 10 inclusive. The caselist must not exceed 255 characters.



It is more efficient to close a number of cases using a list rather than running multiple close commands.

**Example** This example closes case 1 of procedure `HIRING`, hosted on the local node.

---

```
CLOSE - hiring 1
```

---

This example closes case 2 and cases 5 to 10 inclusive:

---

```
CLOSE - hiring 2,5-10
```

---

## Purge a Case

To purge one or more cases of a specific (locally hosted) procedure, use the following command:

**Syntax** PURGE [-P *priority*] [-d] <*node*|-> *procname* <*caselist*|ALL\_CASES|*dd\_mm\_yyyy*>

where:

- -P *priority* specifies the internal message queue priorities. For more information about message priorities, see "Message Prioritizing" in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide*.
- -d purges cases without triggering events that are set for the OnBeforePurge event. The events are triggered before purging the case if running the PURGE command without the -d option. For more information about the OnBeforePurge event, see *TIBCO iProcess Modeler Procedure Management*.
- *node* is the name of the remote host node. Enter a hyphen (-) if the procedure is hosted locally.
- *procname* is the name of the procedure, including sub-procedures.
- *caselist* purges all cases in a given range, whether they are open or closed. Use:
  - ALL\_CASES to specify all cases of the procedure.
  - a comma (,) to separate case numbers, and a hyphen (-) to specify a range. The caselist must not exceed 255 characters.

For example, the entry 2, 5-10 means case 2 and cases 5 to 10 inclusive..



It is more efficient to purge a number of cases using a list rather than running multiple purge commands.

- *dd\_mm\_yyyy* purges all cases terminated up to and including the specified date. The date is specified as the day, month and year numbers (in that order) separated by underscores.

**Notes** If you use a syntactically incorrect PURGE command, the utility may not return an error message (in verbose mode) indicating that it could not process the command.

Purging many cases can be made more efficient by setting the USE\_NEXT\_MESSAGE\_TO\_DEQUEUE process attribute to 1. This is only of benefit when there is a large build up of messages in the Oracle AQs. See "Message and Mbox Processing Configuration" in *TIBCO iProcess Engine Administrator's Guide* for more information.

**Example** This example purges all cases of procedure HIRING on the local node.

---

```
PURGE - hiring ALL_CASES
```

---

This example purges case 2 and cases 5-10 inclusive of procedure HIRING on the local node:

---

```
PURGE - hiring 2,5-10
```

---

# Index

## C

### Case

- Close [12](#)
- Purge [13](#)
- Reopen Closed [24](#)
- Start [133](#)

### Close Case [12](#)

- Closed Case, Reopen [24](#)
- customer support [xvii](#)

## D

### Delete

- Table [123](#)

## E

### Event, Invocation [24](#)

### Export

- List [59, 67, 119](#)
- Procedure [119, 125](#)
- Role Information [100, 106, 112](#)
- Table [119, 125](#)

### Extract

- Table [122](#)

## I

### Import

- List [59, 73, 118, 124](#)
- Procedure [59, 73, 118, 124](#)
- Table [59, 73, 118, 124](#)
- XFR file [59, 73, 118, 124](#)

## L

### List

- Export [59, 67, 119](#)
- Import [59, 73, 118, 124](#)

## M

- Move SysInfo [100, 128](#)

## P

### Procedure

- Export [119, 125](#)
- Import [59, 73, 118, 124](#)
- Purge Case [13](#)

## Q

- Queue, Resend [16](#)

**R**

- Reopen Closed Case [24](#)
- Resend Work Items [16](#)
- Role Information
  - Control File Format [41](#), [101](#), [107](#), [113](#)
  - Export [100](#), [106](#), [112](#)
  - Reset [102](#)

**S**

- Start Case [133](#)
- support, contacting [xvii](#)
- SWUTIL
  - CLOSE [12](#)
  - CSTART [6](#), [6](#)
  - EVENT [24](#)
  - EXPLST [119](#)
  - EXPORT [79](#), [80](#)
  - EXPTAB [125](#)
  - EXTCD [8](#), [9](#)
  - IMPORT [79](#), [124](#)
  - Menu [2](#), [2](#)
  - MOVESYSINFO [100](#), [128](#)
  - PURGE [13](#)
  - RESEND [16](#)
  - ROLEINFO [100](#), [106](#), [112](#)
- swutil
  - EXTTAB [122](#)

**T**

- Table
  - Delete [123](#)
  - Export [119](#), [125](#)
  - Extract [122](#)
  - Import [59](#), [73](#), [118](#), [124](#)
- technical support [xvii](#)
- TIBCO\_HOME [xii](#)

**U**

- User Information
  - Attribute [92](#)
  - Attribute Value [94](#)
  - Control File Format [91](#)
  - Export Format [96](#), [96](#)
  - Group [94](#)
  - Group Membership [95](#)
  - Instruction Codes [102](#)
  - Reset [92](#), [108](#), [114](#)
  - Security [98](#)
  - User [93](#)
- Utility
  - Menu [2](#)
  - Program [1](#), [105](#), [111](#)

**W**

- Work
  - Item, Resend [16](#)
  - Queue, Resend [16](#)

**X**

- XFR File
  - Create [59](#), [67](#), [119](#), [119](#), [119](#), [125](#), [125](#)
  - Import [59](#), [73](#), [118](#), [124](#)