TIBCO iProcess® Engine

System Messages Guide

Software Release 11.4.1
April 2014

Two-Second Advantage®

Important Information

Contents

Preface

This guide describes the system error and warning messages that can be returned by the TIBCO iProcess Engine in the *SWDIR*`\logs\sw_warn` and `sw_error` files.

Topics

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® 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*

Other TIBCO Product Documentation

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

- TIBCO ActiveMatrix BusinessWorks™
- 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
SWDIR	TIBCO iProcess Engine installs into a directory. This directory is referenced in documentation as *SWDIR*. The value of *SWDIR* depends on the operating system. For example, • on a Windows server (on the `C:` drive) if *SWDIR* is set to the `C:\swserver\staffw_nod1` directory, then the full path to the `swutil` command is in the `C:\swserver\staffw_nod1\bin\swutil` directory. • on a UNIX or Linux server if *SWDIR* is set to the `/swserver/staffw_nod1` directory, then the full path to the `swutil` command is in the `/swserver/staffw_nod1/bin/swutil` directory or the *$SWDIR*`/bin/swutil` directory. **Note**: On a UNIX or Linux system, the environment variable $*SWDIR* should be set to point to the iProcess system directory for the *root* and *swadmin* users.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms. For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
📝	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
💡	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
⚠	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`

Table 2 Syntax Typographical Conventions (Cont'd)

Convention	Use
`|`	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand param1 | param2 | param3`

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	`MyCommand {param1 param2} | {param3 param4}`
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	`MyCommand {param1 | param2} {param3 | param4}`
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	`MyCommand param1 [param2] {param3 | param4}`

Connecting with TIBCO Resources

How to Join TIBCOmmunity

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

How to Access TIBCO Documentation

You can access TIBCO documentation here:

http://docs.tibco.com

How to Contact TIBCO Support

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

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

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

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

  https://support.tibco.com

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

Chapter 1 Introduction

This chapter describes the format of messages in the *SWDIR*`\logs\sw_warn` or *SWDIR*`\logs\sw_error` files, the format of system message descripton, the use of debugging when trying to resolve problems, and monitoring the system log files.

Topics

Format of System Messages

iProcess system messages in the *SWDIR***logs\sw_warn** or *SWDIR***logs\sw_error** file have the following format:

```
YYYY/MM/DD HH:MM:SS.ssssss
(Process:Instance:PID:Severity:ProcName:CaseNum:StepName:User:Module:
Revision:Line):MessageID-MessageText
```

where:

- *YYYY/MM/DD HH:MM:SS.ssssss* is the date and time that the message was created, where *ssssss* denotes microseconds.
- *Process* is the iProcess process name.

See Server Processes in *TIBCO iProcess Engine: Administrator's Guide*, for a complete list of process names.

- *Instance* is the iProcess process instance number. Processes that do not run permanently, such as **swbatch**, show a value of **0**.
- *PID* is the operating system process identification.
- *Severity* is the severity code of the problem. See Message Categories on page 5 for a definition of the different categories.
- *ProcName* is the name of the procedure that the message applies to. (This is blank if not applicable to the current message.)
- *CaseNum* is the case number of the case that the message applies to. (This is blank if not applicable to the current message.)
- *StepName* is the name of the procedure step that the message applies to. (This is blank if not applicable to the current message.)
- *User* is the iProcess user that the process used to connect to iProcess - for example, **pro** (UNIX), **swpro** (Windows) or **swuser**.
- *Module* is the name of the source module (for TIBCO Software Inc. internal use only).
- *Revision*. The module revision number.
- *Line* is the line number in the source module that generated the error message (for TIBCO Software Inc. internal use only).
- *MessageID* is the numeric identifier of this message, as defined in the *SWDIR***etc****swerwarn.mes** file.

- *MessageText* is the error message text.

The following is an example message:

```
2008/04/16
09:53:10.012673(PROCMGR:2:6012:0:$SYSTEM:302:SYSINFO:pro:fillogin.c:69813:721):
1631-WARNING: <Login Daemon auto-logout> <Windows User sww.uid file not being
updated> <user23> <sww.uid file is 123 seconds old>.
```

Format of Message Descriptions

Chapter 2 lists all the messages that can be returned by the iProcess Engine in the *SWDIR***logs****sw_warn** and **sw_error** files. Messages are listed in ascending Message ID order.

The following table describes the information that is provided about each message.

Item	Description
Message ID	The unique message identifier, as displayed in the *SWDIR***logs****sw_warn** or **sw_error** file. **Note: Message IDs that are defined in the SWDIR\etc\swerwarn.mes file, but are not documented in Chapter 2, are not currently used by the iProcess Engine.**
Message Text	The message itself, as displayed in the *SWDIR***logs****sw_warn** or **sw_error** file.
Description	Information on what the problem is and its possible causes.
Appears In	Whether the message can appear in the *SWDIR***logs****sw_error** fie, *SWDIR***logs**\ **sw_warn** file, or in both.
Severity	A numeric value that categorizes the severity of the problem. See Message Categories on page 5 for a definition of the different categories.
Action	Information on what action to take to rectify the problem.
See Also	References to other pieces of iProcess user documentation that provide more information that is relevant to the message.

Message Categories

iProcess system messages are classified as either Critical, Severe, Warning or Information. The following table provides a definition of each of these categories.

Code	Severity	Description
1	**Critical**	**System down.** **Error must be resolved before the system can be restarted.**
2	**Severe**	**Potential data integrity problems, reduction in system performance, or system component is experiencing a serious problem.** **Urgent investigation and rectification of the error is required.**
3	Warning	Minor potential problems or reduction in system performance. Investigation and rectification of the error is recommended at the first suitable opportunity.
4	Information	System running normally. No remedial action required.

Debugging

IMPORTANT - For some system messages, the Action section recommends that you turn debugging on to assist in resolving the problem.

You may need to capture debug information either:

- for one or more iProcess Engine server processes or,
- for the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process. This process is called by the *SWDIR***bin\swutil** command.

Before using debugging, you should be aware of the following:

- Enabling debugging for iProcess Engine server processes can have a significant impact on system performance, particularly for processes such as the Background (**BG**), Work Queue Server (**WQS**) or Work Item Server (**WIS**).
- Enabling debugging for the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process means that *SWDIR***bin\swutil** commands will take longer to run. It will not affect the performance of the iProcess Engine server processes.
- Whenever possible, enable debugging at a time that will minimize the impact on system performance.
- Always disable debugging again immediately after you have obtained the necessary debug data.
- Debug files are created in the *SWDIR***logs** directory. Depending on the debug settings used, debug log files can be extremely large. Ensure that the system has adequate disk space available before enabling debugging, and remove the debug log files as soon as they are no longer needed.
- Use debugging only when you are explicitly advised to do so, either in this guide, or by TIBCO Support.

Enabling and Disabling Debugging

The following sections explain how to enable and disable debugging for each type of process.

Debugging iProcess Engine Server Processes

To enable or disable debugging for an iProcess Engine server process, use the *SWDIR***util\swadm** command to set the `DEBUG` process attribute, using the required debug string.

If the debugging is disabled, the `MEMSET` debug string will be applied to record the debug information to the `.dmp` log file, which is located in the *SWDIR*`\logs` directory. The default `MEMSET` debug strings for each iProcess Engine server process is as follows:

Process	Default Debug String
`BG`	`MEMSET,ALL=1,G=13,c=15,T=3,m=41,q=57`
`BGPREDICT`	`MEMSET,ALL=1,G=13,c=15,T=3,m=41,q=57`
`DLMGR`	`MEMSET,ALL=1,G=13,T=7`
`RPCBG`	`MEMSET,ALL=1,G=13,T=7,c=15,q=57`
`RPC_POOL`	`MEMSET,ALL=1,G=13,T=7,q=57`
`RPC_TCP_LI`	`MEMSET,ALL=1,G=13,T=7,q=57`
`RPC_UDP_LI`	`MEMSET,ALL=1,G=13,T=7,q=57`
`SPO`	`MEMSET,ALL=1,G=13,m=41,q=57,Q=5,v=5,E=5`
`WIS`	`MEMSET,ALL=1,G=13,m=41,R=5,w=5,Q=5,T=3`
`WISMBD`	`MEMSET,ALL=1,G=13,m=41,T=7,q=57,Q=5`
`WQS`	`MEMSET,ALL=1,G=13,q=57,T=5,R=5`
`PROCMGR`	`MEMSET,ALL=1,G=13,q=57,T=7`

See the individual system message descriptions for details of the debug string that is required when investigating a particular error.

See Administering Process Attributes in *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR***\util\swadm** command.

Debugging the staffifc.exe (Windows) Process

To enable debugging for the **staffifc.exe** process, put the required debug string in the following registry key:

> **HKLM/SOFTWARE/Staffware plc/Staffware Server/Nodes/***nodename***/DEBUG**

where *nodename* is the iProcess Engine node name.

Refer to the individual system message descriptions for details of the debug string that is required when investigating a particular error.

To disable debugging for the **staffifc.exe** process, delete or blank the value of the DEBUG registry key.

Debugging the Pstaffifc (UNIX) Process

To enable debugging for the **Pstaffifc** process, put the required debug string in an environment variable called DEBUG, then export DEBUG.

Refer to the individual system message descriptions for details of the debug string that is required when investigating a particular error.

To disable debugging for the **Pstaffifc** process, delete the DEBUG environment variable.

Monitoring the System Log Files

iProcess simply appends system messages to the *SWDIR***\logs\sw_warn** and *SWDIR***\logs\sw_error** files as they occur. There is no iProcess-imposed limit on the size of these files, and messages do not "roll back" to the beginning of the file to overwrite older messages.

You are therefore advised to monitor the size of these files on a regular basis, and archive or delete them if the information in them is no longer required. (iProcess automatically creates either file if it does not exist when it wants to write a system message to it.)

Chapter 2 System Message Descriptions

This chapter describes all the messages that can be returned by the iProcess Engine in the *SWDIR*`\logs\sw_warn` and `sw_error` files. .

- Messages are listed in ascending Message ID order.
- There are a number of Message IDs that are defined in the *SWDIR*`\etc\swerwarn.mes` file but are not documented in this chapter. These message IDs are not currently used by the iProcess Engine.

See Format of Message Descriptions on page 4 for a description of the information supplied about each message.

If the information in this chapter does not help you to solve the problem, or if you are at any point or for any reason unsure of how to proceed to resolve a problem, please contact TIBCO Support for further assistance.

1601 FATAL ERROR <start_case: Reached maximum case number> <*ProcName*>

or

FATAL ERROR <*FuncName*> <Maximum REQID reached>

Description: An iProcess system limit has been reached. Either:

- The maximum number of cases has been reached, when trying to start a new case of procedure *ProcName*.

  or

- The maximum number of request ids (REQID) has been reached, by function *FuncName*, when trying to either start a case or send out a new work item.

Appears In: *SWDIR***logs****sw_error**

Severity: **1 (Critical)**

Action: Investigate the error immediately. Contact TIBCO Support for further assistance.

See Also: "Sequence Number Caching" in *TIBCO iProcess Engine Administrator's Guide* for more information about REQIDs.

1609 WARNING: Only *Num* kilobytes free in *VolumeNames*'s volume, at least *MinFreeKB* are needed

Description: The Process Sentinels have detected that the *VolumeName* file system has only got *Num* KB of free space left, which is less than the *MinFreeKB* minimum required for an iProcess background (**BG**) process to run.

The Process Sentinels regularly check that the system has sufficient disk space for iProcess to operate. The *SWDIR***\etc\fs_part** file defines the list of file systems that are used by iProcess and checked by the Process Sentinels.

Appears In: *SWDIR***\logs\sw_error**

Severity: 2 (Severe)

Action: Ensure that there is sufficient disc space in all the file systems that iProcess is using.

See Also: "Administering Process Attributes" in *TIBCO iProcess Engine Administrator's Guide* for more information about the following process attributes:

- **CHECKFREQ** defines the number of processing loops that the Process Sentinels cycle through before checking that the system has sufficient available disk space. The default value is 50.
- **MINFREEKB** defines the minimum free disk space required (in KB) for a **BG** process to run. The default value is 10000.
- **PROCESS_SLEEP** defines the amount of time (in seconds) the Process Sentinels will sleep for. The default value is 5.

1613 ERROR: STAFFIFC '*ErrorMessage*'

Description: A bespoke application has encountered an error and generated this message by using the command:

```
SWDIR\util\staffifc SWLOG SW_ERROR "ErrorMessage"
```

Appears In: *SWDIR*\logs\sw_error

Severity: As determined by the application developer.

Action: Investigate the error. Contact the application developers for further assistance.

1631 WARNING: <*Msg1*> <*Msg2*> <*Msg3*> <*Msg4*>

Description: This message is returned in a wide variety of circumstances. Up to four individual pieces of information (explanatory text or numeric codes) may be returned as part of the message.

Appears In: *SWDIR***logs\sw_warn**

Severity: Varies according to the particular circumstances that caused the warning.

Action: Investigate the problem according to the information returned with the warning. Contact TIBCO Support if you require further assistance.

1636 Insufficient Memory. Function *FuncName*() *Num* bytes

Description: The process specified in the message header tried to allocate *Num* bytes of memory when calling function *FuncName*, but was unable to do so.

Appears In: *SWDIR*\logs\\sw_error

Severity: Either:

- **1 (Critical)**, if the process is one that runs continuously, such as **SPO**, **WQS** or **WIS**,

  or

- 2 (Severe), if the affected process is one that can be stopped and restarted, such as **BG** or **swbatch**.

Action: Investigate whether the process size is growing or whether the amount of free memory on the iProcess Engine machine is getting low.

Processes can be expected to grow quite a bit initially, as they cache up various pieces of information that they require. However, if a process is growing unexpectedly, stop and restart it.

Background (**BG**) processes can be stopped and restarted without affecting foreground processes. Restarting SPO, WQS or WIS processes will affect client users.

If a process still continues to grow there may be a memory leak. Gather evidence of the size of the memory growth, then contact TIBCO Support for further assistance.

See Also: "Administering iProcess Engine Server Processes" in *TIBCO iProcess Engine Administrator's Guide* for more information about how to stop and start iProcess processes.

1642 Possible infinite loop in Procedure:<*ProcName* > Case:*CaseNum*... Sending Step <*StepName* > exceeded max send/withdraw actions per workflow transaction (process attribute BG_MAX_ACTIONS_PER_TRANS = *Value*).

Description: iProcess has detected a possible infinite loop in case *CaseNum* of procedure *ProcName*. Sending step *StepName* of this case has exceeded the maximum number of actions permitted per transaction, as defined by the BG_MAX_ACTIONS_PER_TRANS process attribute.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Check the procedure to determine whether it contains an infinite loop - either by design, or because an exit condition is not being met. If it does, redesign the procedure to remove it.

If the procedure contains a number of consecutive EAI steps and/or complex routers that is greater than the value of the BG_MAX_ACTIONS_PER_TRANS process attribute, increase the attribute value as required.

See Also: "Administering Process Attributes" in *TIBCO iProcess Engine Administrator's Guide* for more information about the BG_MAX_ACTIONS_PER_TRANS process attribute.

1644 **Invalid field Value: Procedure <*ProcName* >, Case <*CaseNum* >, Step <*StepName* > : <*FieldName-FieldValue* >**

Description:	A Background (**BG**) process received an invalid *FieldName-FieldValue* assignment - for example, a number, date or time may not have been in the correct format.
Appears In:	*SWDIR*\logs\\sw_warn
Severity:	2 (Severe)
Action:	Refer the problem to the application developers responsible for procedure *ProcName*. They will need to investigate how the invalid *FieldName-FieldValue* is being passed to iProcess. For the affected *CaseNum* case, you can issue an event to set the correct field value (for example, by using the *SWDIR*\bin\\swutil or *SWDIR*\util\\swbatch utilities, or TIBCO iProcess™ Objects\TIBCO iProcess™ Server Objects). However, the affected field may have been used in other calculated fields or in conditions in the procedure. The application developers will need to investigate further to determine the scope of the problem and what needs to be done to rectify it.

1645 Invalid field Name: Procedure <*ProcName* >, Case <*CaseNum* >, Step <*StepName* > : <*ProcName-FieldName* >

Description: An attempt has been made to assign a field in a case that does not recognize the specified *FieldName* as a valid field.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer the problem to the application developers responsible for procedure *ProcName* so that they can investigate where the erroneous *FieldName* is coming from. For example, the *FieldName* may have been deleted from the procedure.

For the affected *CaseNum* case, if:

- the indicated *FieldName* should not be assigned, no further action is required.
- the wrong field has been assigned, you can issue an event to set the correct field (for example, by using the *SWDIR***bin\swutil** or *SWDIR***util\swbatch** utilities, or TIBCO iProcess Objects\TIBCO iProcess Server Objects). However, if the affected *FieldName* has been used in other calculated fields or in conditions in the procedure, the application developers will need to investigate further to determine the scope of the problem and what needs to be done to rectify it.

1646 Cannot find field in Procedure <*ProcName* >, Step <*StepName* >: Field <*FieldName* >

Description: A Background process (**BG**) needs to find a value for field *FieldName*, but it cannot find a field definition for the field in the *ProcName* procedure.

This can happen if, for example, a field is used in an expression (such as a conditional action) in a procedure, but the procedure definer later deletes the field from the procedure definition.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer the problem to the application developers responsible for procedure *ProcName*. They will need to investigate where the invalid *FieldName* is being referenced and rectify the procedure.

If an invalid field is used in a calculation or a condition the correct working of a case can be seriously impacted:

- If the field is used in a calculation, you can issue an event to set the correct field value (for example, by using the *SWDIR***bin\swutil** or *SWDIR***util\swbatch** utilities, or TIBCO iProcess Objects\TIBCO iProcess Server Objects).
- If the field is used in conditions between steps, then the wrong steps may have been sent out. The application developers will need to investigate further to determine the scope of the problem and what needs to be done to rectify it.

1647 Invalid step in actions, Procedure <*ProcName* >, Step <*StepName* >: Name <*WaitedForStep*>

Description: The indicated *StepName* has been defined to wait on the *WaitedForStep*, but the *WaitedForStep* does not exist in the *ProcName* procedure.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer the problem to the application developers responsible for procedure *ProcName*. They will need to investigate why the specified *StepName* is waiting for a *WaitedForStep* that no longer exists in the procedure. For example, the step may have been deleted from the procedure.

1649 Cannot start case for procedure <*ProcName* >, since the DEMO licence limit of <20> cases has been reached

Description: iProcess cannot start a case of procedure *ProcName*. This could be because the iProcess Engine still has a demonstration license rather than a full license. When the iProcess Engine is initially installed, it is set up with a demonstration license that allows you to create a maximum of 4 steps per procedure, and to start a maximum of 20 cases per procedure.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Obtain a full iProcess license for the iProcess Engine.

See Also: The appropriate *TIBCO iProcess Engine Installation Guide*, for more information about how to obtain and implement a full license.

1667 Asynchronous JUMPTO attempted by '*UserName*', treating it as a Synchronous request

Description: The iProcess Engine has received an asynchronous "Jump To" command from an TIBCO iProcess Objects, TIBCO iProcess Server Objects or SAL application. Asynchronous “Jump To” commands are not supported; the asynchronous parameter is ignored. The command is still processed, but it is processed synchronously.

Appears In: *SWDIR***logs\sw_warn**

Severity: 4 (Information)

Action: Modify your application to change the “Jump To” command parameter from asynchronous to synchronous.

1670 Illegal admin access attempt, Proc *ProcName* (host *NodeName*) user *UserName*

Description: User *UserName* has attempted to administer a case of the *ProcName* procedure (using the **Case Administration** tool in Work Queue manager), but is not allowed to do so. This is because **Case Administration** access to this procedure has been restricted (using the **Procedure** > **Access** > **Case Admin** option in the TIBCO iProcess Modeler).

Appears In: *SWDIR***logs\sw_warn**

Severity: 4 (Information)

Action: Either grant Case Administration access to this user, or log in as another user who does have Case Administration access to this procedure.

See Also: “Controlling Access to Procedures” in the *TIBCO iProcess Modeler - Procedure Management Guide*.

1671 Illegal start attempt, Proc *ProcName* (host *NodeName*) user *Username*

Description: User *UserName* has attempted to start a case of the *ProcName* procedure, but is not allowed to do so. This is because **Case Start** access to this procedure has been restricted (using the **Procedure** > **Access** > **Case Start** option in the TIBCO iProcess Modeler).

Appears In: *SWDIR***logs\sw_warn**

Severity: 4 (Information)

Action: Either grant Case Start access to this user, or log in as another user who does have Case Start access to this procedure.

See Also: "Controlling Access to Procedures" in the *TIBCO iProcess Modeler - Procedure Management Guide*.

1672 Can't access definition files for procedure *ProcName* (host *NodeName*)

Description: A *SWDIR***bin\swutil** command tried to access the procedure definition database records for the *ProcName* procedure, but was unable to do so.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,p=255,q=57,P=255,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR***bin\swutil** command that failed.
3. Send the resulting *SWDIR***logs\pstaffifc01.log** log file to TIBCO Support for further investigation.
4. Disable debugging on the **staffifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

1673 Can't access file 'staffusr'

Description: A *SWDIR***bin****swutil** command tried to access the iProcess user tables, but was unable to do so.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,U=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR***bin****swutil** command that failed.
3. Send the resulting *SWDIR***logs****pstaffifc01.log** log file to TIBCO Support for further investigation.
4. Disable debugging on the **staffifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

1674 User '*UserName*' is not a valid user

Description: The specified operating system *UserName* is not an iProcess user, but attempted to run a *SWDIR***bin****swutil** command that can only be run by iProcess users - for example, *SWDIR***bin****swutil** AUDIT.

Appears In: *SWDIR***logs****sw_warn**

Severity: 4 (Information)

Action: Either:

- Re-run the *SWDIR***bin****swutil** command that failed as an appropriate iProcess user.

  or

- Make the *UserName* user an iProcess user, and then re-run the *SWDIR***bin****swutil** command that failed.

1676 Procedure '*ProcName*' not found (host node *NodeName*)

Description: A *SWDIR***bin****swutil** command tried to access a procedure that does not exist on the system.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: If the specified *ProcName* is incorrect, re-run the command using the correct *ProcName*.

If the specified *ProcName* is correct, import the procedure to the system, then re-run the command.

1682 Work item '*ReqID:HostName*' in queue '*QueueName*' unlocked.

Description: The indicated work item has been unlocked using the *SWDIR***bin\swutil UNLOCKMAIL** command.

Appears In: *SWDIR***logs\sw_warn**

Severity: 4 (Information)

Action: None.

See Also: "Clearing Long-Locked Work Queue Items" in the *TIBCO iProcess swutil and swbatch Reference Guide*.

1694 No addressee for step *StepName*, case *CaseNum*, procedure *ProcName*

Description: The specified *StepName* cannot be delivered because it has no addressee. The work item is instead sent to user **swadmin**'s **Undelivered** work queue.

This error can occur if the *StepName* uses a field addressee, but the field has not been initialized when the step is sent out.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. To stop the error happening again, correct the procedure definition for the specified procedure and step so that it does have a valid addressee.
2. To repair the affected *CaseNum*, log in to the TIBCO iProcess Workspace as user **swadmin**, and forward the *StepName* work item from the **Undelivered** work queue to the intended recipient.

See Also: "Forwarding Work Items from a Queue" in the *TIBCO iProcess Workspace (Windows) User's Guide*.

1712 Resend Outstanding Mail Failed - master procedure list (staffp) access error

Description: A Background (**BG**) process cannot access the list of procedures after receiving a *SWDIR***bin\swutil RESEND** command.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debug settings on the **BG** process:

```
SWDIR\util\swadm set_attribute 0 bg 0 debug
c=255,m=41,P=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR***bin\swutil RESEND** command.
3. Send the resulting log files (*SWDIR***logs\bg*.log**) to TIBCO Support for further investigation.

See Also: The following:

- the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the **swutil RESEND** command.
- "Set a Process Attribute", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR***util\swadm** command.

1713 Resend Outstanding Mail Failed - cannot find procedure in master procedure list

Description: A Background (**BG**) process cannot find a procedure that it needs after receiving a *SWDIR***bin\swutil RESEND** command.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debug settings on the **BG** process:

```
SWDIR\util\swadm set_attribute 0 bg 0 debug
c=255,m=41,P=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR***bin\swutil RESEND** command.
3. Send the resulting log files (*SWDIR***logs\bg*.log**) to TIBCO Support for further investigation.

See Also: The following:

- the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the **swutil RESEND** command.
- “Set a Process Attribute”, in the *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR***util\swadm** command.

1714 Resend Outstanding Mail for '*ProcName*' Failed - Case data access error

Description: A Background (**BG**) process cannot access the case information for a case of the *ProcName* procedure after receiving a *SWDIR***bin\swutil RESEND** command.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debug settings on the **BG** process:

```
SWDIR\util\swadm set_attribute 0 bg 0 debug
c=255,m=41,C=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR***bin\swutil RESEND** command.
3. Send the resulting log files (*SWDIR***logs\bg*.log**) to TIBCO Support for further investigation.

See Also: The following:

- the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the **swutil RESEND** command.
- "Set a Process Attribute", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR***util\swadm** command.

1715 Resend Outstanding Mail for '*ProcName*' Failed - Procedure Definition access error

Description: A Background (**BG**) process cannot access the procedure definition for the *ProcName* procedure that it needs after receiving a *SWDIR*\bin\\swutil RESEND command.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debug settings on the **BG** process:

```
SWDIR\util\swadm set_attribute 0 bg 0 debug
c=255,m=41,p=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the *SWDIR*\bin\\swutil RESEND command.
3. Send the resulting log files (*SWDIR*\**logs\\bg*.log**) to TIBCO Support for further investigation.

See Also: The following:

- the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the **swutil RESEND** command.
- "Set a Process Attribute", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR*\util\\swadm command.

1723 WARNING: STAFFIFC '*ErrorMessage*'

Description: A bespoke application has encountered an error and generated this message by using the command:

```
SWDIR\util\staffifc SWLOG SW_WARN "ErrorMessage"
```

Appears In: *SWDIR***logs\sw_warn**

Severity: As determined by the application developer.

Action: Contact the application developers for further assistance in dealing with this error.

This error message is generated by the bespoke application - it is not a system error message. This error code simply provides a vehicle for a bespoke application to generate a message in the SWDIR**logs\sw_error** or SWDIR**logs\sw_warn** file.

1792 SetStepStatus - not processed (invalid step list argument)

Description: A call to the SETSTEPSTATUS function has not been processed because the function's *StepNameList* parameter has an invalid value or is SW_NA. The status of the target step has not been changed.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Correct the SETSTEPSTATUS function.

See Also: The following:

- the *TIBCO iProcess Expressions and Functions Reference Guide*, for details of the SETSTEPSTATUS function syntax.
- the *TIBCO iProcess Modeler - Basic Design Guide*, for information about how to use the SETSTEPSTATUS function to control Waits in loops or conditions.

1793 SetStepStatus - not processed (invalid step status argument)

Description: A call to the SETSTEPSTATUS function has not been processed because the function's *NewStatus* parameter has an invalid value or is SW_NA.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 2 (Severe)

Action: Correct the SETSTEPSTATUS function.

See Also: The following:

- the *TIBCO iProcess Expressions and Functions Reference Guide*, for details of the SETSTEPSTATUS function syntax.
- the *TIBCO iProcess Modeler - Basic Design* guide, for information about how to use the SETSTEPSTATUS function to control Waits in loops or conditions.

1794 SetStepStatus - step *StepName* does not exist in procedure

Description: A call to the SETSTEPSTATUS function has not been processed because the *StepName* supplied in the function's *StepNameList* parameter does not exist.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Correct the SETSTEPSTATUS function.

See Also: The following:

- the *TIBCO iProcess Expressions and Functions Reference Guide*, for details of the SETSTEPSTATUS function syntax.
- the *TIBCO iProcess Modeler - Basic Design* guide, for information about how to use the SETSTEPSTATUS function to control Waits in loops or conditions.

1795 **SetStepStatus - step *StepName* is outstanding, cannot be set to *StepStatus***

Description: A call to the SETSTEPSTATUS function has not been processed because the *StepName* supplied in the function's *StepNameList* parameter is **Outstanding**, so it could not be set to the indicated *StepStatus* (**Not Processed** or **Released**).

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: Correct the SETSTEPSTATUS function.

See Also: The following:

- the *TIBCO iProcess Expressions and Functions Reference Guide*, for details of the SETSTEPSTATUS function syntax.
- the *TIBCO iProcess Modeler - Basic Design* guide, for information about how to use the SETSTEPSTATUS function to control Waits in loops or conditions.

1796 SetStepStatus - Failed to set the status of step *StepName* to *StepStatus*

Description: A call to the SETSTEPSTATUS function has not been processed. The *StepName* supplied in the function's *StepNameList* parameter could not be set to the indicated *StepStatus* (**Not Processed** or **Released**), because of an internal error.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debug settings on the **BG** process:

```
SWDIR\util\swadm set_attribute 0 bg 0 debug
c=255,m=41,C=255,q=57,TIME=ABS,FLAGS=U
```

2. Re-run the **SETSTEPSTATUS** command.
3. Send the resulting log files (*SWDIR***logs\bg*.log**) to TIBCO Support for further investigation.

See Also: The following:

- the *TIBCO iProcess Expressions and Functions Reference Guide*, for details of the SETSTEPSTATUS function syntax.
- the *TIBCO iProcess Modeler - Basic Design* guide, for information about how to use the SETSTEPSTATUS function to control Waits in loops or conditions.

1810 **Sub-case not started due to reaching nesting limit of *NestingLimit* in *ProcName*-*CaseNum*-*StepName*.**

Description: A Background process (**BG**) failed to start a case of a sub-procedure because the system's maximum sub-procedure nesting depth has been reached.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve this problem, do one of the following:

- Check the specified *ProcName* and *StepName* to see if there is a problem in the procedure logic. For example, a sub-procedure that calls itself, a group of procedures that call each other in a loop, or a condition in which the exit criteria are not being met, thus creating an infinite loop.

  or

- If this is not the case, use the *SWDIR***util\swadm** command to increase the value of the MAX_SUB_PROCEDURE_DEPTH process attribute, which defines the system's maximum sub-procedure nesting depth.

See Also: "Administering Process Attributes", in *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR***util\swadm** command.

1811 **RELEASED procedure (*ParentProcName*-*CallingStepName*-*CaseNum*) started case of UNRELEASED procedure *SubProcName*.**

Description: The specified *CaseNum* of released procedure *ParentProcName* started a sub-case of unreleased procedure *SubProcName,* because no released version of *SubProcName* was available.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 2 (Severe)

Action: Ensure that released versions are available of all sub-procedures that can be called by live cases of released procedures.

1814 **Sub-case completion field mapping error** *SubProcName:SubProcFieldName* **to** *ParentProcName:ParentProcFieldName* **in step** *ParentProcCallingStep***.**

Description:	The specified sub-procedure field is mapped to a parent procedure field of the wrong type, which means that the mapping cannot be done.
Appears In:	*SWDIR***\logs\sw_warn**
Severity:	2 (Severe)
Action:	Check the field types and lengths in the two procedures for the specified fields.
See Also:	The *TIBCO iProcess Modeler - Advanced Design* guide, for more information about how to define and use sub-procedures.

1815 Sub-case started of a procedure that isn't a sub-procedure: *ParentProcName* called *SubProcCallStepName* (precedence *Precedence*).

Description:	The specified sub-procedure call step is trying to call a procedure/version that is not a sub-procedure.
Appears In:	*SWDIR*\logs\\sw_warn
Severity:	2 (Severe)
Action:	Check the specified *SubProcCallStepName* and make sure that it is calling a sub-procedure and not a main procedure.
See Also:	The *TIBCO iProcess Modeler - Advanced Design* guide, for more information about how to define and use sub-procedures.

1840 Directory <*DirName*> does not exist. Not adding to the partition list cache.

Description: The *DirName* directory does not exist, so iProcess cannot monitor the space available in that directory.

The *SWDIR***etc****fs_part** file defines the directories that iProcess monitors for available space. *DirName* is listed in that file.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: Check the names of the directories in the *SWDIR***etc****fs_part** file to make sure that they are all correct.

1850 Invalid work queue address '*QueueName*' for new work item (procedure:*ProcName*@*NodeName* case :*CaseNum*) - user/group does not exist.

Description: A new work item message instruction is undeliverable because the *QueueName* does not resolve to a valid user or group on the system. The work item is forwarded to the undelivered work items queue on the procedure's host node.

The **WISMBD** process may also terminate when this error occurs.

Appears In: *SWDIR***logs****sw_warn**

Severity: Either:

- **1 (Critical)**, if the **WISMBD** process also terminates,

  or

- 2 (Severe), if the **WISMBD** process is still running.

Action: To resolve the problem:

1. Restart the **WISMBD** process (if necessary).
2. To repair the affected *CaseNum*, log in to the SPC as user **swadmin**, and forward the work item from the **Undelivered** work queue to the intended recipient.
3. Investigate why the procedure was trying to send this work item to a queue that does not exist.

See Also: "Forwarding Work Items from a Queue" in the *TIBCO iProcess Workspace (Windows) User's Guide*.

1851 Invalid work queue address '*QueueName*' for MBOX message 'PURGE^*Instruction*'.

Description: The WISMBD process has received a PURGE instruction for the *QueueName* work queue, but that *QueueName* does not exist on the system (according to the WQS process). The PURGE instruction will be ignored.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Investigate why the iProcess Engine was trying to send the PURGE instruction to a work queue that does not exist. If the work queue:
   - has existed for some time, check that it does still exist (using the *SWDIR***util\plist -u** command).
   - has recently been added or deleted, the problem may just be a timing issue.
2. If necessary, use the *SWDIR***bin\swutil** command to resend the PURGE instruction.

See Also: "Purging Cases", in the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about how to purge a case using *SWDIR***bin\swutil**.

1852 Error forwarding MBOX message to WIS (error=*ErrCode*).

Description: The WIS process specified in the message header is not responding. This is because the WISMBD process failed to forward a message to the WIS.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Note the *ErrCode* and contact TIBCO Support for further assistance.

1853 Invalid or missing MBOX Set process attribute: '*AttributeName*'

Description: Either:

- the specified “MBSET_” process attribute is either not set, or is set to 0, for the process specified in the message header, or
- the MBox set specified by the “MBSET_” process attribute has been deleted.

Appears In: *SWDIR***logs\sw_warn**

Severity: **1 (Critical)**

Action: Define the specified *AttributeName* as a valid MBOX Set ID for its access type.

See Also: “Administering Process Attributes”, in the *TIBCO iProcess Engine Administrator’s Guide*, for more information about the “MBSET_” process attributes.

1854 **Failed to load MBOX Set:** *MboxSetID*

Description: The specified *MboxSetID* could not be loaded, or there are no Mboxes in this Mbox set.

Appears In: *SWDIR***logs\sw_warn**

Severity: **1 (Critical)**

Action: Check the Mbox set configuration for the specified *MboxSetID*.

See Also: "Administering Message Queues and Mbox Sets", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about Mboxes and Mbox sets.

1855 Could not initialise access to MBOX *MboxID* in MBOX Set *MboxSetID*

Description: iProcess cannot initialize access to the specified *MboxID* in the specified *MboxSetID*.

Appears In: *SWDIR*\logs\\sw_warn

Severity: **1 (Critical)**

Action: To resolve the problem:

1. Check that an entry exists for the specified *MboxID* in the **iql_queue** table .
2. Check that the physical queue (that the specified *MboxID* is mapped to) is correctly set up.

See Also: The following:

- "Administering Message Queues and Mbox Sets", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about Mboxes and Mbox sets.
- "Mbox Sets and Message Queues", in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide* for your database, for a definition of the **proc_defn** table.

1856 **Error sending msg to queue** *QueueName*, **msginfo=***Info*, **msgtype=***Type*

Description: An error has occurred while sending a message to the specified *QueueName*. The IQL driver has returned a send failure message. The affected transaction is being rolled back.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Monitor the transaction. The transaction will be automatically retried a number of times. (On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value.On Oracle versions the retry parameters are attributes of the Oracle AQ queues.) If:

- the message is successfully delivered on a subsequent retry, no further action is required (unless other error messages or symptoms have occurred).
- the maximum retry count is exceeded, the message will be put on the exception queue. Contact TIBCO Support for further assistance in resolving the error.

See Also: “IQL_RETRY_COUNT”, in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1857 Error receiving msg from queue *QueueName*, msgtype=*Type*

Description: An error has occurred while receiving a message from the specified *QueueName*. The IQL driver has returned a receive failure message. The affected transaction is being rolled back.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Monitor the transaction. The transaction will be automatically retried a number of times. (On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value.On Oracle versions the retry parameters are attributes of the Oracle AQ queues.) If:

- the message is successfully received on a subsequent retry, no further action is required (unless other error messages or symptoms have occurred).
- the maximum retry count is exceeded, the message will be put on the exception queue. Contact TIBCO Support for further assistance in resolving the error.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1858 Error initialising physical queue *QueueDesc* for QID = *QueueID*

Description: iProcess cannot initialize the specified physical *QueueID*, because the physical queue description *QueueDesc* has been specified using an incorrect format.

The ID of the physical queue that is used to hold a message queue is specified in the **queue_phys_descr** column of the **iql_queue** table.

Messages cannot be added to or read from the message queue that is mapped to this physical queue, which means that the iProcess Engine cannot function correctly.

Appears In: *SWDIR*\logs\\sw_warn

Severity: **1 (Critical)**

Action: Use the *SWDIR*\util\\swadm update_queue command to correct the physical queue description *QueueDesc* for the queue *QueueID*.

See Also: The following:

- "Update a Message Queue", in the *TIBCO iProcess Engine Administrator's Guide*, for more information about the *SWDIR*\util\\swadm update_queue command.
- "iql_queue", in the appropriate *TIBCO iProcess Engine (Database) Administrator's Guide* for your database, for more information about the required format of the physical queue description.

1861 Oracle Context information not present in Environment Context

Description: A program has attempted to read or write a message to a queue but has found that the handles to access the database are not available. This can happen if:

- a database transaction has not been started, or
- a connection to the database has been invalidated during the processing of a transaction.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Set debugging for the process specified in the message header. For example, use the following command to turn on debugging for the SPO process:

```
swadm set_attribute 1 spo 0 debug TID=1,q=63,e=255,h=255,
i=255,m=41,TIME=ABS,FLAGS=U
```

⚠ Turning on debugging can have a significant impact on system performance. Please see Debugging on page 6 before using it.

2. Reproduce the problem.
3. Send the log files to TIBCO Support for further assistance.
4. Turn debugging off as soon as you no longer need it, using the following command:

```
swadm delete_attribute 0 ProcessName ProcessInstance debug
```

1864 Transaction failure : Reason code = *Code*

Description:	The process specified in the message header has attempted to process an instruction but the attempt has failed with error *Code*. The affected transaction is being rolled back.
Appears In:	*SWDIR***logs\sw_warn**
Severity:	2 (Severe)
Action:	Examine the *SWDIR***logs\sw_warn** file for a message produced by the **BG** process prior to this message. That message provides additional information about this error and how to resolve it.

1865 MBOX Set Dequeue error: *ErrText*

Description: A **BG** process has encountered an error while attempting to dequeue a message from an Mbox set. The following table lists the possible *ErrText* values.

ErrText	Description
INVALID MESSAGE	The message was dequeued successfully, but an error occurred while getting message data or header information.
DEQUEUE FAILED	An error occurred during the dequeue from the Mbox.
MBOX SET INVALID	Either: • The process attribute for the given access type is either not set for the process, or is set to < 1, or, • An error occurred accessing the Mbox Set definition (via the process attribute value appropriate to the access type in the database).
MBOX SET INIT ERROR	Failed to initialize access to every Mbox in the Mbox set.
MBSET SYSTEM ERROR	Other errors. For example, wrong type of access or out of memory.

Appears In: *SWDIR***logs****sw_warn**

Severity: Either:

- 1 (Critical), if MBox sets have been incorrectly configured. (For example, if the **WIS** processes do not have an MBSET_WRITE_BG attribute set no work items can be released.)

  or

- 2 (Severe), if the system is still usable. For example, if an error has occurred accessing one of the message queues in an Mbox set.

Action: Check the following:

1. System mappings between Mbox sets, message queues, and physical database tables are correctly configured.
2. Message queues are mapped to physical tables that exist and conform to the correct format.

See Also: “Administering Message Queues and Mbox Sets”, in the *TIBCO iProcess Engine Administrator’s Guide*, for more information about how to administer Mbox sets, message queues and the physical database tables that support them.

1866 Failed to load '*EAIType*' EAI Run-Time Plug-In library '*Path*', error '*ErrMsg*'

Description: The iProcess Engine failed to load the specified *EAIType* EAI Server Plug-in from the specified *Path*. The *ErrMsg* text provides more details about the reason for failure.

The BG processes may stop (for example if the problem is with the TIBCO iProcess Java Plug-in), or may continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the specified EAI Server Plug-in exists at the specified *Path and* that the necessary access permissions exist on the library.
2. Use the *SWDIR***util\sweaireg LIST** command to list the existing EAI Plug-in entries. Search this list for *EAIType* to ensure that the plug-in is correctly registered. Note that Java-related EAI types (such as the TIBCO iProcess BusinessWorks Plug-in) will not be listed separately.
3. If this error occurs after the uninstallation of a component, follow the instructions in the section "Unregister (Remove) an EAI Plug-in" in TIBCO iProcess Engine Administrator's Guide to unregister the *EAIType* component.
4. Check that the operating-system specific shared library path requirements are satisfied. For example, on Solaris, ensure that the LD_LIBRARY_PATH environment variable is set correctly when the process sentinels are started.
5. If you have made any changes then try restarting the background processes.
6. If this does not resolve the problem, please use the following command to set debugging on:

```
swadm set_attribute 0 BG 0 debug ALL=1,x=255
```

and restart the background process. Contact TIBCO Support with the resulting log files.

See Also: The following:

- "IQL_RETRY_COUNT", in *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.
- "Managing EAI Step Server Plug-ins", in the *TIBCO iProcess Engine Administrator's Guide*, for information about how to use the *SWDIR*\util\\sweaireg utility.
- The appropriate *TIBCO iProcess Engine Installation Guide*, for more information about shared library path requirements.

1867 '*EAIType*' EAI Run-Time Plug-In failed to initialise, error *ErrMsg*

Description: The specified EAI Server Plug-in failed to initialize correctly. The *ErrMsg* text provides more details about the reason for failure.

The BG processes will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

> The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer to the documentation for the specified EAI Server Plug-in for more information about the *ErrMsg* text, the possible causes of the error and how to resolve it.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1868 The interface function '*FuncName*' (required for interface version *VerNum*) is not available in EAI Run-Time Plug-In '*EAIType*'.

Description: The *EAIType* EAI Server Plug-in claims to conform to the specified version of the EAI Server Plug-in interface, but does not meet the requirements of that interface

The BG processes will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: Contact TIBCO Support (or the support organization for the specified EAI plug-in if it is not supplied by TIBCO) for further assistance.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1869 **Procedure:***ProcName@HostName* **EAI Step:***StepName* **Case:***CaseNum* **EAI plug-in EAIRun_Call() failed, error:** *ErrMsg*

Description: The EAI plug-in has encountered an infrastructure-level error while initiating the call-out for the specified EAI step. For example, an EAI plug-in has failed to connect with some external resource. The *ErrMsg* text provides more details about the reason for failure.

The BG processes will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe) or 3 (Warning), depending on the specific error and the EAI Server Plug-in that returned it.

Action: Refer to the documentation for the specified EAI Server Plug-in for more information about the *ErrMsg* text, its severity, the possible causes of the error and how to resolve it.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1870 Failure in Process Startup routines: *Routine* : *RetCode*

Description: An error has occurred while initializing basic process resources. *Routine* is either **Process Startup** or **Process status**, and *RetCode* is the return code from the specified *Routine*.

Appears In: *SWDIR***logs\sw_error** or *SWDIR***logs\sw_warn**

Severity: Either:

- **1 (Critical)**, if a *required* system process fails to start up. (For example, if the BGPREDICT process fails to start, it is only critical if you actually use prediction in your procedures.)

  or

- 2 (Severe) otherwise. For example, if a utility process such as **swbatch** does not start, the problem will not prevent the system being used, but should still be investigated.

Action: Contact TIBCO Support for further assistance if the error is not self-explanatory.

1871 The '*EAIType*' EAI Run-Time Plug-In does not support any interface specification version that the background server supports.

Description: The **BG** process does not support any of the EAI run-time plug-in interface versions that are provided by the EAI Server Plug-in. The BG process and EAI Server Plug-in cannot communicate with each other.

The **BG** process will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

> The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: Install versions of the iProcess Engine and/or the EAI Server Plug-in that support compatible interface versions.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine Administrator's Guide*, for a detailed description of this process attribute.

1872 '*EAIType*' EAI Run-Time Plug-In's '*FunctionName*' function provided an invalid value for '*ExpectedData*'.

Description: The *EAIType* EAI Server Plug-in has returned an unexpected value for the specified item of data. (*ExpectedData* is a description of the expected data for which the value in question is invalid.)

If this happens:

- while the plug-in is being initialized, the **BG** process will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.
- during the run-time processing of an EAI step, the **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

Transactions will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe) or 3 (Warning), depending on the specific error and the EAI Server Plug-in that returned it.

Action: Refer to the documentation for the specified EAI Server Plug-in for more information about the *ErrMsg* text, its severity, the possible causes of the error and how to resolve it.

See Also: “IQL_RETRY_COUNT”, in the *TIBCO iProcess Engine Administrator’s Guide*, for a detailed description of this process attribute.

1873 Procedure:*ProcName* EAI Step:*StepName* Case:*CaseNum* failed. Could not run the EAI call-out - the run-time plug-in for this EAI step type (*EAIType*) is either not registered or failed to load and initialise correctly.

Description: The **BG** process has attempted to process an EAI step of the given *EAIType*, but the *EAIType* EAI Server Plug-in is either not installed or has failed to initialize correctly.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

> The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the *EAIType* EAI Server Plug-in has been installed correctly.
2. Check both the *SWDIR***logs\sw_warn** and **sw_error** files for any initialization errors for the *EAIType* EAI Server Plug-in.
3. Contact TIBCO Support (or the support organization for the specified EAI plug-in if it is not supplied by TIBCO) for further assistance.

See Also: “IQL_RETRY_COUNT”, in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1874 Procedure:*ProcName* EAI Step:*StepName* Case:*CaseNum* failed. Could not withdraw the delayed release EAI step - the run-time plug-in for this EAI step type (*EAIType*) is either not registered or failed to load and initialise correctly.

Description: The **BG** process has attempted to process a withdraw of an EAI step of the given *EAIType*, but the *EAIType* EAI Server Plug-in is either not installed or has failed to initialize correctly.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the *EAIType* EAI Server Plug-in has been installed correctly.
2. Check both the *SWDIR***logs****sw_warn** and **sw_error** files for any initialization errors for the *EAIType* EAI Server Plug-in.
3. Contact TIBCO Support (or the support organization for the specified EAI plug-in if it is not supplied by TIBCO) for further assistance.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1875

Procedure:*ProcName@HostName* Case:*CaseNum* failed. The run-time plug-in for EAI Type '*EAIType*', is using 'asynchronous with reply' call-out methods but has not provided an EAIRun_GetReply () function.

Description: The *EAIType* EAI Server Plug-in is not self-consistent in terms of its communication protocol with the **BG** process.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: Contact TIBCO Support (or the support organization for the specified EAI plug-in if it is not supplied by TIBCO) for further assistance.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1876 Procedure:*ProcName@HostName* EAI Step:*EAIStepName* Case:*CaseNum* failed. The EAI step failed to withdraw, error: *ErrMsg*.

Description: The **BG** process has attempted to withdraw the specified *EAIStepName* as part of a transaction, but has received the specified *ErrMsg* from the EAI Server Plug-in.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

Transactions will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: Refer to the documentation for the specified EAI Server Plug-in for more information about the *ErrMsg* text, the possible causes of the error and how to resolve it.

See Also: “IQL_RETRY_COUNT”, in the *TIBCO iProcess Engine: Administrator’s Guide*, for a detailed description of this process attribute.

1877 **Procedure:***ProcName@HostName* **Case:***CaseNum* **EAI plug-in type '***EAIType***' EAIRun_GetReply() failed, error:** *ErrMsg*

Description: The **BG** process has attempted to process the specified *EAIStepName* as part of a transaction, but has received the specified *ErrMsg* from the EAI Server Plug-in.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

Transactions will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: Refer to the documentation for the specified EAI Server Plug-in for more information about the *ErrMsg* text, the possible causes of the error and how to resolve it.

See Also: “IQL_RETRY_COUNT”, in the *TIBCO iProcess Engine: Administrator’s Guide*, for a detailed description of this process attribute.

1878 AQ PubSub - Function *FuncName*, Error Message *Text*, OCI Error *OCIError*.

Description: The specified Oracle *OCIError* has occurred while attempting to subscribe to, unsubscribe from or publish an event using Oracle AQ.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 2 (Severe)

Action: Refer to your Oracle documentation for more information about the *OCIError*, what might have caused it, and how to resolve it.

If *OCIError* is “**User does not exist**”, and the process named in the message header is **swsvrmgr** or **spo**, this means that the process cannot use the iProcess event management system, which is provided by Oracle AQ. Possible causes for this are:

- The Oracle Event Manager process (EMN0) has failed (usually due to the process running out of handles). iProcess should function normally again when this process is restarted. However, if the problem persists, you should shut down and restart the iProcess Engine.
- The Oracle database is hosted on a different computer from the iProcess Engine, and the Oracle database computer cannot resolve the hostname of the iProcess Engine computer. Correct the network configuration so that both computers can communicate.

1879

Procedure:*ProcName@HostName* EAI Step:*EAIStepName* Case:*CaseNum* - Invalid Delayed Release condition expression defined.

Description: The **BG** process has attempted to process the specified *EAIStepName*, but has encountered an invalid expression in the *EAIStepName*'s delayed release condition. This can happen if the expression has become invalid since it was defined - for example, if a data field that the expression references has subsequently been deleted.

The **BG** process rolls back the transaction being processed at the time of failure and continues processing other transactions.

Transactions will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Check that the delayed release condition expression is valid, using the TIBCO iProcess Modeler. (Clicking **OK** in the *EAIStepName* dialog validates the expression.)

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1880 The process cannot be started as not all of its logical parent processes are running. Machine ID = *MachineID*, Process Name = *ProcessName*, Process Instance = *ProcessInstance*.

Description: The specified process instance cannot be started because it has dependencies on parent processes that are not running.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. Make sure that *ProcName*'s hierarchy of parent processes are already running.
2. Try to start the *ProcName* process again.

See Also: The following:

- "Starting iProcess iPE Processes", in the *TIBCO iProcess Engine Architecture Guide*, for more information about the process hierarchy.
- "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for more information about how to start processes.

1881 Unable to establish communication with the peer (*PMType*) process manager. Comms error (*Errno*).

Description: The *PMType* (either **Worker** or **Watcher**) process sentinel cannot create a TCP/IP socket connection to its peer process, because of a socket Errno. This means that the *PMType* process sentinel will be unable to restart its peer process, if the latter fails.

The watcher and worker process sentinels ensure that they remain running at all times by connecting to each other over TCP/IP sockets. Each sentinel process on a machine creates a connection to the other process to ensure that its peer is running and to detect any failures.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To rectify this problem:

1. Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it:
2. In the UNIX **/etc/services** (or Windows **%systemroot%\system32\drivers\etc\services**) file, make sure that the TCP/IP port numbers assigned to the *nodename*_**watcher** and *nodename*_**worker** processes are unique (system-wide).
3. Ensure that all network adapters are correctly configured.

See Also: *TIBCO iProcess Engine Installation Guide*, for more information about allocating TCP/IP port numbers to the watcher/worker processes.

1882 Communication terminated abnormally with peer process manager (*PMType*) and could not be re-established after *Num* retries. Assuming process has died and re-starting. Socket error (*Errno*).

Description: One of the process sentinel processes has lost its TCP/IP socket connection with its peer *PMType* (either **Worker** or **Watcher**) process, due to a socket Errno. The process is attempting to restart the *PMType* peer process.

The watcher and worker process sentinels ensure that they remain running at all times by connecting to each other over TCP/IP sockets. Each sentinel process on a machine creates a connection to the other process to ensure that its peer is running and to detect any failures.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the peer process sentinel has restarted successfully. There should be two **procmgr** processes running for this iProcess Engine node.
2. Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it.

1883 The process cannot be stopped as not all of its logical child processes have been stopped. Machine ID = *MachineID*, Process Name = *ProcessName*, Process Instance = *ProcessInstance*.

Description: The specified process instance cannot be stopped because child processes that are still running have dependencies on it.

This can occur if, for example, a user tries to stop a WIS process on a node where one or more WISMBD processes, which depend on the WIS processes, are still running.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 3 (Warning)

Action: To resolve the problem:

1. Make sure that *ProcName*'s hierarchy of child processes are already stopped.
2. Try to stop the *ProcName* process again.

See Also: The following:

- "Starting iProcess iPE Processes", in the *TIBCO iProcess Engine Architecture Guide*, for more information about the process hierarchy.
- "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for more information about how to stop processes.

1884 Failed to create monitor socket. Socket error (*ErrNo*) on port number *PortNum*.

Description: The child process on port number *PortNum* has reported a socket Errno. It cannot create a server-side socket because the port number it has been given by the worker process sentinel is still in use.

The worker process sentinel uses a TCP/IP socket connection with each started iProcess Engine process to monitor that process. Each process is given a unique port number on which to create a server-side socket which the worker process sentinel can connect to and monitor. The worker process sentinel simply requests the next free port number from the operating system.

Possible causes of this error are:

- Another application creates a socket on the same port number, in between the worker process sentinel requesting the next free port number and the child process creating the server-side socket. This can happen if, for example, an application is configured to always use this port number.
- One of the worker process sentinel's child processes (for example, a **BG** process) is using a specified port number and fails. The process is automatically restarted by the worker process sentinel and attempts to create a new socket connection on the same port number as before, but for some reason the operating system has not yet marked that port as unused.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Ensure that the specified *PortNum* is not being used by another application.
2. Refer to the socket documentation for your particular operating system for more information about the socket *ErrNo*, what might have caused it, and how to resolve it.

1885 **Failed to connect to monitor socket for process (LMID =** ***MachineID*****, LPN =** ***ProcessName*****, LPI =** ***ProcessInstance*****). Socket error (*****Errno*****) on port number** ***PortNum*****.**

Description: The worker process sentinel cannot connect to the socket that should have been created by the child *ProcessName*, because of a socket Errno.

The worker process sentinel uses a TCP/IP socket connection with each started iProcess Engine process to monitor that process. Each process is given a unique port number on which to create a server-side socket which the worker process sentinel can connect to and monitor. The worker process sentinel simply requests the next free port number from the operating system.

Possible causes of this error are:

- The system is extremely heavily loaded. The child *ProcessName* cannot create its server-side socket before the worker process sentinel attempts to connect to it.
- The child *ProcessName* fails after it is created, but before the worker process sentinel attempts to connect to it. Consequently, there is no server-side socket for the worker process sentinel to connect to.
- A network infrastructure problem. For example, if multiple network cards are installed on the computer where the iProcess Engine is running, and a card is incorrectly configured. This can prevent the child *ProcessName* from being able to create its server-side socket.

Appears In: *SWDIR***logs\sw_warn**

Severity: Either:

- **1 (Critical)**, if the specified *ProcessName* is critical to the system's operation and the process sentinels have been unable to automatically restart it,

  or

- 2 (Severe), if the specified *ProcessName* is *not* critical to the system's operation but the process sentinels have been unable to automatically restart it,

  or

- 3 (Warning), if the process sentinels have automatically restarted the specified *ProcessName*.

Action: To resolve the problem:

1. Check to see if the specified *ProcessInstance* of *ProcessName* is running (and that there is only one copy of it). If it has been marked as suspended, manually restart it and check that the error does not happen again.
2. Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it.

See Also: "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for more information about how to monitor and restart processes.

1886 Failed to accept connections to socket. Socket error (*Errno*).

Description: The child process cannot accept connections on its server-side socket due to a socket Errno.

The worker process sentinel uses a TCP/IP socket connection with each started iProcess Engine process to monitor that process. Each process is given a unique port number on which to create a server-side socket which the worker process sentinel can connect to and monitor.

This error may indicate a problem with the network infrastructure.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it.

1887 Failed to send data to socket. Socket error (*Errno*).

Description: The child process cannot send its connection confirmation message to the worker process sentinel.

The worker process sentinel uses a TCP/IP socket connection with each started iProcess Engine process to monitor that process. Each process is given a unique port number on which to create a server-side socket which the worker process sentinel can connect to and monitor.

When the child process has successfully accepted a connection on its server-side socket it then attempts to send a confirmation message to the connecting process. This confirmation allows the worker process sentinel to check that it has connected to the correct process and that it can now be monitored.

This socket Errno may indicate a problem with the network infrastructure.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it.

1888 The process manager detected that process (*ProcessName*) has terminated abnormally. Socket error (*Errno*).

Description: The worker process sentinel has detected that the specified *ProcessName* has failed, with a socket Errno. (The connection to the process being monitored has been prematurely disconnected without an appropriate termination signal.)

The worker process sentinel uses a TCP/IP socket connection with each started iProcess Engine process to monitor that process. Each process is given a unique port number on which to create a server-side socket which the worker process sentinel can connect to and monitor.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the specified *ProcessName* has been automatically restarted by the worker process sentinel.

> The process will only be automatically restarted if its PROCESS_AUTO_RESTARTS process attribute is set to 1.

2. If the failed *ProcessName* is the SPO server (**SPO**), change the value of the **POOLCONNTIMEOUT** parameter, in the **DBPOOL** section of the file, from **600** to **86400**.
3. If the process has not been automatically restarted, manually restart it using the *SWDIR***util\swsvrmgr RESTART** command.
4. Monitor the process closely to ensure that it remains running.
5. Refer to the socket documentation for your particular operating system for more information about the socket *Errno*, what might have caused it, and how to resolve it.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- "Administering Process Attributes", for more information about the PROCESS_AUTO_RESTARTS process attribute.
- "Tuning the iProcess Engine Using SWDIR\etc\staffcfg Parameters", for more information about the POOLCONNTIMEOUT parameter.
- "Administering iProcess Engine Server Processes", for more information about *SWDIR***util\swsvrmgr RESTART**.

1889 **Process cannot be started as it is already running. Machine ID =** *MachineID***, Process Name =** *ProcessName***, Process Instance =** *ProcessInstance***.**

Description: A *SWDIR***\util\swsvrmgr START** command has attempted to start the specified *ProcessInstance* of *ProcessName*, but has failed because that instance is already running.

Appears In: *SWDIR***\logs\sw_warn**

Severity: 3 (Warning)

Action: None

See Also: "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for more information about how to monitor and start processes.

1891 The process manager is unable to start as no '*NodeName_PMType*' entry has been specified in /etc/services file for this installation. See installation guide.

Description: The process sentinels cannot start because the entries in the **services** file that allocate TCP/IP port numbers to the watcher and worker processes are missing.

Appears In: *SWDIR***logs\sw_warn**

Severity: **1 (Critical)**

Action: In the UNIX **/etc/services** (or Windows **%systemroot%\system32\drivers\etc\services**) file, add entries to assign unique (system-wide) TCP/IP port numbers for the worker and watcher process sentinels.

See Also: "Allocate TCP/IP Port Numbers to Services", in the *TIBCO iProcess Engine Installation Guide*, for more information about how to create the required entries.

1892 AQIQL: OCI function '*FuncName*' failed: ORA Error: *OracleError*

Description: The iProcess Engine has received an error from Oracle while sending a message to or receiving a message from an Oracle AQ queue (acting as an iProcess Mbox).

FuncName is the Oracle function that was called and OracleError is the error message returned by Oracle.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Refer to your Oracle documentation for more information about the *OracleError*, what might have caused it, and how to resolve it.

1893 '*EAIType*' EAI Run-Time called EAI call-back function '*FuncName*' with an invalid '*DataItem*' parameter value.

Description: The *EAIType* EAI server plug-in *FuncName* function has returned an unexpected value for the specified *DataItem*. The **BG** process will roll back the affected transaction and continue processing other transactions.

The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value.On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Contact TIBCO Support (or the support organization for the specified EAI plug-in if it is not supplied by TIBCO) for further assistance.

See Also: "IQL_RETRY_COUNT", in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1894 Possible infinite loop in Procedure:<*ProcName* > Case:*CaseNum*... Withdrawing Step <*StepName*> exceeded max send/withdraw actions per workflow transaction (process attribute BG_MAX_ACTIONS_PER_TRANS = *Value*).

Description: The current transaction has been aborted because the number of steps sent or withdrawn as part of the transaction has exceeded the system limit. This limit is defined by the BG_MAX_ACTIONS_PER_TRAN process attribute *Value*.

This message is produced if there is a loop in a procedure that causes too many steps to be sent.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Investigate the *ProcName* procedure to see if there is a loop in the procedure that is causing too many steps to be sent.

See Also: "BG_MAX_ACTIONS_PER_TRAN", in the *TIBCO iProcess Engine: Administrator's Guide*, for a detailed description of this process attribute.

1896 Unable to verify process type (*ProcessType*), either not a valid type or database error has occurred. Cannot start requested number of new processes.

Description: The process sentinels have either:

- been unable to obtain a database connection to start a new process because of an internal system failure, or
- received a START_NEW event (from an application other than iProcess) for an invalid *ProcessType*.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Use the following command to set debugging for the **procmgr** process (the process used by the *SWDIR***bin\swutil QINFO** command):

```
swadm set_attribute 0 procmgr 0 debug T=255,q=63,TID=1,
TIME=ABS,FLAGS=U
```

Turning on debugging can impact system performance. See Debugging on page 6 for more information.

2. Reproduce the problem.
3. Send the resulting log files to TIBCO Support for further assistance.
4. Turn debugging off as soon as you no longer need it, using the following command:

```
swadm delete_attribute 0 procmgr 0 debug
```

1898 MSIEL PubSub - Function: *FuncName*, Error Message: *ErrorType*, *Error*

Description: An error occurred while subscribing to or publishing an event (on a Windows platform iProcess Engine). If *ErrorType* is:

- **subscription**, a user tried to run the specified program when they do not have the necessary privileges in Microsoft Windows to subscribe to events that iProcess requires.
- **publish**, an error occurred publishing the event.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: Add the user named in the message header to the Local **Administrators** group. This will ensure that the user has sufficient privileges to add a transient subscription for the **iProcess Events** COM+ application.

1899 Error Parsing Multi-Subprocedure call step definition (Case: *CaseNum*, Proc: *ProcName*, Step: *StepName*).

Description: The **BG** process cannot read the definition of the *StepName* dynamic sub-procedure call step in the *ProcName* procedure.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Export the *ProcName* procedure to an **.xfr** file.
2. Send the **.xfr** file to TIBCO Support for further analysis and investigation.

See Also: "Export a Procedure", in the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about how to export a procedure definition.

1900 **Invalid sub-procedure '*CaseNum*, Proc: *ProcName*, Step: *StepName*' specified in sub-proc name array field '*SubProcName*' ().**

Description: A dynamic sub-procedure call step has attempted to start a case of the *ProcName* procedure, which does not exist on the system

Appears In: *SWDIR***logs\sw_error** or *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Fix the application so it uses correct sub-procedure names.

If the specified *ProcName* is incorrect, correct the application so that the dynamic sub-procedure call step calls the correct *ProcName*.

If the specified *ProcName* is correct, import the sub-procedure to the system.

1901

No SubProcs specified for (*CaseNum*, Proc: *ProcName*, Step: *StepName*) possibly empty array field <*FieldName*>

Description: A dynamic sub-procedure call step has been processed that did not start any sub-procedures.

Appears In: *SWDIR***logs\sw_warn**

Severity: Either:

- 2 (Severe), if sub-procedures should have been started by this *StepName*,

  or

- 3 (Warning), if it is expected application behavior that no sub-procedures were started.

Action: Check whether or not this is expected application behaviour.

1902 Worker(*PortNumber*) and watcher(*PortNumber*) port numbers must be unique in /etc/services.

Description: The process sentinels cannot start because the entries in the **services** file for the watcher and worker processes have identical *PortNumber* values.

Each entry must have a unique (system-wide) *PortNumber* value.

Appears In: *SWDIR*\logs\\sw_warn

Severity: **1 (Critical)**

Action: To resolve the problem:

1. In the UNIX **/etc/services** (or Windows **%systemroot%\\system32\\drivers\\etc\\services**) file, make sure that the TCP/IP port numbers assigned to the *nodename*_**watcher** and *nodename*_**worker** processes are unique (system-wide).
2. Restart the iProcess Engine.

See Also: The following:

- "Allocate TCP/IP Port Numbers to Services", in the *Installing the iProcess Engine* guide, for more information about how to create the required entries.
- "Controlling the iProcess Engine", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about how to restart the iProcess Engine.

1903 IEL - Function: iel_swrpc_clnt_create, Error Message: Failed to connect to the RPC event daemon after 5 attempts,

or

IEL - Function: iel_swrpc_clnt_create, Error Message: Succeeded in connect to the RPC event daemon after *N* attempts,

Description: The process specified in the message header has either been unable to connect to the RPC event daemon, or has experienced problems in doing so. This process will not be able to publish or subscribe to events.

Appears In: *SWDIR*\logs\sw_error or *SWDIR*\logs\sw_warn

Severity: 2 (Severe)

Action: Check that the process is running and that other processes can communicate with it successfully.

1904 fil_lock L_SHORT lock timed-out (user *UserName*, file *FileName*).

Description: An attempt by user *UserName* to lock the specified file timed out, where *FileName* is one of the following:

- *SWDIR***util\swldap** (which holds the current connection information, search parameters, attribute mappings and X500_DIT flag setting used by the *SWDIR***util\ldapconf.exe** utility)
- *SWDIR***tsys\staffcfg**

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: Find out if another user was using the *SWDIR***util\ldapconf.exe** utility.

1938 **fil_lock CreateMutex() failed with error :** *ErrCode* **(user** *UserName***, file** *FileName***)**

Description:	A CreateMutex() call failed on Microsoft Windows while attempting to lock a file.
Appears In:	*SWDIR***\logs\sw_warn**
Severity:	2 (Severe)
Action:	Investigate why the process could not create the mutex.

1939 *DD***/***MM***/***YYYY HH:MM* **(***ServerFil***-<***UserName***>)Caching of NT permissions for process '<***ProcessName***>' succeeded after** *Num* **seconds and** *Num* **retries**

Description: The *ProcessName* process encountered difficulties loading the file permissions definitions on Microsoft Windows, but did succeed in doing so after *Num* retries.

Appears In: *SWDIR***\logs\sw_warn**

Severity: 3 (Warning)

Action: None.

1941 pd_publish: Failed to copy VBA projects for *ProcName* **v***MajorVers.MinorVers*

Description: An error has occurred while saving a new procedure definition for *ProcName*. The error occurred while copying unchanged VBA projects across to the specified procedure version.

Appears In: *SWDIR***\logs\sw_warn**

Severity: 2 (Severe)

Action: Check that there is sufficient space in the **proc_defn** table to hold the VBA projects.

See Also: "proc_defn", in the appropriate *TIBCO iProcess Engine (Database): Administrator's Guide* for your database, for a definition of the **proc_defn** table.

1942 Failed to recreate database connection[*Num*]

Description: An attempt to recreate a database connection that had timed out has failed.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe)

Action: To resolve the problem:

1. If the problem persists, use the following command to set debugging for the affected process (as specified in the message header):

```
swadm set_attribute 0 ProcessName ProcessInstance debug
q=255,TID=1,TIME=ABS,FLAGS=U
```

⚠ Turning on debugging can have a significant impact on system performance. Please see Debugging on page 6 before using it.

2. Investigate any errors in the log files.
3. Turn debugging off as soon as you no longer need it, using the following command:

```
swadm delete_attribute 0 ProcessName ProcessInstance debug
```

1943 FIL session assigned to NULL connection

Description:	A database connection that was being used by a FIL session has been destroyed.
Appears In:	*SWDIR*\logs\sw_error
Severity:	2 (Severe)
Action:	Investigate if any database failures have occurred that might have caused a database failover.

1944 Failed to terminate invalid database connection

Description: A connection to the database has been flagged as invalid (for example, because it was timed out by the database), but the attempt to delete the invalid connection has failed.

Appears In: *SWDIR***logs\sw_error**

Severity: 4 (Information)

Action: Investigate why database connections are being terminated.

1945 Database error: *Error*

Description: iProcess has received the specified *Error* from the database.

Appears In: *SWDIR***logs\sw_warn**

Severity: Depending on the particular error, either:

- **1 (Critical)** - for example, the database has shut down,

  or

- 2 (Severe) - for example, cannot insert a row due to an integrity problem,

  or

- 3 (Warning), for example, cursor type changed.

Action: Refer to your database documentation for more information about the error, its possible causes and how to resolve it. Contact TIBCO Support if you require further assistance.

1946 Process failed to subscribe to events after *Num* attempts

Description: The process specified in the message header has failed to start because it cannot subscribe to the Microsoft Event sub-system, and so cannot publish or receive events.

Appears In: *SWDIR***logs****sw_error**

Severity: **1 (Critical)**

Action: To resolve the problem:

1. Restart the process.
2. If the problem persists, investigate why events are not working.

1947 Missing Version information from Instruction: *Instruction*

Description: The process specified in the message header has received an iProcess instruction that does not contain a procedure version field.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: Investigate what software created this instruction and why it does not contain the procedure version information. Contact the application developers for further assistance.

1948 Blank Version information from Instruction: *%s*

Description: The process specified in the message header has received an iProcess instruction that contains a blank procedure version field.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: Investigate what software created this instruction and why it does not contain the procedure version information. Contact the application developers for further assistance.

1949 Output template name for Case: *CaseNum*, Proc: *ProcName*, Step: *StepName* doesn't match name for subproc *SubProcName*

Description: The specified *StepName* is a graft step or dynamic sub-procedure call step, and has called the *SubProcName* sub-procedure. However, *StepName* and *SubProcName* are associated with different sub-procedure parameter templates, and so cannot map parameters between each other.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: Make sure that both *StepName* and *SubProcName* use the same sub-procedure parameter template.

See Also: “Defining Sub-Procedure Parameter Templates”, in the *TIBCO iProcess Modeler - Advanced Design Guide*, for more information about how to define and use sub-procedure parameter templates.

1950 Output template version for Case: *CaseNum*, Proc: *ProcName*, Step: *StepName* doesn't match version for subproc *SubProcName*

Description: The specified *StepName* is a graft step or dynamic sub-procedure call step, and has called the *SubProcName* sub-procedure. However, *StepName* and *SubProcName* are associated with different versions of the same sub-procedure parameter template, and so cannot map parameters between each other.

Appears In: *SWDIR***\logs\sw_error**

Severity: 2 (Severe)

Action: Make sure that both *StepName* and *SubProcName* use the same version of the same sub-procedure parameter template.

See Also: "Defining Sub-Procedure Parameter Templates", in the *TIBCO iProcess Modeler - Advanced Design Guide*, for more information about how to define and use sub-procedure parameter templates.

1951 Error(*Num*) finding main case number for case *CaseNum* of procedure *ProcName*

Description: An error *Num* has occurred while attempting to find the top level parent case for a sub-procedure call tree, where *Num* is one of the following:

- **-1**, if the parent case of a sub-procedure cannot be found.
- **-4**, if a sub-case does not contain a SW_PARENTREF field, or if the procedure named in the field does not exist.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Investigate why the parent case for the specified *CaseNum* cannot be found and why the system has got in this state.

1952 **Input template name for Case:** *CaseNum***, Proc:** *ProcName***, Step:** *StepName* **doesn't match name for sub-procedure** *SubProcName*

Description: The specified *StepName* is a graft step or dynamic sub-procedure call step, and has called the *SubProcName* sub-procedure. However, *StepName* and *SubProcName* are associated with different versions of the same sub-procedure parameter template, and so cannot map parameters between each other.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: Make sure that both *StepName* and *SubProcName* use the same version of the same sub-procedure parameter template.

See Also: “Defining Sub-Procedure Parameter Templates”, in the *TIBCO iProcess Modeler - Advanced Design Guide*, for more information about how to define and use sub-procedure parameter templates.

1953 Failed to register our RPC number *Num* with RPC server *ProblemInfo*

Description: The RPC listener process (RPC_TCP_LI or RPC_UDP_LI) cannot process the RPC port that it is listening on. *ProblemInfo* describes the nature of the problem.

If the problem affects the RPC background process (RPCBG), the problem only occurs when that process is being used - for example, if synchronous JumpTo or Suspend case operations are being done.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: Investigate which RPC ports are in use (using **rpcinfo**).

On UNIX **rpcinfo** is a system command. On Windows it can be found in SWDIR**rpc**.

If the RPC ports are in use by another program, this can be caused by a clash between programs. iProcess processes that listen on dynamic RPC ports listen on an RPC port that is based on their PID (+0x40000000). Alternatively, a process that was using such an RPC port has died or been killed, but the RPC port has not been released. The port must therefore be deleted before it can be registered by a new process.

Use **rpcinfo -t** to check if a process is listening on an RPC port. If there is no response, you can use **rpcinfo -d** to delete the port.

1954 Script expression @ line *LineNum* failed: *Expression-Line*

Description: The process specified in the message header has detected a syntax error in a script that it was processing.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe) or 3 (Warning)

The severity must be determined by the script developer.

Action: Correct the expression in the script.

1955 Minimum Port/RPC numbers for ranging could not be allocated (Required: *Num* Acquired: *Num*)

Description: The RPC Listener process could not allocate as many ports as it has been configured to allocate in port-ranging mode, as defined by the RNGTHRESHOLD parameter in the *SWDIR***etc****staffcfg** file.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: Investigate why the configured number of ports cannot be allocated.

See Also: The following:

- "RNGTHRESHOLD", in the *TIBCO iProcess Engine Administrator's Guide*, for a description of this parameter.
- "Setting up Port Range Filtering", in the *TIBCO iProcess Engine Architecture Guide*, for a description of how to use port-range filtering when using iProcess through a firewall.

1956 RPC - Client attempted to LOCK file *FileName*

Description: Either iProcess Workspace, or a 3rd-party SAL Application accessing iProcess via RPC, has attempted to lock the specified *FileName* using the FIL_CEXCLUSIVE flag.

This should not be done, due to the risk of the client not releasing the lock correctly

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: Investigate the source of the request to lock the file. If it comes from:

- a TIBCO iProcess Workspace, contact TIBCO Support.
- a 3rd-party application, contact technical support organization for that application.

1957 QINFO import: Could not delete existing CDQP definitions: *ErrorCode*

Description: A *SWDIR***bin\swutil QINFO** command has been used to import updated CDQP configuration data to the iProcess Engine, but an error has occurred while attempting to delete the existing CDQP definitions.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,q=255,TIME=ABS,FLAGS=U
```

2. Investigate any errors in the *SWDIR***logs\pstaffifc01.log** log file.
3. Disable debugging on the **staffifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

See Also: “Case Data Queue Parameters” in the *TIBCO iProcess swutil and swbatch: Reference Guide*, for more information about the *SWDIR***bin\swutil QINFO** command.

1958 **QINFO import: Could not add global definition:** *ErrorCode*

Description: A *SWDIR***bin\swutil QINFO** command has been used to import updated CDQP configuration data to the iProcess Engine, but an error has occurred while attempting to add a new CDQP definition.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,q=255,TIME=ABS,FLAGS=U
```

2. Investigate any errors in the *SWDIR***logs\pstaffifc01.log** log file.
3. Disable debugging on the **staffifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

See Also: "Case Data Queue Parameters" in the *TIBCO iProcess swutil and swbatch: Reference Guide*, for more information about the *SWDIR***bin\swutil QINFO** command.

1959 QINFO import: Could not initialise CDQP access for *QueueName*: *ErrorCode*

Description: A *SWDIR***bin****swutil QINFO** command has been used to import updated CDQP configuration data to the iProcess Engine, but an error has occurred while attempting to initialize access to, or load up the existing CDQP configuration.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,q=255,TIME=ABS,FLAGS=U
```

2. Investigate any errors in the *SWDIR***logs****pstaffifc01.log** log file.
3. Disable debugging on the st**affifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

See Also: "Case Data Queue Parameters" in the *TIBCO iProcess swutil and swbatch: Reference Guide*, for more information about the *SWDIR***bin****swutil QINFO** command.

1960 QINFO import: Could not add CDQP for *QueueName*: *ErrorCode*

Description: A *SWDIR***bin****swutil QINFO** command has been used to import updated CDQP configuration data to the iProcess Engine, but an error has occurred while attempting to add a new CDQP configuration setting.

Appears In: *SWDIR***logs****sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Enable debugging on the **staffifc.exe** (on Windows) or **Pstaffifc** (on UNIX) process, using the following debug string:

```
T=255,q=255,TIME=ABS,FLAGS=U
```

2. Investigate any errors in the *SWDIR***logs****pstaffifc01.log** log file.
3. Disable debugging on the **staffifc.exe** (or **Pstaffifc**) process as soon as you no longer need it.

See Also: "Case Data Queue Parameters" in the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the *SWDIR***bin****swutil QINFO** command.

1961 Could not open file *FileName*

Description: A *SWDIR***bin\swutil QINFO** command has been used to import new or updated CDQP configuration data to the iProcess Engine. However, the command could not open the specified *FileName*, which is either:

- the *SWDIR***logs\qinfoimp.log** log file, in which case no log information will be available for the command.

  or

- the name of the CDQP control file, in which case no CDQP commands will have been processed by the command.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Make sure that:
   - — the name of the CDQP control file (if specified) is correct, and is a full path name.
   - — the user that you are executing this command as has access to create and write to the *SWDIR***logs\qinfoimp.log** log file.
2. Re-run the *SWDIR***bin\swutil QINFO** command.

See Also: "Case Data Queue Parameters" in the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the *SWDIR***bin\swutil QINFO** command.

1962 Could not open stdin for input

Description: The following command has been used to import new or updated CDQP configuration data to the iProcess Engine, reading CDQP commands from the command line:

```
SWDIR\bin\swutil QINFO -
```

However, the program could not access its Standard Input. Consequently, no CDQP commands have been processed.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 2 (Severe)

Action: To resolve the problem:

1. Put the desired CDQP commands in a CDQP control file.
2. Use the following command, which reads the CDQP commands from the CDQP control file:

```
SWDIR\bin\swutil QINFO ControlFileName
```

See Also: "Case Data Queue Parameters" in the *TIBCO iProcess swutil and swbatch Reference Guide*, for more information about the *SWDIR*\bin\\swutil QINFO command.

1967 REQUEST: without procedure version information: *RequestData*

Description: A WIS process has received a REQUEST instruction for a new work item. The instruction cannot be processed because it does not contain all the information required by a Version 10 WIS process.

Appears In: *SWDIR***logs\sw_warn**

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you have recently upgraded the iProcess Engine, check that all the processes have been upgraded successfully.
2. Use the following command to set debugging for the **BG** processes:

```
swadm set_attribute 0 bg 0 debug
T=255,c=255,m=41,TIME=ABS,FLAGS=U
```

3. Use the following command to set debugging for the **WIS** processes:

```
swadm set_attribute o wis 0 debug T=255,R=255,m=41,
TIME=ABS,FLAGS=U
```

⚠ Turning on debugging can have a significant impact on system performance. Please see Debugging on page 6 before using it.

4. Reproduce the problem.
5. Send the log files to TIBCO Support for further assistance.
6. Turn debugging off as soon as you no longer need it, using the following commands:

```
swadm delete_attribute 0 bg 0 debug
swadm delete_attribute 0 wis 0 debug
```

See Also: “Set a Process Attribute”, in the *TIBCO iProcess Engine: Administrator's Guide*, for more information about the *SWDIR***util\swadm** command.

1968 WQS_WIS_USER_COUNT ignored - too big

Description: The WQS_WIS_USER_COUNT process attribute defines the number of WIS processes that should be dedicated to handling user queues and group queues.

The current (absolute) value of this attribute attempts to dedicate more WIS processes than there are available to handling user queues, and has therefore been ignored.

Queues are being allocated to WIS processes alphabetically, irrespective of whether they are user or group queues (either by round robin or on-demand allocation).

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. Recalculate the maximum value for the WQS_WIS_USER_COUNT attribute. This is:
   a. the number of WIS processes configured on the system, *minus*
   b. the number of WIS processes dedicated to handling explicitly specified queues (using the SW_WISINST queue attribute), *minus*
   c. one. One WIS must be reserved to handle group queues.

   For example, if you have 10 WIS processes, none of which are explicitly dedicated to handling specific queues, the maximum number of WIS processes available to handle user queues is 10-0-1=9, so the maximum value for WQS_WIS_USER_COUNT is 9.
2. Use the *SWDIR***util\swadm set_attribute** command to reset WQS_WIS_USER_COUNT to a value that is *less than or equal to* your calculated maximum value.

You can specify WQS_WIS_USER_COUNT as a percentage instead of an absolute value if you wish. This may be more appropriate if you regularly change the number of WIS processes on your system.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- "WQS_WIS_USER_COUNT", for a detailed description of this process attribute.
- "Controlling the Assignment of Queues to WIS Processes", for information about different methods of assigning queues to WIS processes.

1969 WQS_WIS_USER_COUNT ignored - percentage too big

Description: The WQS_WIS_USER_COUNT process attribute defines the number of WIS processes that should be dedicated to handling user queues and group queues.

The current (percentage) value of this attribute attempts to dedicate more WIS processes than there are available to handling user queues, and has therefore been ignored.

Queues are being allocated to WIS processes alphabetically, irrespective of whether they are user or group queues (either by round robin or on-demand allocation).

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. Recalculate the maximum value for the WQS_WIS_USER_COUNT attribute. This is:
   a. the number of WIS processes configured on the system, *minus*
   b. the number of WIS processes dedicated to handling explicitly specified queues (using the SW_WISINST queue attribute), *minus*
   c. one. One WIS must be reserved to handle group queues.

   For example, if you have 10 WIS processes, none of which are explicitly dedicated to handling specific queues, the maximum number of WIS processes available to handle user queues is 10-0-1=9, so the maximum value for WQS_WIS_USER_COUNT is 90%.
2. Use the *SWDIR***util\swadm set_attribute** command to reset WQS_WIS_USER_COUNT to a value that is *less than or equal to* your calculated maximum value.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- "WQS_WIS_USER_COUNT", for a detailed description of this process attribute.
- "Controlling the Assignment of Queues to WIS Processes", for information about different methods of assigning queues to WIS processes.

1970 WQS_WIS_USER_COUNT ignored as not enough WIS processes

Description: The WQS_WIS_USER_COUNT process attribute defines the number of WIS processes that should be dedicated to handling user queues and group queues.

The current value of this attribute attempts to dedicate at least one WIS process to user queues and one to group queues. However, there are only zero or one WIS processes available to handle user/group queues. (If more WIS processes exist, they have been dedicated to handling specific work queues.) WQS_WIS_USER_COUNT has therefore been ignored.

Queues are being allocated to WIS processes alphabetically, irrespective of whether they are user or group queues (either by round robin or on-demand allocation).

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve this problem, do one of the following:

- Add more WIS processes to the system.
- Reduce the number of WIS processes that are dedicated to handling explicitly specified queues (using the SW_WISINST queue attribute).
- Delete the WQS_WIS_USER_COUNT process attribute.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- “WQS_WIS_USER_COUNT”, for a detailed description of this process attribute.
- “Controlling the Assignment of Queues to WIS Processes”, for information about different methods of assigning queues to WIS processes (including how to specifically assign queues using the SW_WISINST queue attribute).

1971 SW_WISINST=*WISInstance* for queue:*QueueName* is too big (max=*MaxWISInstance*) - ignored

Description: The SW_WISINST queue attribute defines the WIS instance that is dedicated to handling this work queue.

The specified *WISInstance* value for queue *QueueName* is greater than the highest value WIS instance *MaxWISInstance* that is currently defined. SW_WISINST has therefore been ignored, and *QueueName* is being dynamically allocated to a WIS process.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (warning)

Action: To resolve this problem, do one of the following:

- Add more WIS processes to the system, so that there is a WIS process with an instance number that matches the *WISInstance* value for the *QueueName* queue.
- Change the SW_WISINST attribute value for the *QueueName* queue to the instance number of a WIS process that does exist.
- Blank the SW_WISINST attribute value for the *QueueName* queue so that the system dynamically assigns the queue to a WIS process.

See Also: “Controlling the Assignment of Queues to WIS Processes” in the *TIBCO iProcess Engine: Administrator’s Guide*, for information about different methods of assigning queues to WIS processes (including how to specifically assign queues using the SW_WISINST queue attribute).

1972 SW_WISINST=*WISInstance* for queue:*QueueName* is assigned to a disabled WIS - ignored

Description: The SW_WISINST queue attribute defines the WIS instance that is dedicated to handling this work queue.

The specified WIS instance is currently disabled. SW_WISINST has therefore been ignored, and *QueueName* is being dynamically allocated to a WIS process.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 3 (Warning)

Action: To resolve this problem, do one of the following:

- Enable the specified WIS instance.
- Change the SW_WISINST attribute value for the *QueueName* queue to the instance number of an enabled WIS process.
- Blank the SW_WISINST attribute value for the *QueueName* queue so that the system dynamically assigns the queue to a WIS process.

See Also: "Controlling the Assignment of Queues to WIS Processes" in the *TIBCO iProcess Engine: Administrator's Guide*, for information about different methods of assigning queues to WIS processes (including how to specifically assign queues using the SW_WISINST queue attribute).

1973 WQS Cannot start because no WIS processes configured

Description: The WQS process cannot run because there are no WIS processes configured. iProcess cannot process work to or from work queues.

Appears In: *SWDIR***logs****sw_error**

Severity: **1 (Critical)**

Action: To resolve the problem:

1. Reconfigure the iProcess Engine so that you have sufficient WIS processes. Make sure that the processes are enabled.
2. Restart the iProcess Engine.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- "Administering iProcess Engine Server Processes", for information about configuring and starting WIS processes.
- "Controlling the iProcess Engine", for information about how to restart the iProcess Engine.

1974 Failed to get shared memory handle for WQSINDEX

Description: iProcess failed to create or attach to the WQSINDEX shared memory segment.

This can happen if the operating system is not configured correctly - for example, if the size of the shared memory segment that iProcess is trying to allocate is too big or too small for the operating system.

Appears In: *SWDIR*\logs\sw_error

Severity: **1 (Critical)**

Action: To resolve the problem:

1. Use the following command to set debugging for the WQS process:

```
swadm set_attribute 0 wqs 0 debug T=255,W=255,TIME=ABS,FLAGS=U
```

> Turning on debugging can have a significant impact on system performance. Please see Debugging on page 6 before using it.

2. Try to restart the WQS process.
3. Search the debug log for lines in the following format, which identifies both the affected shared memory segment and the *errno* returned by the operating system.

```
W001:sw_shm_create () : shmget(shared_memory_key) failed: <Errno>
```

> If you are running the iProcess Engine on UNIX, you can look up the *Errno* in the system include file (**/usr/include/sys/errno.h**) to find the name of the error code, then look up the error in the **man** pages for the **shmget** call.

Other W*xxx*: messages may provide further information about the cause of the problem.

4. If the cause of the error is still unclear, send the log files to TIBCO Support for further assistance.

5. Turn debugging off as soon as you no longer need it, using the following command:

```
swadm delete_attribute 0 wqs 0 debug
```

See Also: "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about how to restart the WQS process.

1975

WQS initialise failed, connected to shared memory for *NodeName*

Description: The **WQS** process has failed to start because of a shared memory conflict. This can happen if, for example, the iProcess Engine is restarted and another application allocates to itself the shared memory key that iProcess expects to use when it restarts.

NodeName is either:

- blank, if the conflict is with a non-iProcess application, or
- the node name of another iProcess Engine that is running on the same computer, if the WQS process has connected to the WQSINDEX shared memory for that iProcess node.

This error message is only applicable on UNIX systems.

Appears In: *SWDIR***logs\sw_error**

Severity: **1 (Critical)**

Action: To resolve this problem, do one of the following:

- If the shared memory is from an iProcess Engine node that is no longer required, remove it by using the UNIX **ipcrm** utility or by rebooting the computer.

  You can use the UNIX **ipcs** utility to list shared memory segments.

- If the shared memory conflict is with another iProcess Engine node on the same computer, use the *SWDIR***util\swadm** command to change the value of the SHMKEY_ID process attribute for this iProcess Engine. This value is a single character (the default is “x”) that is used to build the shared memory key.

See Also: The following sections in the *TIBCO iProcess Engine: Administrator's Guide*:

- “SHMKEY_ID”, for a detailed description of this process attribute.
- “Set a Process Attribute”, for more information about the *SWDIR***util\swadm** command.

1976 *QueueName*:SW_WISINST = '*WISInstance*' - Value is invalid

Description: The SW_WISINST queue attribute defines the WIS instance that is dedicated to handling this work queue.*WISInstance* must be a number greater than 0 - this message occurs if it is not.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve this problem, do one of the following:

- Change the SW_WISINST attribute value for the specified *QueueName* to the instance number of the enabled WIS process that is to be dedicated to this queue.
- Blank the SW_WISINST attribute value for the specified *QueueName* so that the system dynamically assigns the queue to a WIS process.

See Also: "Controlling the Assignment of Queues to WIS Processes" in the *TIBCO iProcess Engine: Administrator's Guide*, for information about different methods of assigning queues to WIS processes (including how to specifically assign queues using the SW_WISINST queue attribute).

1977 WQS - Failed to register our RPC number (*RPCNumber*) with RPC Server (rcode = *Code*)

Description: The **WQS** process failed to register its address with the RPC listener process (**RPC_TCP_LI** or **RPC_UDP_LI**). Other iProcess processes therefore cannot communicate with the **WQS** process and hence cannot access work queues.

Possible causes for this error are:

- Another process is using the RPC port that the **WQS** process is trying to use. iProcess processes try to use RPC ports based on their process IDs (PID) + 0x40000000.
- A process was using the RPC port that the **WQS** process is trying to use. That process has died or been killed, but the RPC port has not been released. The port must therefore be deleted before it can be registered by the **WQS** process.

Appears In: *SWDIR***logs\sw_warn**

Severity: **1 (Critical)**

Action: To resolve the problem:

1. Make sure that the TCP and UDP RPC listener processes are running. Restart them if necessary.
2. Investigate which RPC ports are in use (using **rpcinfo**); delete any that are not being used but have not been released.

On UNIX **rpcinfo** is a system command. On Windows it can be found in SWDIR**rpc**. Use **rpcinfo -t** to check if a process is listening on an RPC port. If there is no response, you can use **rpcinfo -d** to delete the port.

See Also: "Administering iProcess Engine Server Processes", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about how to restart the RPC listener processes.

1978 (*Value*):SW_WISINST value (*Value*) doesn't match already allocated WIS (*WISInstance*)

Description: This message is generated by the **WQS** process when there is a mismatch between the *WISIntance* that is currently handling a queue and the *Value* of the SW_WISINST attribute. The SW_WISNST attribute is used to specify a *WISinstance* that a work queue is to be handled by. On a MOVESYSINFO the value of SW_WISINST is checked for all queues. For existing queues that are already being handled by a **WIS** process, any mismatch between the value of the SW_WISINST attribute and the **WIS** process handling it, is logged.

This message is for information only. It's to inform the administrator of the system that a queue is not being handled by the *WISinstance* that the SW_WISINST attribute *Value* is set to. There is no action that you can take as you cannot move a queue from one **WIS** process to another.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: No action required.

See Also: "Controlling the Assignment of Queues to WIS Processes" in the *TIBCO iProcess Engine: Administrator's Guide*, for information about different methods of assigning queues to WIS processes.

1979 **IAPJMS Failed to send audit message for Case: (*CaseNum*) Event: (*EventName*) Reason: (*ReturnCode*)**

Description: When the activity occurred, the message that should have been output has not been sent where:

- *CaseNum* is the number of the case.
- *EventName* is the name of the activity.
- *ReturnCode* is an internal return code from the software.

This means that the message has failed. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc\IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java **.jar** files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: For more information about administering activity monitoring, see "Administering Activity Monitoring" in TIBCO iProcess Engine Administrator's Guide.

1980 IAPJMS Message receipt (*Receipt*) invalid. MsgID (*MessageID*) returned receipt value (*ReceiptValue*)

Description: If the JMS delivery method is synchronous, when the message is sent, a receipt is requested. The **BG** process waits until the **IAPJMS** process has confirmed the message has been published. If the message that has been received is invalid, this error is reported where:

- *Receipt* is a unique ID.
- *MessageID* is the unique ID of the message.
- *ReceiptValue* is one of the following:
  - **01** = Failed to initialize sender. This means that the **IAPJMS** process failed to initialize.
  - **02** = Mismatched message. This means that your JMS delivery method is synchronous and you have received a receipt message for the wrong message.

This means that the message has failed. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java **.jar** files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: For more information about administering activity monitoring, see "Administering Activity Monitoring" in TIBCO iProcess Engine Administrator's Guide.

1981 IAPJMS Message receipt receive failed reason (*ReturnCode*)

Description: If the JMS delivery method is synchronous, when the message is sent, a receipt is requested. The **BG** process waits until the **IAPJMS** process has confirmed the message has been published. If the message is not published, this error is reported. *ReturnCode* is an internal return code from the software.

This means that the message has failed. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java **.jar** files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: For more information about administering activity monitoring, see "Administering Activity Monitoring" in TIBCO iProcess Engine Administrator's Guide.

1982 XML message error due to (*Reason*) in (*FileName*)

Description: This error is generated when the **swutil EXPMONITOR** command is run and if there is an error with the XML. The error means that the XML cannot be validated where:

- *Reason* is the error message that describes why the XML is not valid.
- *FileName* is the full pathname of the file that contains the error.

See "Exporting Activity Monitoring Configuration" in the *TIBCO iProcess swutil and swbatch: Reference Guide* for more information about the **swutil EXPMONITOR** command.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe)

Action: To resolve the problem, check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: “Understanding the Activity Monitoring Schemas”, in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1983 XML caught exception due to (*Reason*) in (*FileName*)

Description: An exception occurred in the XML where:

- *Reason* is the error message that describes why the XML is not caused an exception.
- *FileName* is the full pathname of the file that contains the error.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem, check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1984 The queue (*QueueName*) is currently being cached by the WIS. The *MsgName* message will be forwarded later (casenum=*CaseNum* reqid=*ReqID*).

Description: The **WISMBD** process has tried to forward the *MsgName* message from a **BG** process to the **WIS** process handling the *QueueName* queue. However, the **WIS** process was caching the *QueueName* queue and so returned an ER_CACHING response to the **WISMBD** process, indicating that it could not accept the message.

MsgName is the name of the message that is being forwarded (for example, REQUEST or PURGE).

Appears In: *SWDIR*\logs\\sw_warn

Severity: 3 (Warning)

Action: No action required. The **WISMBD** process will automatically attempt to resend the *MsgName* message again later.

See Also: "Configuring When WIS Processes Cache Their Queues" in the *TIBCO iProcess Engine: Administrator's Guide*.

1985 XML Failed to find information for field (*FieldName*)

Description: The XML that you have created to configure your activity monitoring information has not found any information for the field.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

- Check that the field exists.
- Check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1986 IAPJMS Failed to find procedure configuration data for Case (*CaseNum*) Event (*EventName*) Reason (*ReturnCode*)

Description: The **IAPJMS** process failed to find activity monitoring configuration for a case of a procedure where:

- *CaseNum* is the unique ID of the case.
- *EventName* is the name of the activity.
- *ReturnCode* is an internal return code from the software.

This means that the message has not been sent. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. Check the activity monitoring schemas to make sure that your XML conforms to them.
2. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
3. Check that the JMS service is running.
4. Check that the **IAPJMS** process is running.
5. Check that the JMS server is running.
6. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
7. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java **.jar** files you should specify.
8. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: For more information, see:

- "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

- "Configuring Activity Monitoring", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about configuring activity monitoring.

1987 XML Failed to access procedure's IAP cache (*FileName*) reason (*ReturnCode*)

Description: An error occurred when trying to monitor activities for this procedure where:

- *FileName* is the full pathname of the file that contains the error.
- *ReturnCode* is an internal return code from the software.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem, check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1988 XML Failed to process line for procedure (*ProcedureName*) activity (*Activity*) and field (*FieldName*) reason (*ReturnCode*)

Description: There is an error with the XML which means that it cannot be processed.

- *ProcedureName* is the name of the procedure.
- *Activity* is the name of the activity being monitored.
- *FieldName* is the name of the field being monitored.
- *ReturnCode* is an internal return code from the software.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem, check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: “Understanding the Activity Monitoring Schemas”, in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1989 XML Failed to process line for procedure (*ProcedureName*) activity (*Activity*) and step (*StepName*) reason (*ReturnCode*)

Description: There is an error with the XML which means that it cannot be processed.

- *ProcedureName* is the name of the procedure.
- *Activity* is the name of the activity being monitored.
- *StepName* is the name of the step being monitored.
- *ReturnCode* is an internal return code from the software.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem, check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1990 IAP JMS Failed to load Java library (*Library*), please refer to the documentation for IAP.

Description: The Java library failed to load where *Library* is the Java library that failed to load.

*SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

- Check that you have installed the Java library correctly. Refer to the documentation supplied with your Java library for more information.
- If you are using Windows, check that there are no missing DLLs that the Java library needs.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1991 XML Invalid procedure (*ProcedureName*) specified.

Description: You have specified an invalid procedure name in the XML that you have generated to configure your activity monitoring.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

- Check that the procedure exists.
- Check the activity monitoring schemas to make sure that your XML conforms to them.

See Also: "Understanding the Activity Monitoring Schemas", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about the activity monitoring schemas.

1992 XML Duplicate field (*FieldName*) specified for procedure (*ProcedureName*) in monitor details

Description:	You have specified a duplicate field in the XML you have generated to configure your activity monitoring information.
Appears In:	*SWDIR*\logs\sw_error
Severity:	2 (Severe)
Action:	Remove the duplicate field from your activity configuration information.
See Also:	"Configuring Activity Monitoring", in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about configuring activity monitoring.

1993 XML Duplicate activity (*Activity*) / step (*StepName*) specified in monitor detail

Description: You have specified a duplicate activity or step in the XML you have generated to configure your activity monitoring information where:

- *Activity* is the name of the activity.
- *StepName* is the name of the step.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: Remove the duplicate activity of step from your activity monitoring configuration information.

See Also: “Configuring Activity Monitoring”, in the *TIBCO iProcess Modeler - Integration Techniques Guide*, for information about configuring activity monitoring.

1994 IAP JMS Message send failed due to timeout (*TimeOut*)

Description: A message took too long to send so it timed out where *TimeOut* is the number of seconds that it took for the message to timeout.

This means that the message has failed. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 3 (Warning)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR\logs\\std_err.log file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR*\etc\\IAPJMS.properties file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1995 IAP JMS Message receive failed due to timeout (*TimeOut*)

Description: If the JMS delivery method is synchronous, when the message is sent, a receipt is requested. The BG process waits until the **IAPJMS** process has confirmed the message has been published. If the receipt message takes too long to send then it times out where *TimeOut* is the number of seconds it took for the message to time-out.

This means that the receipt message has failed. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1996 Failure in JPROC Layer (*Location*)

Description: There has been a failure in the Java layer, for example, because of some missing DLLs where *Location* is the location of the error.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe)

Action: To resolve the problem:

- If you are using Windows, check that there are no missing DLLs.
- Check that any external dependencies required for your application server are set up. Refer to the documentation supplied with your application server for more information.
- Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1997 IAP JMS Initialisation failure. Reason: (*ReturnCode*)

Description: The **IAPJMS** process failed to initialize where *ReturnCode* is an internal return code from the software.

This means that the message has failed to send. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1998 IAP JMS Failed to send message. Reason: (*ReturnCode*)

Description: The **IAPJMS** process failed to send a message where *ReturnCode* is an internal return code from the software.

This means that the message has failed to be sent. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

1999 IAP JMS Failed to start due to an environment issue. Reason: (*ReturnCode*)

Description: The **IAPJMS** process failed to start due to an environment issue where *ReturnCode* is an internal return code from the software.

This means that the message has not been sent. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR\logs\std_err.log file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR*\etc\\IAPJMS.properties file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

2000 IAP JMS Failed to start due to a JVM creation error. Error code: (*Code*), Reason: (*Reason*)

Description: The **IAPJMS** process failed to start due to a JVM error, where *Code* is an internal error code from the software and *Reason* is the error message that describes the failure.

This means that the message has not been sent. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR***logs****sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR**logs****std_err.log** file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR***etc****IAPJMS.properties** file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

2001 ’*EAIType*’ EAI Validation-Time Plug-in failed to initialize

Description: The iProcess Engine failed to load the specified *EAIType* EAI Server Plug-in.

The BG processes will continue to run normally. However, transactions that attempt processing of EAI step types handled by the *EAIType* EAI Server Plug-in will be aborted and rolled back.

> The transaction will be automatically retried a number of times. On SQL Server and DB2 versions, this is defined by the IQL_RETRY_COUNT process attribute value. On Oracle versions the retry parameters are attributes of the Oracle AQ queues.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check that the specified EAI Server Plug-in exists *and* that the necessary access permissions exist on the library.
2. Use the *SWDIR***util\sweaireg** utility to check that the plug-in is correctly registered.
3. Check that the operating-system specific shared library path requirements are satisfied. For example, on Solaris, ensure that the LD_LIBRARY_PATH environment variable is set correctly when the process sentinels are started.

See Also: “Managing EAI Step Server Plug-ins”, in the *TIBCO iProcess Engine Administrator’s Guide*, for information about how to use the *SWDIR***util\sweaireg** utility.

2002 EAI_WEBSERVICES: SocketProxy server status change: Server number (*Server*), Hostname: (*Hostname*), Port Number (*Port*). (*Message*)

Description: This message occurs when iProcess becomes aware that any one of the Jetty servers in a High Availability environment has become available, or has become unavailable.

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: If the *Message* indicates that a server has gone offline, investigate the reason and ensure that it remains available whenever possible.

See Also: "Manually Configuring the HTTP Proxy Server Settings" in *TIBCO iProcess Web Services Plug-in User's Guide*.

2003 EAI_WEBSERVICES: All SocketProxy servers are offline, or they are timed out after being offline.

Description: This error occurs if the iProcess Web Services Client Plug-in fails to communicate with the Web Services Engine component (which resides within Jetty). It may occur for one of the following reasons:

- Because none of the Jetty servers configured are currently available, either because of a configuration error or because they are not running. (There is usually one Jetty server, but there can be more in a High Availability environment.)
- In a High Availability environment, because all of the servers configured have been recently tagged as unavailable. (In High Availability mode, any server tagged as unavailable is left uncontacted for a pre-determined duration, which defaults to 5 minutes. However, if no servers are available, this retry timer is ignored and all servers are retried regardless of when they were last checked.)

Appears In: *SWDIR***logs\sw_warn**

Severity: 3 (Warning)

Action: To resolve the problem, ensure that Jetty server(s) are available for use by the iProcess Web Services Client Plug-in. Also ensure that they remain available whenever possible.

See Also: "Manually Configuring the HTTP Proxy Server Settings", in the *TIBCO iProcess Web Services Plug-in User's Guide*.

2004 PROCMGR: No (*value*) entry has been found in /etc/services file. Using default port (*port*). See installation guide.

Description: The **procmgr** process has not found a port defined for (*value*). It is therefore using a default port.

Appears In: *SWDIR***logs\sw_error**

Severity: 3 (Warning)

Action: In the UNIX **/etc/services** (or Windows **%systemroot%\system32\drivers\etc\services**) file, add entries to assign unique (system-wide) TCP/IP port numbers for the affected processes.

See Also: The appropriate *TIBCO iProcess Engine Installation Guide*, for more information about **/etc/services.**

2005 IAP JMS Failed to process messages due to nodename mismatch. Reason: (*ReturnCode*)

Description: The **IAPJMS** process failed to process a message because

This means that the message has not been sent. Whether or not the failed message transaction is rolled back depends on how you have configured your IAPJMS_ROLLBACK process attribute. See "IAPJMS_ROLLBACK" in TIBCO iProcess Engine Administrator's Guide for more information.

Appears In: *SWDIR*\logs\\sw_warn

Severity: 3 (Warning)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR\logs\std_err.log file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check that the JMS server is running.
5. Check the properties are specified correctly in the *SWDIR*\etc\\IAPJMS.properties file.
6. Check the Java **.jar** files are specified correctly in the CLASSPATH variable. Look in the **IAPJMS.classpath** file to see which Java .jar files you should specify.
7. Set Java debug using Jakarta's **Log4J** logging facility. Information about **Log4J** can be found at:

   http://logging.apache.org/log4j/docs/index.html.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about activity monitoring.

2006 WQDJMS Failed to send delta publications for case (*case*). Reason: (*ReasonCode*).

Description: The **IAPJMS** process failed to send a Work Queue Delta message for the specified case, where *ReasonCode* is an internal return code from the software.

This means that the message has failed to be sent.

Appears In: *SWDIR*\logs\\sw_error

Severity: 2 (Severe)

Action: To resolve the problem:

1. If you are using UNIX, check the $SWDIR\logs\\std_err.log file for errors.
2. Check that the JMS service is running.
3. Check that the **IAPJMS** process is running.
4. Check the properties are specified correctly in the *SWDIR*\**etc\\IAPJMS**.properties file.
5. Check the Java **.jar** files are specified correctly in the **CLASSPATH** variable. Look in the **IAPJMS.classpath** file to see which Java **.jar** files you should specify.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about publishing Work Queue Deltas via JMS.

2007 WQDJMS IAPJMS_PORTNO and WQDJ MS_PORTNO are set to the same port number (*Port*). This could cause missing messages.

Description: The process attributes for the **IAPJMS_PORTNO** and the **WQDJMS_PORTNO** have identical values. If the server continued, this would cause incorrect JMS message routing.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Using **swadm**, change one of the two values to be a unique port number.
2. Restart the server.

See Also: "Monitoring Activities", in the *TIBCO iProcess Engine: Administrator's Guide*, for information about publishing Work Queue Deltas via JMS.

2008 WQDJMS Failed to (*Action*) WQD record for WQDID (*WQDID*) to database. Reason: (*ReasonCode*).

Description: The server failed either to add or to delete an entry for a Work |Queue Delta subscription. The *ReasonCode* should provide more information to resolve the issue.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Check the *ReasonCode* for further information.
2. Check the *SWDIR***logs\sw_error** file for any related database errors that may have caused the problem.

See Also: "Monitoring Activities" in *TIBCO iProcess Engine: Administrator's Guide*, for information about publishing Work Queue Deltas via JMS.

2009 WQDJMS Failed to invalidate WQD sessions on restart. Reason: (*ReasonCode*).

Description: The **WIS** process was restarted . However, outstanding subscriptions could not be flagged as invalid on restart. This means that there may be applications expecting Work Queue Deltas that have not been informed that the current subscription is now invalid.

Appears In: *SWDIR***logs\sw_error**

Severity: 2 (Severe)

Action: To resolve the problem:

1. Since this has been caused by a **WIS** process restart, attempt to resolve the problem that caused the **WIS** process failure.
2. Restart all listening applications and request a new subscription.

See Also: “Monitoring Activities” in *TIBCO iProcess Engine Administrator's Guide*, for information about publishing Work Queue Deltas via JMS.

Appendix A Socket Error Codes

Several system messages return a socket error code (*Errno*), which is either:

- a valid system error code returned from the operating system, or
- *Unknown error*, if the operating system has returned an error code that iProcess does not recognize.

For more information about these Errno codes, you should refer to your operating system socket documentation, as follows:

- On UNIX, the system include file (`/usr/include/sys/errno.h`) provides a brief definition of each *Errno*. The Intro(2) man page provides a complete list of the error numbers and their names as defined in `errno.h`.
- On Windows, the Windows Sockets Error Codes are documented in the MSDN. See http://msdn2.microsoft.com/en-us/library/ms740668.aspx.