

TIBCO® Staffware Process Objects (SPO) COM Client

Release Notes

*Software Release 10.2.0
February 2005*

Important Information

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

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

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

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

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

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

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

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

Copyright © 1999-2005 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

New Features	2
Version 10.2.0	2
Version 10.0(6.0)	5
Version 10.0(4.1)	8
Version 10.0(3.0)	10
Version 10.0(2.0)	11
Version 10.0(1.0)	13
Version 10.0(0.0)	21
Changes in Functionality	31
Version 10.0(4.2)	31
Version 10.0(2.0)	32
Compatibility	33
TIBCO SPO Server	33
TIBCO Staffware Server/Engine	33
Development Environment	34
Other Information	35
Must Pass a TextBox Control as a Function Parameter in a VB Application	35
Full Kanji Character Set Not Supported	35
License Count may be a Negative Number	36
Procedure Key must include Version Number	36
Closed Issues	37
Known Issues	50

Release Notes

Check the TIBCO Product Support web site at <http://support.tibco.com> for product information that was not available at release time. Entry to this site requires a username and password. If you do not have a username, you can request one. You must have a valid maintenance or support contract to use this site.

Topics

- *New Features, page 2*
- *Changes in Functionality, page 31*
- *Compatibility, page 33*
- *Other Information, page 35*
- *Closed Issues, page 37*
- *Known Issues, page 50*

New Features

This section lists new features in each release of the TIBCO SPO COM Client.



As we are transitioning from "Staffware" to "TIBCO," our version numbering system is also transitioning. Staffware version numbers included major, minor, maintenance release, and patch numbers, with parentheses (e.g., 10.2(0.0)). TIBCO version numbers include major, minor, and maintenance release numbers, without parentheses (e.g., 10.2.0). Hotfix numbers (the equivalent to a "patch") are not shown in the product version number. Until the transition is complete, you will see both numbering systems used in documentation.

Version 10.2.0

The following are new features in this release.

New TIBCO SPO Server Status Types (16814)

The following constants have been added to the **SWNodeInfoStatusType** enumeration to describe the statuses of TIBCO SPO Servers made available by a TIBCO SPO Director:

Constant	Description	Value
swNotRunning	The TIBCO SPO Server is not running	'N'
swStarting	The TIBCO SPO Server is starting	'I'
swShuttingDown	The TIBCO SPO Server is shutting down	'D'
swStopped	The TIBCO SPO Server is stopped and not available	'S'
swServerSuspended	The TIBCO SPO Server is suspended	'U'
swNoResponse	The TIBCO SPO Server is not responding	'R'

New Error Types Added to SWServerErrorType Enumeration (16667)

The following error types have been added to the SWServerErrorType enumeration:

Constant	Error Message	Hex Value
swSAL_InvValueErr	Invalid process attribute value	0x0022
swSAL_InvFilterErr	Invalid filter	0x0023
swSAL_XactAbortErr	Global transaction has been aborted	0x0024
swSAL_ComplexErr	Filter is too complex	0x0045

New Methods Added to Retrieve Markings When Locking Work Items (16177)

The following methods have been added to allow you to specify which markings to return from the TIBCO SPO Server, regardless of whether they are visible on the form or whether they are the result of conditional statements on the form:

- **LockItemMarkings** - This method on **SWWorkItem** contains a *MarkingNames* parameter that allows you to specify which markings to return from the server when the work item is locked. This allows you to control resources by specifying only the needed markings.
- **LockItemsMarkings** - This method on **SWWorkQ** contains a *MarkingNames* parameter that allows you to specify which markings to return from the server when the work items are locked. This allows you to control resources by specifying only the needed markings.

New Property Added to Specify Which CDQP Fields to Return from Server (12550)

The following property has been added to allow you to specify which Case Data Queue Parameters (CDQPs) to return from the server with work items that reside in an **SWXList**. This allows you to more closely control resources by returning only the CDQP fields that are needed.

- **CDQPNames** - This read/write property on **SWCriteriaWI** can be set to an array of strings, each specifying a CDQP to return. To specify that no CDQPs should be returned from the server, set this property to an empty string (""). To specify that all CDQPs should be returned, set this property to "&ALL&".

New Methods Added to Create Work Items and Specify CDQPs and Case Fields (10642)

The following methods have been added to allow you to create work items and be able to specify which Case Data Queue Parameters (CDQPs) and case fields to return from the server with the work items that are created:

- **MakeWorkItemEx** - This new method on **SWNode** extends the **MakeWorkItem** method by providing *CDQPNames* and *CaseFieldNames* parameters to specify the CDQPs and case fields to return from the server.
- **MakeWorkItemByTagEx** - This new method on **SWNode** extends the **MakeWorkItemByTag** method by providing *CDQPNames* and *CaseFieldNames* parameters to specify the CDQPs and case fields to return from the server.
- **MakeXListItemsEx** - This new method on **SWWorkQ** extends the **MakeXListItems** method by providing *CDQPNames* and *CaseFieldNames* parameters to specify the CDQPs and case fields to return from the server.

All three of these new methods allow you to more closely control resources by specifying only the CDQPs and/or case fields that you need. See the on-line help for specifics about the valid entries for the *CDQPNames* and *CaseFieldNames* parameters.

Version 10.0(6.0)

The following are new features in this release.

Extended Description Property Added to SWStep (16065)

The **DescriptionEx** property has been added to the **SWStep** object. This property returns the extended description that can be entered when defining a normal-type step in the SPD. This new property returns an empty string if the step type does not support an extended description.

Transaction Control Steps Added (16045)

SPO has been modified to support a new step type — transaction control steps. Transaction control steps provide a mechanism, within a TIBCO procedure, to allow more transaction granularity within a sequence of EAI steps.

By default, the background process groups a series of connected EAI steps into one transaction. If a failure occurs in any EAI step in the series, the entire transaction is rolled back. A transaction control step can now be placed within the series of EAI steps to break the single transaction into multiple transactions. When the process flow reaches the transaction control step, the current transaction can be either committed and a new transaction started, or the current transaction can be aborted, depending on how the transaction control step has been defined in the SPD.

See the *Defining Procedures - Integration Techniques* manual for more information about placement and use of transaction control steps in a procedure.

The following changes have been made in SPO to support transaction control steps:

- **SWStepType** - The following constant has been added to this enumeration:

Constant	Description	Value
swTransControl	Transaction Control Step.	'T'

This new constant is returned by the **Type** property on **SWStep** if the step is a transaction control step.

- **SWTransControlStep** - This new object represents an outstanding transaction control step. A transaction control step becomes outstanding when the process flow reaches the step. The transaction control step is no longer outstanding when the transaction started by the transaction control step is

either committed or aborted. This new object contains the following properties:

- **Arrived** - Returns the date and time the transaction control step became outstanding.
- **CaseNumber** - Identifies the case containing the outstanding transaction control step.
- **ClassId** - Identifies the object class.
- **Key** - Returns the key for the step, in the form:
ProcName | StepName | CaseNumber.
- **ProcMajorVer** - Returns the MajorVersion# portion of the procedure's version number.
- **ProcMinorVer** - Returns the MinorVersion# portion of the procedure's version number.
- **ProcName** - Returns the name of the procedure with which the step is associated.
- **ProcPath** - Provides the complete path to the outstanding transaction control step.
- **RetryTime** - Returns the date and time that the transaction will be retried if it fails.
- **StepName** - Returns the name of the transaction control step.
- **TransControlSteps** - This new property on **SWCase** returns an **SWList** of **SWTransControlStep** objects (see above), one for each outstanding transaction control step in the case.
- **TransControlType** - This new property on **SWStep** returns an enumeration constant (**SWTransControlType** - see below) that identifies the type of transaction control step. The type, which is specified when the step is defined in the SPD, are:
 - **Commit and Continue** - This type specifies that the current transaction be committed, and that a new transaction be started for subsequent steps using the same background process. The benefit of choosing this option is that it is faster, as the same process starts the new transaction.
 - **Commit and Concede** - This type specifies that the current transaction be committed, and that a new transaction be started for subsequent steps, except a different background process will be used for the second transaction.

The background process processes the first transaction and updates the database. It then sends a message back to the Mbox where the messages are

stored. Processing of the process continues when the background process (either the same one that processed the first transaction or another one) reads the message from the Mbox and processes it. The benefit of choosing this option is that it enables load balancing because a different background process can process the second transaction.

- **Abort** - This option causes the abortion of the current transaction when the process flow reaches the transaction control step. This option is always used with a condition. This means that you can specify a condition on which the transaction should be rolled back. A Transaction control step that is configured with the Abort option must always follow an EAI step. It cannot follow any other type of step.

The **TransControlType** property returns **swNA** if the step is not a transaction control step.

- **SWTransControlType** - This new enumeration has been added to support transaction control steps. It contains the following constants:

Constant	Description	Value
swNA	Not applicable for this step type.	'N'
swAbort	Abort transaction control step.	'A'
swContinue	Commit and continue transaction control step.	'C'
swConcede	Commit and concede transaction control step.	'D'

This new enumeration is returned by the **TransControlType** property on **SWStep**.

- **RetryDelay** - This new property on **SWStep** returns the number of minutes in which a transaction will be retried if it fails. This value, which is only applicable for **swContinue**-type transaction control steps, is specified when the transaction control step is defined with the SPD. A failed transaction will be retried one time.
- **SWAuditActionType** - The following new constants have been added to this enumeration to support transaction control steps:

Constant	Description	Value
swTransProcessed	Transaction Control Step Processed	54
swTransStarted	Transaction Control Step - New Transaction Started	55

Constant	Description	Value
swTransRestart	Transaction Control Step - Retry Time Expired	56
swTransAborted	Transaction Control Step Processed - Transaction Aborted	87

These new constants are added to the case's audit trail (**SWAuditStep.Action**) when an action is performed against a transaction control step.

Version 10.0(4.1)

Enumeration Constants Added to SWAuditActionType (15247)

The following constants have been added to the **SWAuditActionType** enumerations:

Constant	Description	Value
swReleasedNoAddressees	The step was released, although there is no addressee defined for the step.	39
swReleasedNoSubProcs	The step was released, although there are no sub-procedures defined for the step (applicable only for dynamic sub-procedure call steps and graft steps).	40

Allow Logging Into Multiple Instances of the TIBCO SPO Server (14985 and 15439)

Changes have been made to the TIBCO SPO COM Client to allow you to log into multiple instances of the TIBCO SPO Server. This results in all instances of the TIBCO SPO Server running from the same \$SWDIR directory and attaching to the same node (TIBCO Staffware Process/iProcess Engine). The following changes have been made to the TIBCO SPO COM Client to support this new functionality:

- **AddNodeEx** - This new method has been added to **SWEnterprise**. It provides an optional *InstanceNumber* parameter that is used to indicate a specific server instance to which you are directing the UDP message when the TIBCO SPO Server supports multiple instances.
- **MakeNodeInfoEx** - This new method has been added to **SWEnterprise**. It provides an optional *InstanceNumber* parameter that is used to indicate a specific server instance for which the **SWNodeInfo** object being created is to represent.

- **InstanceNumber** - This new property on **SWNodeInfo** returns the instance number of the TIBCO SPO Server represented by the **SWNodeInfo** object. This is only applicable to TIBCO SPO Servers that support multiple instances. For TIBCO SPO Servers that are not multiple-instance capable, this property returns 1.
- **Login** - This method has been modified to log the user into a specific instance of the TIBCO SPO Server if the *NodeKey* passed in the method call includes an instance number. If the *NodeKey* does not include an instance number, and the item is not found in the list of **NodeInfos**, the **Login** method will add instance "1" to the key and reattempt to locate the item. This allows applications that use the old key format (prior to the instance number being added) to access instance "1" of a server that has the new key format without modifying code.
- **Key** - This property on **SWNode** and **SWNodeInfo** now includes the instance number for TIBCO SPO Servers that support multiple instances, or if the **SWNodeInfo** object was created with the optional *InstanceNumber* parameter on **AddNodeEx** or **MakeNodeInfoEx**.
- **ItemByKey** - This method has been modified to accommodate SPO applications that have hard-coded keys that do not include an instance number. **ItemByKey** will add instance "1" to the key and re-attempt to return the item if it fails to find an item without an instance number. This allows applications that use the old key format to access instance "1" of a server that has the new key format without modifying code.
- **Tag** - This property on **SWNodeInfo** now returns a tag that includes the instance number of the TIBCO SPO Server. The instance number defaults to "1" in the tag if the TIBCO SPO Server does not support multiple instances.
The tag for an **SWNodeInfo** object has also been modified to always include the *IsDirector* flag: "Y" if the **SWNodeInfo** object represents a TIBCO SPO Director; "N" if the **SWNodeInfo** object represents a TIBCO SPO Server. (Previously, the *IsDirector* flag was blank if the **SWNodeInfo** object represented a TIBCO SPO Server.)

Ability to Limit Number of Cases Retrieved from TIBCO SPO Server Added (14985)

Changes have been made to the TIBCO SPO COM Client to allow you to limit the number of cases that are retrieved from the TIBCO SPO Server when using **SWXLists**. The following properties have been added to support this new functionality:

- **MaxCnt** - This new property on **SWCriteriaC** specifies the maximum number of cases to retrieve from the TIBCO SPO Server and place in the raw data buffers on the TIBCO SPO COM Client. Note that this property is initialized to -1, which causes all of the requested cases (that satisfy the filter expression, if specified) to be retrieved from the server and placed in the raw data buffers.

Setting this property allows you to prevent the wait that results from retrieving all cases when there is a very large number of cases.

- **OverMaxCnt** - This new property on **SWCriteriaC** indicates the number of cases excluded from the **SWXList** because they are in excess of the maximum requested number of items (**MaxCnt**).

Version 10.0(3.0)

Properties Added to Obtain Case Start Date and Time (14770)

The following properties have been added to the **SWCase** object so that you can obtain the date and time the case was started:

- **TimeStarted** - Returns the date and time the case was started.
- **TimeStartedOffset** - Returns the additional microseconds to calculate the exact time the case was started. For example, if the date/time returned by the **TimeStarted** property is October 4, 2001 09:13:40 (the time format is HH:MM:SS) and the **TimeStartedOffset** property contains 500, the case was started at 09:13:40 and 500 microseconds.

Method Added to Perform MOVESYSINFO Function (14619)

The **MoveSysInfo** method has been added to the **SWNode** object to allow the client application to explicitly call the **MOVESYSINFO** function. Prior to this new method, the TIBCO SPO Server called the **MOVESYSINFO** function whenever an administrative function was performed, i.e., any function that affected a user, group, role, attribute, or queue supervisor definition. This can tie up the background and WIS/WQS processes for long periods of time if there are lots of users.

This new method is used in conjunction with the **ImplicitMoveSysInfo** TIBCO SPO Server configuration parameter to control when the **MOVESYSINFO** function occurs. See your TIBCO SPO Server readme for information about this configuration parameter.

Version 10.0(2.0)

New Method Added to Set Default Criteria (14585)

The **SetDefCriteriaEx** method has been added to the **SWWorkQ** object. This method allows you to set the default filter and sort criteria for the work queue. This causes the criteria you pass in as a parameter to this method to persist on the current instance of the work queue, causing future **SWViews** or **SWXLists** of work items on that instance of the queue to use the default criteria.

Note that the existing **SetDefCriteria** method can also be used to establish default criteria; it causes the current setting of the **FilterExpression** property and the **SortFields** property on the **SWView** in the **WorkItems** property in the work queue to be used to establish the default criteria. The new **SetDefCriteriaEx** method is probably a better choice to set default criteria if you are using **SWXLists**.

New SWServerErrorType Enumerations for Prediction (14446)

The following new enumerations have been added to **SWServerErrorType** to support case prediction:

Constant	Error Message	Hex Value
swMSB_NoneErr	No future workitems found or no deadline info found.	0x0042
swMSB_SubProcErr	Sub-proc step type found.	0x0043
swMSB_Err	General error.	0x0044
swMSB_NoServerErr	TIBCO Staffware Process Engine unavailable.	0x0045
swMSB_BadAuthErr	User has no access to start proc.	0x0046
swMSB_BadPwdErr	Username or password is incorrect.	0x0047
swMSB_InvalidStepErr	Invalid Step.	0x0048
swMSB_InvalidProcErr	Invalid Procedure.	0x0049
swMSB_InvalidFormErr	Invalid form.	0x004A
swMSB_NoProcErr	Couldn't find procedure.	0x004B
swMSB_NoFieldErr	Couldn't locate named field.	0x004C

Constant	Error Message	Hex Value
swMSB_ExternalErr	Error from External elink call.	0x004D
swMSB_InvalidSessErr	Invalid session handle.	0x004E
swMSB_NoParamErr	Invalid parameter.	0x004F
swMSB_ProcNotOpenErr	Procedure is not open.	0x0050
swMSB_OutOfRangeErr	Request is out of range.	0x0051
swMSB_LockFailErr	Failed to lock session.	0x0052
swMSB_SessFailErr	Failed to allocate session.	0x0053
swMSB_WaitingErr	Waiting on other steps to execute.	0x0054
swMSB_MaxLoopsErr	Maximum prediction loop count exceeded.	0x0055
swMSB_LoggedInErr	Prediction service failed to login user.	0x0056
swMSB_NoTextErr	Undefined prediction error.	0x0057
swMSB_OverflowErr	The text string was longer than the passed in buffer size.	0x0058
swMSB_EOLErr	Beyond list or invalid index.	0x005A
swMSB_QErr	Invalid query string passed to find future workitems.	0x005C
swMSB_SortErr	Invalid sort parameter.	0x005D
swMSB_SubProcDepthErr	Sub procedure depth exceeded.	0x005E
swMSB_NoSALerr	No SAL session handle.	0x005F
swMSB_NoCaseRefErr	No case reference passed for a live case.	0x0060
swMSB_NoStepErr	Outstanding step passed in isn't in procedure.	0x0061
swMSB_StepNotFoundErr	Cannot find step in procedure in cc_exec_proc.	0x0062
swMSB_NotMainErr	Sub-procedure name invalid for simulate case.	0x0063

New SWServerErrorType Enumeration for Buffer Overflow (14158)

The following new **SWServerErrorType** enumeration is returned by the TIBCO SPO Server if an attempt is made to send a message larger than the value specified for the TIBCO SPO Server message response buffer (which is defined by the **TCPReponsePages** TIBCO SPO Server configuration parameter — the default is 2 pages (4K)).

Constant	Error Message	Hex Value
swBuffOverrunErr	Internal SPO buffer too small.	0x00DF

Version 10.0(1.0)

Properties Added to List Supervised Work Queues (13785)

The current **SWAdmin.AWorkQs** property returns a list of *all* administrative work queues if the user accessing the property is an “Admin” user (the user’s **MENUNAME** attribute = ADMIN). If the user is a non-Admin user, **AWorkQs** returns only the administrative work queues the user is authorized to supervise (for the purpose of defining participation and redirection schedules). This complete list of supervised work queues is needed by Admin users so they can add supervisors to or remove supervisors from a work queue. Therefore, the functionality of **AWorkQs** will not change. There is, however, a need for Admin users to get a list that contains only the work queues they are authorized to supervise. The following new property is added to allow this:

- **SWAdmin.SupervisedAWorkQs** - This returns an **SWList** of **SWAWorkQ** objects, one for each administrative work queue the user is authorized to supervise. For non-Admin users, this new property returns the same list of work queues as the **AWorkQs** property.

To provide naming consistency, the **SWUser.AdminQNames** property is being deprecated and replaced by the following property:

- **SWUser.SupervisedAWorkQNames** - This returns an **SWList** of strings, one for each work queue the user is authorized to supervise.

The **AdminQNames** property still exists, for backward compatibility, but will be hidden from the interface.

Sub-Procedure Precedence Added to SPO (13783)

There is a need to be able to specify the order sub-procedure versions (released, unreleased, or model) are looked for first, second, or third when a sub-procedure is started from the main procedure.

A new *SubProcPrecedence* parameter has been added to the **StartCaseEx** method to provide this functionality. This parameter specifies the precedence of sub-procedure versions that are launched from the main procedure. The default is to start only released procedures. (This new parameter is also added to the **StartCase** method, which is a deprecated method. This provides backward compatibility.)

The *SubProcPrecedence* parameter accepts an integer value that is defined in the new constant enumeration, **SWSUBPROCPRECEDENCE**. This new constant enumeration defines the following values:

Constant	Description	Value
swPrecedenceR	Released only	0
swPrecedenceUR	Unreleased > Released	1
swPrecedenceMR	Model > Released	2
swPrecedenceUMR	Unreleased > Model > Released	3
swPrecedenceMUR	Model > Unreleased > Released	4

For example, specifying **swPrecedenceUR** causes the engine to first look for an unreleased version of the sub-procedure, then a released version. If neither is found, the error message “Sub-case started of a procedure that isn’t a sub-procedure” is written to the **sw_warn** file.

Ability to Identify all Outstanding Steps Added (13678)

The ability to identify outstanding steps for each type of step in a case family, as well as obtain the path to those outstanding steps, has been added to SPO. This ability is needed by certain functions. For instance, the **JumpTo** method requires that you provide a list of the outstanding steps you would like to withdraw. Other methods, such as **SetState** and **TriggerEvent**, require that you provide a complete path to outstanding sub-procedures if you are updating case data for fields that are defined in those sub-procedures.

You can now determine which normal, event, EAI, sub-procedure call, dynamic sub-procedure call, and graft steps are currently outstanding in the case. You can also determine the complete path, from the main case, to each outstanding step or sub-procedure. To provide this ability, the following additions and changes have been made to the TIBCO SPO COM Client.

The **SWCase** object has been updated as follows:

- **DynamicSubProcSteps** - This new property returns an **SWList** of **SWDynamicSubProcStep** objects, one for each dynamic sub-procedure call step that is currently outstanding in the case family.
- **EAISteps** - This new property returns an **SWList** of **SWEAIStep** objects, one for each EAI step that is currently outstanding in the case family.
- **EventSteps** - This new property returns an **SWList** of **SWEventStep** objects, one for each event step that is currently outstanding in the case family.
- **OutstandingItems** - This property has been modified to now return only outstanding normal steps, i.e., it returns an **SWList** of **SWOutstandingItem** objects, one for each normal step that is currently outstanding in the case family.
- **IsRecurseGrafts** - This property has been deprecated. The **IsRecurseProcPath** property (see below) should now be used for all step types. This property will continue to be supported to provide backward compatibility, but will be hidden in the COM interface.
- **IsRecurseProcPath** - This new property specifies whether or not the "outstanding step" properties (**OutstandingItems**, **SubProcSteps**, **DynamicSubProcSteps**, **EAISteps**, **EventSteps**, and **GraftSteps**) return a recursive list of outstanding steps, i.e., whether the list also includes outstanding steps from sub-procedures that have been launched from the main case.

The following new objects have been added:

- **SWDynamicSubProcStep** - This new object represents an outstanding dynamic sub-procedure call step. A list of these objects is returned by the **DynamicSubProcSteps** property on **SWCase**. This object contains the following properties:
 - **Arrived** - The date and time the step became outstanding.
 - **CaseNumber** - Identifies the case containing the outstanding step.
 - **ClassId** - Identifies the object class.
 - **Deadline** - The date and time of a deadline on the step, if set.
 - **Key** - The key for the step: ProcName | StepName | CaseNumber
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
 - **ProcName** - The name of the procedure with which the step is associated.
 - **ProcPath** - Provides the complete path to the outstanding dynamic sub-procedure call step. This can be used in the *WithdrawList* parameter in the **JumpTo** method.
 - **StepName** - The name of the dynamic sub-procedure call step.
 - **SubProcSteps** - Returns an **SWLocList** of **SWSubProcStep** objects, one for each sub-procedure that was started by the dynamic sub-procedure call step. This returns *all* sub-procedures that were started by the dynamic sub-procedure call step, whether they have completed or not.

You can determine which of the sub-procedures are still outstanding (have not completed) by accessing the **IsOutstanding** property on the **SWSubProcStep** object that represents the sub-procedure you are interested in.

- **SWEAISStep** - This new object represents an outstanding EAI step. A list of these objects is returned by the **EAISteps** property on **SWCase**. This object contains the following properties:
 - **Arrived** - The date and time the step became outstanding.
 - **CaseNumber** - Identifies the case containing the outstanding step.
 - **ClassId** - Identifies the object class.
 - **Deadline** - The date and time of a deadline on the step, if set.
 - **ExternalId** - A string that uniquely identifies an outstanding EAI step (also called an "external work item"). This is the unique identifier that is passed to the third-party application when an external work item is passed to the application. This allows the third-party application to identify the external work item when it passes it back after processing.
 - **Key** - The key for the step: ProcName | StepName | CaseNumber
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
 - **ProcName** - The name of the procedure with which the step is associated.
 - **ProcPath** - Provides the complete path to the outstanding EAI step. This can be used in the *WithdrawList* parameter in the **JumpTo** method.
 - **StepName** - The name of the EAI step.

- **SWEventStep** - This new object represents an outstanding Event step. A list of these objects is returned by the **EventSteps** property on **SWCase**. This object contains the following properties:
 - **Arrived** - The date and time the step became outstanding.
 - **CaseNumber** - Identifies the case containing the outstanding step.
 - **ClassId** - Identifies the object class.
 - **Deadline** - The date and time of a deadline on the step, if set.
 - **Key** - The key for the step: ProcName | StepName | CaseNumber
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
 - **ProcName** - The name of the procedure with which the step is associated.
 - **ProcPath** - Provides the complete path to the outstanding Event step. This can be used in the *WithdrawList* parameter in the **JumpTo** method.
 - **StepName** - The name of the Event step.
- **SWExtProcess** - This new object represents an outstanding external process that was started by a graft step. A list of these objects is returned by the **ExtProcesses** property on **SWGraftStep**. This object contains the following properties:
 - **ClassId** - Identifies the object class.
 - **IsOutstanding** - Identifies whether or not the external process has completed. This will return True until the application informs the engine that the external process has completed by calling **GraftExtProcessComp**, at which point this flag is set to False.
 - **Key** - The key for the external process object: ExtProcessName
 - **Name** - The name of the external process.
 - **ReturnStatus** - A status that is specified by the user when the external process is initiated with the **StartGraftTask** method, then again when the external process is flagged as complete with the **GraftExtProcessComp** method.
 - **StartIndex** - A zero-based index that indicates the sequential order in which the engine started the external process that is represented by the **SWExtProcess** object.

The following new properties were added to the **SWOutstandingItem** object:

- **Arrived** - The date and time the normal step became outstanding, i.e., the date and time the work item arrived in the work queue.
- **Deadline** - The date and time of a deadline on the step, if set.
- **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
- **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.

The following changes were made to the **SWSubProcStep** object:

- **Arrived** - This new property returns the date and time the sub-procedure call step became outstanding.
- **Deadline** - This new property returns the date and time of a deadline on the step, if set.
- **IsOutstanding** - This new property functions as follows: If the **SWSubProcStep** represents a sub-procedure call step, this property will always return True. If the **SWSubProcStep** object represents a sub-procedure that was started by either a dynamic sub-procedure call step or graft step, this will return True if the sub-procedure is still outstanding (has not completed yet). (The engine will keep track of when sub-procedures complete, and will set this property accordingly.)
- **Key** - This property has been modified because it was not sufficiently unique to identify a sub-procedure that was launched from a dynamic sub-procedure call step or graft step. It has been expanded to include the case number of the sub-case started by the dynamic sub-procedure call step or graft step, as follows:
`StepName | SubCaseNumber`
- **ReturnStatus** - This new property returns a system-generated status that indicates the current status for the sub-procedure that is represented by the **SWSubProcStep** object. These are enumerated in **SWSubProcStatusType**.
- **StartIndex** - This new property returns a zero-based index that indicates the sequential order in which the engine started the sub-procedure that is represented by the **SWSubProcStep** object. This index is only applicable to sub-procedures that are started by dynamic sub-procedure call steps and graft steps. If the **SWSubProcStep** object represents a sub-procedure call step, this property returns -1.
- **SubProcMajorVer** - This new property returns the *MajorVersion#* portion of the sub-procedure's version number.

- **SubProcMinorVer** - This new property returns the *MinorVersion#* portion of the sub-procedure's version number.
- **SubProcPath** - This is a new property. If the **SWSubProcStep** object represents an outstanding sub-procedure call step, this property returns the path to the sub-procedure call step. If the **SWSubProcStep** object represents a sub-procedure that was started by a dynamic sub-procedure call step or graft step, this property returns the path to the sub-procedure itself.

The following changes have been made to the **SWGraftStep** object:

- **Arrived** - This new property returns the date and time the graft step became outstanding. Graft steps are considered outstanding when they are initiated by calling the **StartGraftTask** or **SetGraftTaskCnt** methods, or when the process flow reaches the graft step.
- **ExtProcessNames** - This property has been deprecated — the **ExtProcesses** property (see below) should be used instead. This property will continue to be supported to provide backward compatibility, but will be hidden in the COM interface.
- **ExtProcesses** - This new property returns an **SWLocList** of **SWExtProcess** objects, one for each external process that has been initiated by the graft step. This returns *all* external processes initiated by the step, whether they have completed or not. You can determine which of the external processes are still outstanding (have not completed) by accessing the **IsOutstanding** property on the **SWExtProcess** object that represents the external process you are interested in.
- **ProcMajorVer** - This new property returns the *MajorVersion#* portion of the procedure's version number.
- **ProcMinorVer** - This new property returns the *MinorVersion#* portion of the procedure's version number.
- **SubProcSteps** - Returns an **SWLocList** of **SWSubProcStep** objects, one for each sub-procedure that was started by the graft step. This returns *all* sub-procedures that were started by the graft step, whether they have completed or not. You can determine which of the sub-procedures are still outstanding (have not completed) by accessing the **IsOutstanding** property on the **SWSubProcStep** object that represents the sub-procedure you are interested in.

New Method Returns Number of Cases that Match Filter Expression (13376)

The **GetFilteredCaseCnt** method has been added to the **SWProc** object. This method returns the number of cases in the procedure that match the specified filter expression. This allows you to determine the case count prior to actually requesting the case objects from the server.

Version 10.0(0.0)

New Server Log Categories Added (12608)

The following new log categories have been added to the **SWSrvLogCategoryType** enumeration.

Constant	Description	Hex Value
swCatSALTiming	SAL Timing	0x04000000
swCatDirector	Director Operations	0x08000000

Array Fields, Ad-Hoc Processing and Procedure Version Control Added (12590)

The following new functionality has been added to the TIBCO SPO COM Client:

- **Array Fields** - Array fields are defined using the SPD's Field Definition dialog in the same way as a standard single-instance field. An option on the Field Definition dialog allows you to designate the field as a single-instance field or an array field. If designated as an array field, the field can hold up to 99,999 data elements, each identified by an index (the field name followed by an index number in brackets "[]"). Array fields are used with ad-hoc processing (see below).
- **Ad-Hoc Processing** - This new functionality involves the addition of two new step types: **Dynamic Sub-Procedure Call Step** and **Graft Step**. These steps provide the ability to specify *at run-time* (rather than at procedure design-time) the number of sub-procedures and/or external processes that will be started by the step.
- **Procedure Version Control** - This new functionality provides the ability to create and track multiple versions of a TIBCO procedure.

The objects, properties, and methods that were added and/or modified in the TIBCO SPO COM Client to support this new functionality are listed below. For more detailed information about these subjects, see the *SPO Programmer's Guide*.

Array Fields

The **SWField**, **SWMarking**, **SWFMarking** objects have the following new property to indicate whether the field is a single-instance field or array field:

- **IsArrayField** - If set to True, the field/marking is an "array field", which has up to 99,999 elements accessible by index. See the *SPO Programmer's Guide* for information about how array fields are used in ad-hoc processing.

Ad-Hoc Processing

The following objects have been created/updated to support ad-hoc processing:

- **SWStepType** - The following constants have been added to this enumeration:

Constant	Description	Value
swDynamicSubProcCall	Dynamic Sub-Procedure Call Step	'D'
swGraft	Graft Step	'G'

- **SWGraftStep** - This new object represents an outstanding or initiated graft step in a live case. It contains the following properties:
 - **CaseNumber** - Identifies the case to which the graft step belongs.
 - **ClassId** - Identifies the object class.
 - **Deadline** - The date and time of deadline, if set.
 - **ExtProcessNames** - The names of the external processes that are outstanding on this graft step. Names are added to this list when external processes are started with the StartGraftTask method. Names are removed from this list when the GraftExtProcessComp and DeleteGraftTask methods are called.
 - **GraftId** - A unique ID that identifies the graft step.
 - **IsGraftOutstanding** - True if the graft step is currently the outstanding step.
 - **IsGraftWithdrawn** - True if the graft step has been withdrawn.
 - **IsTaskCntSet** - True if the task count has been set with the SetGraftTaskCnt method on **SWCase**.
 - **Key** - The key for the graft step: ProcName | StepName | CaseNumber
 - **ProcName** - The name of the procedure containing the graft step.
 - **ProcPath** - The name of the graft step. If the graft step is in a sub-procedure, this provides the path to the step in the form: "sub1 | stepname", where "sub1" is the name of the step that calls the sub-procedure in which the graft step resides.
 - **StepName** - The name of the graft step.
 - **TaskCnt** - The number of tasks that must be completed for the graft step to be released. This is set with the SetGraftTaskCnt method on **SWCase**.

- **SWCase** - The following properties and methods have been added to this object:
 - **GraftSteps** - Returns a list of **SWGraftStep** objects, one for each outstanding or initiated graft step in the case.
 - **IsRecurseGrafts** - If set to True, the list returned by **GraftSteps** will include recursive graft steps.
 - **DeleteGraftTask** - Decrements the task count by one.
 - **GraftExtProcessComp** - Informs the process engine that an external process that was started with the **StartGraftTask** method has completed.
 - **SetGraftTaskCnt** - Increments the task count to inform the process engine how many tasks must be completed before releasing the graft step.
 - **StartGraftTask** - Initiates a task by specifying the sub-procedures and/or external processes to start.
- **SWStep** - The following properties and methods have been added to this object:
 - **IsHaltOnSubProc** - If set to True, process flow is halted if the step attempts to start a non-existent sub-procedure. (This flag is set in the step definition in the SPD.)
 - **IsHaltOnTemplate** - If set to True, process flow is halted if different parameter templates are used when starting multiple sub-procedures. (This flag is set in the step definition in the SPD.)
 - **IsHaltOnTemplateVer** - If set to True, process flow is halted if different versions of parameter templates are used when starting multiple sub-procedures. (This flag is set in the step definition in the SPD.)
 - **IsKeepOnWithdraw** - If set to True, the work item is not deleted from the work queue on withdrawal (i.e., it is "kept" in the work queue). (This flag is set in the step definition in the SPD.)
 - **SubProcStatus** - Returns a numeric array field that contains return statuses for each of the sub-procedures started from the dynamic sub-procedure call step or graft step. The return statuses are defined in the new **SWSubProcStatusType** enumeration.
- **SWWorkItem** - The following property has been added to this object:
 - **IsKeepOnWithdraw** - If set to True, the work item is not deleted from the work queue on withdrawal (i.e., it is "kept" in the work queue). (This flag is set in the step definition in the SPD.)

- **SWSubProcStatusType** - This new enumeration, which identifies the sub-procedure return statuses, contains the following constants:

Constant	Description	Value
swAttempt	Sub-case start has not been attempted.	0
swStarted	Sub-case was started successfully	1
swCompleted	Sub-case completed successfully.	2
swErrSubProc	Error starting the sub-case	-1
swErrTemplate	Error starting the sub-case because different parameter templates were used.	-2
swErrInTemplateVer	Error starting sub-case because different versions of parameter template were used.	-3
swErrOutTemplateVer	Error completing sub-case because different versions of parameter template were used.	-4

- **SWAuditActionType** - The following constants have been added to this enumeration to identify the actions performed on dynamic sub-procedure call steps and graft steps:

Constant	Description	Value
swDynaGraftCaseStart	Case started for sub-case of dynamic or graft step.	25
swTaskCountSet	Task count received.	26
swTaskDeleted	Graft task deleted.	27
swSubCaseGrafted	Sub-case grafted.	28
swExtProcessGrafted	External process grafted.	29
swGraftInitiated	Graft task initiated.	30
swExtProcessReleased	External process released.	31
swGraftReleased	Graft step released.	32

Constant	Description	Value
swDynamicReleased	Dynamic sub-procedure call step released.	33
swGraftWithdrawn	Graft step withdrawn	35
swDynaGraftDeadlineExp	Deadline Expired for Dynamic or Graft Step	36
swDynamicWithdrawn	Dynamic sub-procedure	37
swKeepOnWithdraw	Step withdrawn and kept in work queue	38
swErrBadSubProc	Invalid sub-procedure error	84
swErrDiffTemplate	Different parameter template error	85
swErrDiffTemplateVer	Different parameter template version error	86

- **SWServerErrorType** - The following constants have been added to this enumeration:

Constant	Description	Hex Value
swInvalidGraftIdErr	Invalid Graft Id.	0x00DC
swExtProcessNameErr	The external process name exceeds 30 characters.	0x00DD
swProcCaseErr	The SAL returned ER_PROC. There is a problem with the procedure or the case may be invalid.	0x00DE

Procedure Version Control

The following objects have been created/updated to support procedure version control:

- **SWNode** - This object has the following new/modified properties and methods:
 - **ProcGroups** - New property. Returns a list of SWProcGroup objects (see the next bullet item), one for each procedure defined on the node.
 - **MakeProcByStatus** - New method. Allows you to make an **SWProc** object that has a specified status: **swReleased**, **swUnreleased**, or **swModel**.
 - **MakeProc** - This method has been modified to allow you to specify a version number in the parameters so that an **SWProc** object can be created for a specific procedure version.
 - **MakeStep** - This method has been modified to allow you to specify a version number in the parameters so that an **SWStep** object can be created for a specific procedure version.
- **SWProcGroup** - This new object provides access to all versions of a particular procedure. It has the following properties:
 - **ClassId** - Identifies the object class.
 - **HostingNode** - The name of the node that hosts this procedure.
 - **Key** - The key used to access a specific **SWProcGroup** object:
HostingNode | Name
 - **Name** - The name of the procedure.
 - **ProcVersions** - This returns a list of **SWProc** objects, one for each version of the procedure defined on the node.
- **SWProcStatusType** - This enumeration has the following new constant.

Constant	Description	Value
swModel	Procedure that has been imported	'M'

- **SWProc** - This object has the following new/modified properties:
 - **DateCreated** - New property. Date and time this version of the procedure was created.
 - **DateModified** - New property. Date and time the procedure was last modified.
 - **DateReleased** - New property. Date and time the procedure was released.
 - **DateWithdrawn** - New property. Date and time the procedure was withdrawn.
 - **LastUpdateUser** - New property. Name of the user who last updated the procedure.
 - **ProcAudits** - New property. This returns a list of **SWProcAudit** objects (see the next bullet item), one for each modification that has been made to the procedure.
 - **ProcMajorVer** - New property. The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - New property. The *MinorVersion#* portion of the procedure's version number.
 - **VersionComment** - New property. Comment that was entered when this version of the procedure was created.
 - **IsWithAuditData** - New property. If set to True, procedure audit data will be returned from the server with the procedure. The audit data is available with the **ProcAudits** property.
 - **Tag** - The tag that is returned by this property has been modified to include the major and minor portions of the procedure's version number so that it can be used with the **MakeProcByTag** method to make a specific version of the procedure. (Only applicable if using a TIBCO SPO Server that supports procedure version control.)
 - **Key** - The value returned by this property is extended to include the major and minor portions of the procedure version number. This new key format must be used when invoking the **ItemByKey** method to extract an **SWProc** object from the **SWNode.Procs**, **SWUser.AuditProcs**, **SWUser.StartProcs** and **SWProcGroup.ProcVersions** lists. (Only applicable if using a TIBCO SPO Server that supports procedure version control.)



SPO applications that have procedure keys “hard coded” must be modified to include the procedure version component (ProcMajorVer|ProcMinorVer) if both the client and TIBCO SPO Server support procedure version control. See [Procedure Key must include Version Number, page 36](#) for more information.

- **SWProcAudit** - This new object represents a modification to a procedure. It contains the following properties:
 - **Action** - The action that was performed on the procedure. These are defined in the new **SWProcAuditActionType** enumeration.
 - **ClassId** - Identifies the object class.
 - **Comment** - Describes the modification made to the procedure.
 - **Date** - Date and time the modification was made.
 - **Key** - The key to the **SWProcAudit** object. The key is an integer that is an index into the list of **SWProcAudit** objects returned by **ProcAudits**.
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
 - **User** - The user who made the modification to the procedure.
- **SWAuditActionType** - The following constant has been added to this enumeration to identify cases that are migrated to a new procedure version:

Constant	Description	Value
swCaseMigrated	Case migrated to new procedure version.	34

- **SWProcAuditActionType** - This new enumeration describes actions performed when modifying a procedure definition. It contains the following constants:

Constant	Description	Value
swProcCreated	Procedure created	'C'
swProcComment	Procedure comment modified	'M'
swProcImported	Procedure imported	'I'
swProcUpdated	Procedure updated	'U'
swProcReleased	Procedure released	'R'
swProcWithdrawn	Procedure withdrawn	'W'

- **SWCase** - This object has the following new properties, which allow you to determine the version of procedure the case is from.
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
- **SWAuditStep** - This object has the following new properties, which allow you to determine the version of procedure when the action in the audit step took place.
 - **ProcMajorVer** - The *MajorVersion#* portion of the procedure's version number.
 - **ProcMinorVer** - The *MinorVersion#* portion of the procedure's version number.
- **SWStep** - The following method has been modified on this object:
 - **Tag** - The tag that is returned by this property has been modified to include the major and minor portions of the procedure's version number so that it can be used with the **MakeStepByTag** method to make a step from a specific version of the procedure. (Only applicable if using a TIBCO SPO Server that supports procedure version control.)
- **SWClientErrorType** - The following constant has been added to this enumeration:

Constant	Dec.	Hex.	Description
swInvalidKeyErr	2077	0x81D	"Invalid Key."

Property Added to get Script Contents (12446)

The **Script** property has been added to the **SWStep** object. This new property provides access to the script that is defined in a step of type **swScript**.

Property Added to Discard Local Blocks of Items (12427)

The **IsKeepLocalItems** property has been added to **SWXList** to allow you to control whether or not more than one block of items will be held locally after they have been retrieved from the TIBCO SPO Server. If this flag is set to True, multiple blocks can be held locally. If set to False (the default), when another block is sent from the server, the previous block is automatically cleared, thereby minimizing the use of local memory.

Property Added to get Audit Trail Message (12411)

The **Message** property has been added to the **SWAuditStep** object. This new property provides access to the actual message that is written to the audit trail for that particular audit step.

New Counts for Work Items on an SWXList (12410)

The following properties have been added to the **SWCriteriaWI** object to provide counts for work items on **SWXLists**:

- **DeadlineCnt** - Returns the number of items with deadlines on the XList.
- **UnopenedCnt** - Returns the number of new (i.e., unopened) work items on the XList.
- **UrgentCnt** - Returns number of urgent work items on the work queue or in the XList.

Changes in Functionality

This section lists changes in functionality in each release of this product.

Version 10.0(4.2)

SWNode and SWNodeInfo Keys Modified (15469)

All **SWNode** and **SWNodeInfo** keys will now take the form:

```
ComputerName | NodeName | IsDirector | InstanceNumber
```

where:

- *IsDirector* indicates if the node is a TIBCO SPO Director ("Y") or a TIBCO SPO Server ("N").
- *InstanceNumber* indicates the instance number of the TIBCO SPO Server. This will default to "1" for non-multiple instance servers.

The **ItemByKey**, **Login**, **Logout**, **Disconnect**, **MakeViewItems**, and **MakeViewCases** methods will automatically add the default values for the *IsDirector* ("N") and *InstanceNumber* ("1") components if they are not provided in the *NodeKey* parameters. This provides backward compatibility for applications that use hard-coded keys.

The **AddNode** and **MakeNodeInfo** methods will automatically add the default *IsDirector* ("N") component to the *SWNodeInfo* key if it is not provided in the method call. They also always add the default *InstanceNumber* ("1") component to the *SWNodeInfo* key. These methods always assume instance 1.

The **AddNodeEx** and **MakeNodeInfoEx** methods will automatically add the *IsDirector* ("N") and *InstanceNumber* ("1") components to the *SWNodeInfo* key if they are not provided in the parameters passed to the method calls.

Version 10.0(2.0)

Client Log Default Categories Changed (14111)

If the TIBCO SPO COM Client Log debug log level (**swLogDebug**) was selected, all log categories were written to the client log by default, including object constructor and destructor information. This could result in a very large log file in a short period of time. Because of this, the default log categories have been changed for the debug log level to include all categories except **swCatConstDestr** (object constructors and destructors).

If you need object constructor/destructor information in the log, you can specify **swCatAll** in the **Categories** property on **SWLog**, or use the **EnableCategory** method to enable the **swCatConstDestr** category.

Compatibility

This section provides compatibility information for the TIBCO SPO COM Client.

TIBCO SPO Server

This version of the TIBCO SPO COM Client is backward-compatible with older versions of the TIBCO SPO Server. However, if you are not using the most recent version of the TIBCO SPO Server, some of the features listed in the [New Features](#) section may not be available to you.

TIBCO Staffware Server/Engine

The “type” of TIBCO Staffware Process/iProcess Engine you are using also determines whether or not new functionality described in the [New Features](#) section is available to you. The two types of engines are:

- **TIBCO Staffware iProcess Engine** - This is also referred to as the “TIBCO Staffware iPE Server.” This type of engine is required for some of the new functionality (e.g., EAI steps, immediate case number availability, etc.) that is described in the [New Features](#) section. If you are using a TIBCO Staffware iProcess Engine, you will also be using a TIBCO SPO Server that supports the functionality provided by the TIBCO Staffware iProcess Engine.
- **TIBCO Staffware Process Engine** - This was previously known as the “Staffware Server.” If you are using this type of engine, some of the new functionality that is described in the [New Features](#) section (e.g., EAI steps, immediate case number availability, etc.) is not available to you. If you are using a TIBCO Staffware Process Engine, you will also be using a TIBCO SPO Server that supports the functionality provided by the TIBCO Staffware Process Engine.

The TIBCO SPO COM Client will work with both “types” of TIBCO SPO Servers and Process/iProcess Engines described above.

Engine and Server Version Numbers

As we are transitioning from “Staffware” to “TIBCO,” the version numbers of the engines and servers are changing as well. Staffware version numbers included major, minor, maintenance release, and patch numbers, with parentheses (e.g., 10.2(0.0)). TIBCO version numbers include major, minor, and maintenance release numbers, without parentheses (e.g., 10.2.0). Hotfix numbers (the equivalent to a “patch”) are not shown in the product version number.

A Staffware version number may also be preceded by an "i" (e.g., i10.0(0.0)), indicating that it is an "iProcess" Engine or an SPO Server that supports the functionality offered by iProcess Engines.

Moving forward from version 10.2.0, all *new releases* of engines, SPO Servers, and SPO Clients will use the 3-digit TIBCO version numbering system. The version number will also not include an "i" to indicate that it is an iProcess Engine or an SPO Server that supports the functionality of an iProcess Engine; by default, all engines from 10.2.0 forward are iProcess Engines, and all SPO Servers from 10.2.0 forward support the functionality of iProcess Engines.

You can determine whether you are using a TIBCO Staffware Process Engine or a TIBCO Staffware iProcess Engine by looking at the version number. The version number can be found in the first line of the `$WDIR\swdefs` (Windows) or `/$WDIR/swdefs` (UNIX) file.

The following summarizes version numbers for engines and servers:

- **TIBCO Staffware iProcess Engine** - Prior to version 10.2.0, these engines had an "i" in the version number, e.g., i10.0-o(5.3). From version 10.2.0 forward, the version number will be 3 digits, with no "i". (The database supported will be indicated in parentheses following the version number, e.g., 10.2.0 (Oracle).)
- **TIBCO Staffware Process Engine** - These will continue to use the Staffware numbering system. Their version number begins with 8 or 9, with no "i", e.g., 9.0-x(0.7).
- **TIBCO SPO Server** - Prior to version 10.2.0, these servers had version numbers either with or without an "i". SPO Servers without the "i" (e.g., 9.3(5.0)) are used with Process Engines; SPO Servers with an "i" (e.g., i10.0(4.0)) are used with iProcess Engines. From version 10.2.0 forward, the version number will be 3 digits, with no "i"; these will be used with iProcess Engines.

Development Environment

This version of the TIBCO SPO COM Client was developed in the following environment:

- Built on Microsoft NT Server 4.0 SP6a
- Tested on Microsoft Windows 2000 SP2, NT 4.0 SP6a, and Windows XP
- Compiled with Microsoft Visual C/C++ Version 6 SP5

Other Information

Must Pass a TextBox Control as a Function Parameter in a VB Application

The COM interface defines interface property ID 0 as a default property. If an application passes SPO a pointer to an **IDispatch** interface without specifying an interface property, SPO asks the COM interface to supply property 0.

In Visual Basic, this means that you can pass the contents of a control's main property without knowing the property name - you simply specify the control's name. For example, if you have a control named **Control1** with a property called **Text**, you can pass the contents of **Control1.Text** to a function by typing *FunctionName(Control1)*, instead of *FunctionName(Control1.Text)*. This can be especially useful for controls which have **Value** or **Data** as their main property.

However, you cannot use this method with the Microsoft **TextBox** control. You can access the default property from a **TextBox IDispatch** pointer in the Integrated Development Environment, but when you compile and run the program the attempt returns the error **E_NOTIMPL** (-2147467263, 80004001 in hex) to the **DispInvoke** method.



The error indicates a parameter error - that no default property (ID=0) is defined. It does not mean that the SPO method isn't defined.

The reason for this error is unknown, but because of it, you *must* specify the **Text** property name when passing the Microsoft **TextBox** control as a function parameter.

Full Kanji Character Set Not Supported

Because the TIBCO Staffware iProcess Engine uses the Shift-JIS character set, the full range of Kanji characters is not supported by the SPO COM object interface. Unsupported Kanji characters are translated either to a period character, or to a different Kanji character.

The TIBCO Staffware iProcess Engine uses the Shift-JIS character set in four places that can be accessed by the SPO COM object interface. They are:

- Field Values (Marking Values) (read/write)
- Attribute Values (read/write)
- Case Descriptions (read/write)
- Step Descriptions (read only)

For more information about this issue, refer to the article *Q170559: PRB: Conversion Problem Between Shift-JIS and Unicode*, which is available from Microsoft's web site at:

<http://support.microsoft.com/support/kb/articles/q170/5/59.asp>

License Count may be a Negative Number

If the number of user licenses exceeds 32767, the **LogonLicenseCnt** and **LogonsAvailableCnt** properties on the **SWNodeInfoEx** object incorrectly return negative values. The workaround for this restriction is to add 65536 to the negative value to obtain the correct license count value.

Note that the **LogonLicenseCnt** and **LogonsAvailableCnt** properties are not applicable if your TIBCO Staffware iProcess Engine includes CR 16592, which eliminates the requirement for user licenses. These methods always return -1 if your engine includes CR 16592.

Procedure Key must include Version Number

When procedure version control was added to the TIBCO SPO COM Client in version 10.0(0.0), the procedure key was extended to include the version number component, as follows:

```
SWProc.Key = HostingNode|Name|ProcMajorVer|ProcMinorVer
```

If both your TIBCO SPO COM Client and TIBCO SPO Server support procedure version control (message interface version 3.0.0 or newer), this new key format must be used when invoking the **ItemByKey** method to extract an **SWProc** object from the lists returned by these properties: **SWNode.Procs**, **SWUser.AuditProcs**, **SWUser.StartProcs**, and **SWProcGroups.ProcVersions**.

SPO applications that have procedure keys "hard coded" must be modified to include the procedure version component (*ProcMajorVer* | *ProcMinorVer*) if both the client and TIBCO SPO Server support procedure version control. If the **ItemByKey** method is invoked without the version component in the key, an **swInvalidKeyErr** error is thrown.

If you include the procedure version number component in the key, but your TIBCO SPO Server does not support procedure version control, the client will be aware that the server does not expect the version component and will not include it when the key is sent to the server.

Closed Issues

The table in this section list issues that were closed in the named releases.

Closed in Release	Change Request #	Summary
10.2.0	16907	The MakeViewItemByTag method is returning an "Unauthorized to view workitems in Work Queue" error, even though the user has the proper authorization for the work queue. Corrected.
10.2.0	16814	New constants need to be added to the SWNodeInfoStatusType enumeration to fully describe the status of a TIBCO SPO Server made available by a TIBCO SPO Director. Implemented. See , New TIBCO SPO Server Status Types (16814) , on page 2 .
10.2.0	16726	The following extraneous message can appear in the Client Log: "swrtns.GetSockRef: Delete Socket Not Found. (ConnectId = 0x9)." Corrected. This message will no longer appear.
10.2.0	16667	New error types need to be added to the SWServerErrorType enumeration. Implemented. See New Error Types Added to SWServerErrorType Enumeration (16667) , page 3 .
10.2.0	16452	When locking a previously locked work item, the LastError property on SWWorkItem returns FF7A , which is not a valid error number. Corrected.
10.2.0	16252	If connected to a TIBCO SPO Server with an interface version older than 1.3.1, the UrgentCnt , DeadlineCnt , and UnopenedCnt properties return unpredictable results (whatever is in memory). Corrected. These properties now return -1 if connected to a TIBCO SPO Server with an interface version older than 1.3.1.

Closed in Release	Change Request #	Summary
10.2.0	16208	<p>If an invalid path is specified for the client log (using either the LogDirectory property on the SWLog object or the LogDirectory Registry entry), the TIBCO SPO COM Client does not write the client log to the default directory. Instead, it inactivates the client log.</p> <p>Corrected. If an invalid path is specified, the COM client will now use the correct default path (see the COM client on-line help for default paths for each operating system). It will also write the default path to the incorrectly set LogDirectory property or Registry entry. The log will not be inactivated.</p>
10.2.0	16207	<p>Accessing the SWNodeInfoEx object when there are a large number of clients can cause a stack overflow, resulting in a crash.</p> <p>Corrected.</p>
10.2.0	16177	<p>An enhancement needs to be made to the TIBCO SPO COM Client to allow you to specify which markings will be returned from the TIBCO SPO Server when the work item is locked, regardless of whether they are marked on the form or result from conditional statements.</p> <p>Implemented. See New Methods Added to Retrieve Markings When Locking Work Items (16177), page 3.</p>
10.2.0	12550	<p>An enhancement needs to be made to the TIBCO SPO COM Client to allow you to specify which CDQP fields to return from the server with work items that reside in an SWXList.</p> <p>Implemented. See New Property Added to Specify Which CDQP Fields to Return from Server (12550), page 3.</p>
10.2.0	10642	<p>An enhancement needs to be made to the TIBCO SPO COM Client to allow you to specify which CDQP fields to return from the server when you create work items with the "Make..." methods.</p> <p>Implemented. See New Methods Added to Create Work Items and Specify CDQPs and Case Fields (10642), page 4.</p>
10.0(6.0)	16182	<p>The TIBCO SPO COM Client components need to be updated with TIBCO branding.</p> <p>Implemented. When viewed in Visual Studio's Project/References list, the reference to SPO is now shown as "TIBCO Staffware Process Objects..." rather than "Staffware Process Objects..." in the alphabetized list.</p>

Closed in Release	Change Request #	Summary
10.0(6.0)	16084	<p>The SWOutstandingItem.WorkItemTag property is returning an invalid tag. This only occurs if the client is connected to an older TIBCO SPO Server (message interface version 3.0.2 or older).</p> <p>Corrected.</p>
10.0(6.0)	16065	<p>The SWStep object needs a property to hold the extended description that can be entered when a normal-type step is defined in the SPD.</p> <p>Implemented. See Extended Description Property Added to SWStep (16065), page 5.</p>
10.0(6.0)	16045	<p>The TIBCO SPO COM Client needs to support Transaction Control Steps.</p> <p>Implemented. See Transaction Control Steps Added (16045), page 5.</p>
10.0(6.0)	15724	<p>The SWWString class is unnecessarily being assigned an empty string. The previously used BSTRs were initialized with a NULL pointer, requiring that they be assigned an empty string; the new SWWString class is initialized with an empty string. Eliminating this unnecessary assignment will improve memory allocation performance.</p> <p>Corrected.</p>
10.0(4.2)	15502	<p>The TIBCO SPO COM Client does not support arrays of fields that are n-based (it does support zero-based arrays). Attempting to pass n-based arrays of field values results in a “Value not in array limits” error.</p> <p>Corrected.</p>
10.0(4.2)	15469	<p>Methods that are passed <i>NodeKey</i> and <i>NodeInfoKey</i> parameters are not properly adding the default instance number to the keys.</p> <p>Corrected. The keys for the SWNode and SWNodeInfo objects will now always include the <i>IsDirector</i> and <i>InstanceNumber</i> values. See SWNode and SWNodeInfo Keys Modified (15469), page 31.</p>
10.0(4.1)	15439	<p>The TIBCO SPO COM Client needs to be enhanced to allow logging into multiple instances of the TIBCO SPO Server.</p> <p>Implemented. See Allow Logging Into Multiple Instances of the TIBCO SPO Server (14985 and 15439), page 8.</p> <p>The changes to the TIBCO SPO COM Client for the multiple-instance enhancement were done under both 14985 and 15439.</p>

Closed in Release	Change Request #	Summary
10.0(4.1)	15392	Attempting to enable persistence on an SWXList returned by the GetXListPredict method results in an “invalid socket” error. Corrected.
10.0(4.1)	15335	The TIBCO SPO COM Client hangs when using Microsoft TCP/IP filtering (available by selecting Advanced > Options > TCP/IP Filtering from the Internet Protocol (TCP/IP) Properties dialog). Corrected.
10.0(4.1)	15247	Enumeration constants need to be added to SWAuditActionType to describe steps that are released without an addressee or sub-procedures specified. Implemented. See Enumeration Constants Added to SWAuditActionType (15247), page 8 .
10.0(4.1)	15211	Enabling persistence on an SWXList that was returned by the SWNode.GetXList method results in an “invalid socket” error. Corrected.
10.0(4.1)	15209	The InterfaceEqual and InterfaceNewer methods are not checking for valid parameters (specifically, missing brackets around the interface version number). Also, the InterfaceEqual method is no longer supporting the use of the “*” wildcard character. Corrected.
10.0(4.1)	15145	Extraneous “item not found” messages are occurring when accessing the Fields and Markings properties. Corrected.
10.0(4.1)	15144	The following exception can occur when retrieving items in an SWList : “Unrecoverable error detected - Call Staffware”. This also results in the following message being written to the client log: “MsgBase.FindDelim: No new buffers”. The problem is a timing issue with parsing the message returned from the TIBCO SPO Server. Corrected.
10.0(4.1)	15034	Steps that have a deadline type of swPeriod return the time incorrectly. Corrected.

Closed in Release	Change Request #	Summary
10.0(4.1)	14985	<p>The TIBCO SPO COM Client needs to be enhanced to allow logging into multiple instances of the TIBCO SPO Server. Also, a means of limiting the number of cases retrieved from the server when using SWXLists needs to be added.</p> <p>Implemented. See Allow Logging Into Multiple Instances of the TIBCO SPO Server (14985 and 15439), page 8 and Ability to Limit Number of Cases Retrieved from TIBCO SPO Server Added (14985), page 9.</p> <p>The changes to the TIBCO SPO COM Client for the multiple-instance enhancement were done under both CRs 14985 and 15439.</p>
10.0(4.1)	14977	<p>Setting the FilterExpression property for an SWXList to an invalid expression or too large of an expression, then rebuilding the SWXList results in the expected “data too big or truncated” error. However, accessing properties on the SWXList after this error causes an assert in <code>MsgBase:GetStrFakeIt</code> and a subsequent access violation crash of the client.</p> <p>Corrected.</p>
10.0(3.0)	14870	<p>The COM Client crashes if the GetFilteredCaseCnt method on SWProc is passed an empty argument.</p> <p>Corrected. Passing an empty argument now causes this method to throw an “Invalid Parameter” error.</p>
10.0(3.0)	14784	<p>Calling the MakeWorkItemByTag method in a multi-node environment returns a “Queue not found” error.</p> <p>Corrected.</p>
10.0(3.0)	14770	<p>The ability to get the date and time a case was started needs to be added.</p> <p>Implemented. See Properties Added to Obtain Case Start Date and Time (14770), page 10.</p>
10.0(3.0)	14769	<p>Visual Basic crashes with the automation error, “error in oleaut32.dll”, if the FieldNames and FieldValues arguments are passed to any of the following methods: SetState, JumpTo, TriggerEvent, StartCaseEx, SetCaseData, SimulateCase.</p> <p>Corrected.</p>
10.0(3.0)	14754	<p>Exceptions sometimes occur during the parsing of external forms.</p> <p>Corrected.</p>

Closed in Release	Change Request #	Summary
10.0(3.0)	14619	<p>Currently, the TIBCO SPO Server calls the MOVESYSINFO function after every administrative change. This can tie up the background and WIS/WQS processes for long periods of time if there are lots of users. A method needs to be added so the client can explicitly call the MOVESYSINFO function.</p> <p>Corrected. The MoveSysInfo method has been added to perform an explicit MOVESYSINFO. See Method Added to Perform MOVESYSINFO Function (14619), page 10.</p>
10.0(3.0)	14611	<p>There are some error code enumeration mismatches.</p> <p>Corrected. The IDL was updated to make error enumerations consistent between the TIBCO SPO COM Clients.</p>
10.0(2.0)	14585	<p>The SetDefCriteria method on SWWorkQ sets the default filter and sort criteria based on the current criteria settings in the SWView in the WorkItems property. This is not practical when using SWXLists.</p> <p>Corrected. A new method has been added to SWWorkQ that allows you to pass in the filter and sort criteria to be used as the default criteria. See New Method Added to Set Default Criteria (14585), page 11.</p>
10.0(2.0)	14467	<p>Some error messages generated by the TIBCO SPO COM Client and Server do not match the messages shown in the on-line help or the messages that appear in the object browser. There are also messages that include "SEO" instead of "SPO."</p> <p>Corrected. Some messages in the SWClientErrorType and SWServerErrorType enumeration have been modified so that there is consistency between the on-line help, the object browser, and the actual message that is generated.</p>
10.0(2.0)	14459	<p>External form data for EAI steps is being returned from the server when procedure definitions are retrieved. This data should be returned only when accessing the SWStep.ExtForm property.</p> <p>Corrected. External form data on EAI steps is now retrieved from the server only when the ExtForm property is accessed.</p>
10.0(2.0)	14446	<p>Enumerations need to be added for errors that occur with case prediction.</p> <p>Corrected. New enumerations have been added to SWServerErrorType. See New SWServerErrorType Enumerations for Prediction (14446), page 11.</p>

Closed in Release	Change Request #	Summary
10.0(2.0)	14436	A value of "00" is always returned for the seconds portion of the Date returned by TimeStarted and TimeEnded on SWPredictedItem . Corrected.
10.0(2.0)	14382	Procedure audit data is being returned from the server when the IsWithAuditData flag on SWProc is set to False. Corrected.
10.0(2.0)	14363	The SWAdmin.MakeAWorkQ method fails if it is called without specifying a node name. If both the node name and the work queue name are specified, but there are a large number of work queues, the method exits without returning an error. Corrected.
10.0(2.0)	14321	If the TIBCO SPO Server generates an error when the IsPersisted flag on SWCriteriaWI is set, the TIBCO SPO COM Client is ignoring the error. Corrected. The TIBCO SPO COM Client will now throw an error returned by the TIBCO SPO Server when setting the IsPersisted flag.
10.0(2.0)	14223	A large number of audit trail records can cause the client to abend with a stack overflow (the exact number of records that will cause a stack overflow is unknown, but it appears to be platform dependent). Corrected. Temporary memory allocated to the stack upon creation of audit trail objects is now deallocated after the object creation, preventing the stack size from increasing.
10.0(2.0)	14176	Using a filter expression greater than 4K causes the TIBCO SPO COM Client to crash. This is caused by memory being overwritten when the filter expression is written to the client log (only occurs if the log level is set to swLogDebug). Corrected. Log entries greater than 4K are now truncated at 4K. It will no longer cause the client to crash.
10.0(2.0)	14165	Accessing a list of AuditSteps that contains more than 32K entries causes the application to abend. Corrected. The AuditSteps list can now have up to 64K entries.

Closed in Release	Change Request #	Summary
10.0(2.0)	14158	<p>If the definition of an external form for an EAI step (available in the SWStep.ExtForm property) exceeds the value set for the TIBCO SPO Server response buffer, the TIBCO SPO Server crashes. (The response buffer size is set with the TIBCO SPO Server configuration parameter, TCPResponsePages — it defaults to 2 pages (4K).)</p> <p>Corrected. The TIBCO SPO COM Client was modified to now send large external form definitions in multiple messages. A new SWServerErrorType enumeration was also added, which is returned by the TIBCO SPO Server if an attempt is made to write a value to the response buffer that is larger than the size specified by the TCPResponsePages parameter. See New SWServerErrorType Enumeration for Buffer Overflow (14158), page 13.</p>
10.0(2.0)	14132	<p>The TIBCO SPO COM Client product version is not being written to the client log. The word “UNKNOWN” is being written instead of the product version. This is only occurring on Windows clients.</p> <p>Corrected. The product version is now properly written to the client log.</p>
10.0(2.0)	14111	<p>Currently, object constructor and destructor information is written to the client log by default, which can result in a very large log file in a short period of time. The default client log categories need to be changed so that all object constructor and destructor information is not written to the log by default.</p> <p>Corrected. Now when the swLogDebug level is used, object constructor and destructor information is not written to the log by default. See Client Log Default Categories Changed (14111), page 32.</p>
10.0(2.0)	13869	<p>Setting the MessageWaitTimeout variable to a very large value (e.g., 3600000 milliseconds) causes client logins to fail and returns error 2055.</p> <p>Corrected.</p>
10.0(2.0)	13656	<p>Improvements need to be made to the TIBCO SPO COM Client to improve performance and scalability.</p> <p>Implemented.</p>
10.0(2.0)	13452	<p>There are bottlenecks related to dynamic memory allocation on Windows platforms.</p> <p>Corrected.</p>

Closed in Release	Change Request #	Summary
10.0(2.0)	12946	<p>Duplicate SWXLists are created on the TIBCO SPO Server, increasing memory usage at the server. This is occurring when the Rebuild method is called on the SWXList, but the reply is not immediately parsed (asynchronous communication). Then if the IsPersisted property is set before any other property on the SWCriteriaWI object is accessed, another message is sent to the server because the PersistenceID property is empty. This causes another SWXList to be created at the server.</p> <p>Corrected. Now when the Rebuild method is called, the reply is immediately parsed (synchronous communication), so that the PersistenceID property is available. Setting the IsPersisted property to inform the server to preserve the SWXList does not cause another SWXList to be created.</p>
10.0(2.0)	12520	<p>Attempting to lock a work item that was already locked with the TIBCO Staffware Process Client causes an assert to be issued.</p> <p>Corrected. This scenario will no longer throw an assert. Instead, the error "Work item already locked by someone else" is thrown.</p>
10.0(1.0)	14045	<p>Rebuilding an SWXList of predicted items can cause a Dr. Watson. This only occurs if the SWXList is the result of calling the MakeXListPredict method, but no predicted items were returned in the SWXList.</p> <p>Corrected.</p>
10.0(1.0)	13902	<p>Attempting to rebuild an SWWorkItem object for a work item that has been released results in a "SWServer Error: Unknown" error. The error message that should be returned is "Queue item not found".</p> <p>Corrected.</p>
10.0(1.0)	13836	<p>Setting IsRebuildAll to True, then rebuilding the work queue causes the COM client to crash.</p> <p>Corrected.</p>
10.0(1.0)	13821	<p>Attempting to access a work item on an SWXList causes an Access Violation error.</p> <p>Corrected.</p>

Closed in Release	Change Request #	Summary
10.0(1.0)	13814	<p>Accessing a work item on an SWXList that has been locked may result in an empty list of markings for the work item. This occurs only on subsequent accesses when the IsKeepLocalItems property is set to False.</p> <p>Corrected.</p>
10.0(1.0)	13785	<p>A new property is needed so that “Admin” users can get a list of the work queues they are authorized to supervise. Currently, the SWAdmin.AWorkQs property returns a list of all work queues when accessed by an Admin user.</p> <p>Implemented. New properties have been added to SWAdmin and SWUser to allow this functionality. See Properties Added to List Supervised Work Queues (13785), page 13 for more information.</p>
10.0(1.0)	13783	<p>Attempting to launch a sub-procedure is causing an error message in the sw_warn file. This is occurring because the TIBCO Staffware Process Engine is assuming a “precedence” of “released” for the sub-procedure, but no released version of the sub-procedure exists. SPO is not specifying a precedence for starting sub-procedures.</p> <p>Corrected. A new parameter is added to the StartCaseEx and StartCase methods to allow the user to specify precedence. A new enumeration type is also added to define each precedence that can be specified. See Sub-Procedure Precedence Added to SPO (13783), page 13 for more information.</p>
10.0(1.0)	13770	<p>The MakeProc and MakeStep methods allow you to pass in only a ProcMinorVer number in the method call, without the ProcMajorVer number. The procedure/step is created and an error is not produced. These methods should require both ProcMajorVer and ProcMinorVer number if either is provided.</p> <p>Corrected. The TIBCO SPO COM Client now throws the “SWClient Error: Invalid Parameter” error if either the <i>ProcMajorVer</i> or <i>ProcMinorVer</i> number is provided, but not the other.</p>

Closed in Release	Change Request #	Summary
10.0(1.0)	13678	<p>The SPO object model needs to be expanded to allow runtime determination of all steps that are currently outstanding in the case. Previously, only steps that resulted in a work item being placed in a work queue were identifiable as outstanding.</p> <p>Implemented. The user can now determine which normal, sub-procedure call, dynamic sub-procedure call, event, EAI, and graft steps are outstanding. See Ability to Identify all Outstanding Steps Added (13678), page 14.</p>
10.0(1.0)	13597	<p>If the SWList returned by SWStep.PublicFields is rebuilt, the list will contain incorrect information.</p> <p>Corrected.</p>
10.0(1.0)	13562	<p>Segmentation violations are occurring when running under heavy load conditions in a multi-threaded environment.</p> <p>Corrected.</p>
10.0(1.0)	13473	<p>Accessing the labels (SWLabel objects) from SWFRow causes memory usage to grow.</p> <p>Corrected.</p>
10.0(1.0)	13468	<p>The SWNode.MakeProc method will fail if it is called without providing the <i>NodeName</i> parameter. The method call is failing to determine the default nodename.</p> <p>Corrected.</p>
10.0(1.0)	13408	<p>When calling the Rebuild method on an SWXList that contains work items, the status returned by the method call does not reflect the current status of the work items.</p> <p>Corrected.</p>
10.0(1.0)	13376	<p>A method is needed to determine the number of cases in a procedure that match a specified filter expression.</p> <p>Implemented. See New Method Returns Number of Cases that Match Filter Expression (13376), page 20.</p>

Closed in Release	Change Request #	Summary
10.0(0.0)	13099	When a step is created with the MakeStepByTag method, the IsPrediction flag on the step returned maybe incorrect. The IsPrediction flag is always taking on the value to which the IsPublic flag is set. Corrected.
10.0(0.0)	13062	If the TIBCO SPO Server is heavily loaded and there is a receive timeout, an assert may occur, resulting in a client crash. Corrected.
10.0(0.0)	12911	Accessing an SWXList of cases that includes case fields causes an exception. Corrected.
10.0(0.0)	12894	Accessing an SWXList of cases causes an exception. Corrected.
10.0(0.0)	12840	Calling the LockItemsEx method on SWWorkQ causes a fatal assertion. Corrected.
10.0(0.0)	12819	Releasing a work item on an SWXList causes an error. Corrected.
10.0(0.0)	12741	If an error is returned from the server as a result of a login , an abend occurs. Corrected.
10.0(0.0)	12644	If the client application creates an SWList object, then clears it without accessing any items on the list, the items will remain in memory. Corrected.
10.0(0.0)	12608	New TIBCO SPO Server log categories need to be added to SWSrvLogCategoryType . Implemented. See New Server Log Categories Added (12608), page 21 .

Closed in Release	Change Request #	Summary
10.0(0.0)	12590	<p>The TIBCO SPO COM Client needs to support ad-hoc processing and procedure version control.</p> <p>Implemented. New objects, properties, and methods have been added to the object model to support this functionality. See Array Fields, Ad-Hoc Processing and Procedure Version Control Added (12590), page 21.</p>
10.0(0.0)	12546	<p>The ItemCount property on SWXList does not contain the correct value.</p> <p>Corrected.</p>
10.0(0.0)	12446	<p>The TIBCO SPO COM Client cannot display the contents of the script on a Script step.</p> <p>Corrected. The Script property has been added to SWStep to provide access to the contents of the script. See Property Added to get Script Contents (12446), page 29.</p>
10.0(0.0)	12427	<p>The ability to specify whether more than one block of items should be held locally after they are retrieved from the TIBCO SPO Server needs to be added. This would be used to prevent blocks of items from accumulating and consuming memory.</p> <p>Corrected. The IsKeepLocalItems property was added to SWXList. See Property Added to Discard Local Blocks of Items (12427), page 29.</p>
10.0(0.0)	12414	<p>Customer configuration information needs to be included in the TIBCO SPO COM Client Log.</p> <p>Implemented. The log will now include general system information, such as the platform, number of processors, etc.</p>
10.0(0.0)	12411	<p>The TIBCO SPO COM Client cannot display the audit trail message.</p> <p>Corrected. The Message property was added to SWAuditStep to provide access to the audit trail message. See Property Added to get Audit Trail Message (12411), page 30.</p>
10.0(0.0)	12410	<p>Additional counts need to be available for work items on an SWXList.</p> <p>Implemented. See New Counts for Work Items on an SWXList (12410), page 30.</p>

Known Issues

The table in this section lists known issues in this release.

Change Request #	Summary/Workaround
14720	<p>Summary Procedure name in predicted items includes version number.</p> <p>If you are using a TIBCO SPO Server version i10.0(2.0) or newer, the procedure version number will incorrectly be appended to the procedure name in predicted item objects (SWPredictedItem) that are the result of calling the PredictCase or SimulateCase method. The procedure name, with the appended version number, appears in the ProcName and Key properties on the SWPredictedItem object. The version number is appended to the procedure name in the form: /#.# (e.g., MyProc/1.1). The forward slash and version number will be removed from the procedure name when this defect is fixed. (Note that the version number is not appended to the procedure name if the predicted item is from a sub-procedure.)</p> <p>Workaround None.</p>
