TIBCO iProcess® Server Objects (.NET)

Release Notes

Version 11.10.0 | May 2025

Contents

New Features

The following new features have been introduced in each release of TIBCO iProcess® Server Objects (.NET).

> **Note:**
> **Earlier Versions**
>
> For details of the MRs/CRs implemented in earlier versions of the product, please see the Readme and Release Notes for that particular product release.

Version 11.10.0

No new features have been added in this release of TIBCO iProcess® Server Objects (.NET).

Version 11.9.0

No new features have been added in this release of TIBCO iProcess® Server Objects (.NET).

Version 11.4.1

New Parameter on Close and Purge Case Methods to Specify Priority (IPSO-242)

A new *aPriority* parameter has been added to the following methods:

- CloseCases
- CloseCasesBy Criteria
- PurgeCases

- PurgeCasesBy Criteria
- PurgeAndReset

This new parameter can be used to increase or decrease the priority at which cases are closed or purged.

Password Size Limit Has Been Increased (IPSO-239)

If the User Validation API (UVAPI) is used to validate users, the current size limit of 4K may not be large enough to accommodate size of the token. Therefore, the password size limit has been increased to 32K.

Message Interface Versions are now Available Through the API (IPSO-232)

The TIBCO iProcess Objects Server and TIBCO iProcess Server Objects (.NET) message interface versions are now available via the following new properties:

- sBase.ServerMsgInterfaceVer
- sBase.ClientMsgInterfaceVer

These message interface versions can be used by GUIs and applications to conditionally show functionality based on the services available in the TIBCO iProcess Objects Server and TIBCO iProcess Server Objects (.NET) client.

IPv6 IP Addressing is Now Supported (IPSO-230)

When constructing a **vNodeId** object for connecting to a TIBCO iProcess Objects Server or Director, IPv6 addressing can now be used if the IBCO iProcess Objects Server or Director is using an IPv6 address.

New Parameter on Close and Purge Case Methods to Trigger Event (IPSO-229)

A new *aDoEvents* parameter has been added to the following methods:

- CloseCases
- CloseCasesBy Criteria
- PurgeCases
- PurgeCasesBy Criteria

- PurgeAndReset

This new parameter can be used to specify that a procedure-level event is to be triggered when cases are closed or purged with these methods. The procedure can be defined to catch the event, then perform business logic either before or after the cases are closed or purged.

Version 11.4.0

Only Strongly Named Assemblies Now Installed (IPSO-205)

Previously, the assemblies that were installed in the top-level installation directory (`ssoDotNet.dll`, `ssoDotNetV.dll`, and `ssoDotNetXML.dll`) were weakly named, and strongly named assemblies were installed in the `\Strongly Named` directory.

Weakly named assemblies are no longer installed, and the `\Strongly Named` directory is no longer created. The assemblies listed above are now all strongly named, and they are installed in the top-level installation directory.

Unlimited Number of Case Numbers Now Possible (IPSO-202)

The case number has been modified internally to allow for an unlimited number of possible case numbers (previously, a limit could be reached).

Note that this has no impact on the publicly available/visible case number (for example, in SW_CASENUM).

Added Ability to Change Client Log File Name via Registry (IPSO-45)

A new **LogId** Registry entry is now created by TIBCO iProcess Server Objects (.NET) that allows the specification of the client log file name via the Registry (it can also be specified programmatically using the **LogId** property on the **sClientLog** object).

Version 11.3.0

No new features in this release.

Version 11.2.0

Ability to Lock First Available Work Item Added (38405)

The following object, properties, and methods have been added to TIBCO iProcess Server Objects (.NET) to allow you to lock the first available work item in a list of work items:

- **LockFirstItem** - This new method has been added to the **sPageableListR** object. It is used to lock the first available work item in a list of work items when you are using pageable lists.
- **LockFirstWorkItem** - This new method has been added to the **sWorkQ** and **xWorkQ** objects. It is used to lock the first available work item in a list of **vWorkItem** objects when you are using "single-block item access" lists (i.e., the **MakeWorkItemList** method was used to create the list of **vWorkItem** objects).
- **LockFirstAWorkItem** - This new method has been added to the **sWorkQ** and **xWorkQ** objects. It is used to lock the first available work item in a list of **vAWorkItem** objects when you are using "single-block item access" lists (i.e., the **MakeAWorkItemList** method was used to create the list of **vAWorkItem** objects).
- **vWILocked** - This new object is returned by the three new "lock" methods listed above. This object contains the following properties, which provide information about the work item that was locked:
  - **Index** - Returns the index at which the first available work item was found.
  - **LockedItem** - Returns the locked work item.
  - **WIFieldGroup** - Returns a **vWIFieldGroup** object, which represents the set of fields in the work item that was locked.

Note: To make use of these new lock methods/properties, you must be using a TIBCO iProcess Objects Server that has implemented MR 38404.

Version 11.1.0

Ability to Limit the Number of Cases Returned from Server Added (36787)

The following changes have been made to the object model to allow you to specify the maximum number of cases to return from the server.

- A new constructor has been added to the **vACaseCriteria** object that contains an *aMaxCnt* parameter. Passing a value in this parameter limits the number of cases returned from the server to that number. Using a constructor without the *aMaxCnt* parameter, or passing -1, causes all cases to be returned.
- A new **MaxCnt** property has been added to the **vACaseCriteria** object. This property returns the value passed in the *aMaxCnt* parameter on the **vACaseCriteria** constructor.
- A new **OverMaxCnt** property has been added to the **vSummary** object. This property returns the number of cases that were *not* returned from the server because the number returned was limited using the *aMaxCnt* parameter on the **vACaseCriteria** constructor when retrieving a list of cases.

Version 11.0.0

New Methods Added to Fetch a Work Item List Only if it Has Changed (34071)

New “fetch” methods have been added that return work items only if there has been a change in work items on the list since you originally obtained it. The following new methods have been added to both the **sWorkQ** and **xWorkQ** objects:

- **FetchWorkItemListIfChanged**
- **FetchAWorkItemListIfChanged**

Both of these new methods *always* refresh the list (there is no *Refresh* parameter — it is assumed to be True), and they return the requested work items only if *any* items in the *entire* list have changed (not just the specified range) since the list was originally created

(with the **MakeWorkItemList** or **MakeAWorkItemList** method). If no work items in the list have changed since it was originally created, the returned list is zero length.

This MR also resulted in changed behavior for the following methods on the **sWorkQ** and **xWorkQ** objects:

- **MakeWorkItemList** and **MakeAWorkItemList** - These methods now *always* return a list status of **swPLChanged**. The list status is available in the **PLStatus** property on the **vWorkItemListState** and **vAWorkItemListState** objects.
- **FetchWorkItemList** and **FetchAWorkItemList** - If these methods are called with *aRefresh*=false, the list status is always returned as **swPLChanged**. The list status is available in the **PLStatus** property on the **vWorkItemListState** and **vAWorkItemListState** objects.

Obtaining Work Queue Deltas Via a Subscription to a JMS Topic (33058)

Client applications can now obtain work queue deltas via a subscription to a JMS topic. For a step-by-step process of how the application accomplishes this, see the *TIBCO iProcess Server Objects (.NET) Programmer's Guide*.

The following changes been made to the object model to accommodate this enhancement:

sWorkQ

The **sWorkQ** object has had the following methods added:

- **GetAWorkItemListJMS** - This method is used to inform the Work Item Server (WIS) that the application will be receiving work queue deltas via a subscription to a JMS topic. A topic name must be passed in the method call to tell the WIS the JMS topic to which it must publish deltas. This method returns an **sPageableListJ** object (a new object) that contains two IDs that were returned by the WIS: **WorkQDeltaId** and **WorkQSyncId**. These IDs are needed by the application when calling the **startWorkQDeltaJMSPublish** method to start JMS delta publication from the WIS.

  This method is used if you are using pageable lists.
- **GetAWorkItemListJMSHeld** - This method is used to retrieve an **sPageableListJ** object that had been previously held.
- **MakeAWorkItemListJMS** - This method is used to inform the WIS that the application will be receiving work queue deltas via a subscription to a JMS topic. A topic name must be passed in the method call to tell the WIS the JMS topic to which it must publish deltas.

This method returns a **vAWorkItemListState** object that contains three IDs that were returned by the WIS: **HeldId, WorkQDeltaId** and **WorkQSyncId**. These IDs are needed by the application when calling the **StartWorkQDeltaJMSPublish** method to start JMS delta publication from the WIS.

This method is used if you are using single-block item access (not pageable lists).

- **StartWorkQDeltaJMSPublish** - This method tells the WIS to establish the baseline list of work items, and to start publishing the work queue deltas to the JMS topic.
- **FetchAWorkItemListJMS** - This method is used to get the baseline list of work items after the application has told the WIS to start publishing deltas.

  This method is used if you are using single-block item access (not pageable lists).

xWorkQ

The **xWorkQ** object has had the following methods added:

- **MakeAWorkItemListJMS** - This method is used to inform the WIS that the application will be receiving work queue deltas via a subscription to a JMS topic. A topic name must be passed in the method call to tell the WIS the JMS topic to which it must publish deltas.

  This method returns a **vWorkQDeltaJMSId** object (a new object) that contains a HeldId, WorkQDeltaId and WorkQSyncId. These IDs are needed by the application when calling the **StartWorkQDeltaJMSPublish** method to start JMS delta publication from the WIS.
- **StartWorkQDeltaJMSPublish** - This method tells the WIS to establish the baseline list of work items, and to start publishing the work queue deltas to the JMS topic.
- **FetchAWorkItemListJMS** - This method is used to get the baseline list of work items after the application has told the WIS to start publishing deltas.

vAWorkItemListState

This object has had the following properties added:

- **WorkQDeltaId** - This returns the WorkQDeltaId, which is returned by the WIS when you set up work queue delta publication by calling the **MakeAWorkItemListJMS** method. It must be passed in the **StartWorkQDeltaJMSPublish** method.
- **WorkQSyncId** - This method returns the WorkQSyncId, which is returned by the WIS when you set up work queue delta publication by calling the

MakeAWorkItemListJMS method. It must be passed in the **StartWorkQDeltaJMSPublish** method.

sSession / xSession

This object has had the following method added:

- **KeepAliveWorkQDeltaJMSPublications** - This method is used to inform the WIS that the application is still active and wants the WIS to continue to publish deltas to JMS. If this method is not called, the WIS session will timeout, by default, after 8 hours of inactivity.

sPageableListJ

This new object, which is returned from the **GetAWorkItemListJMS** method, contains the following method / properties:

- **StartWorkQDeltaJMSPublish** - This method tells the WIS to establish the baseline list of work items, and to start publishing the work queue deltas to the JMS topic.

  This method is used if you are using pageable lists to access work items.
- **WorkQDeltaId** - This property returns the WorkQDeltaId, which is returned by the WIS when you set up work queue delta publication by calling the **GetAWorkItemListJMS** method. It must be passed in the **StartWorkQDeltaJMSPublish** method.
- **WorkQSyncId** - This property returns the WorkQSyncId, which is returned by the WIS when you set up work queue delta publication by calling the **GetAWorkItemListJMS** method. It must be passed in the **StartWorkQDeltaJMSPublish** method.

vWorkQDeltaJMSId

This new Value Object is returned by the **xWorkQ.MakeAWorkItemListJMS** method when you are setting up delta publication to JMS. It contains IDs that are needed when calling the **StartWorkQDeltaJMSPublish** method to inform the WIS to create the baseline list of work items and to start publication of work queue deltas to a JMS topic. The HeldId is also needed when calling the **FetchAWorkItemListJMS** method to get the baseline list of work items.

This new object contains the following properties:

- **HeldId - This property returns the held ID needed to identify the work item list when calling the StartWorkQDeltaJMSPublish and FetchAWorkItemListJMS methods.**
- **WorkQDeltaId** - This property returns the WorkQDeltaId, which is returned by the WIS when you set up work queue delta publication by calling the **MakeAWorkItemListJMS** method. It must be passed in the **StartWorkQDeltaJMSPublish** method.
- **WorkQSyncId** - This property returns the WorkQSyncId, which is returned by the WIS when you set up work queue delta publication by calling the **MakeAWorkItemListJMS** method. It must be passed in the **StartWorkQDeltaJMSPublish** method.

Ability to Set, Get, and Delete User Preference Data Added (32432)

The TIBCO iProcess Server Objects (.NET) object model has been updated to allow you to set, get, and delete user preference data as a text string in the iProcess Engine database.

User preference data is stored as a name/value pair in the database. The value is usually XML, but can be any text data. The value length is limited to 128K.

The following changes have been made to implement this enhancement:

- A **vPreference** object has been added that holds a name/value pair that represents the user preference data. This object must be constructed, then passed in the new **SetUserPreference** method to save a user's user preference data. This object is returned by the new **GetUserPreference** method. This new object contains the following properties:
  - **Name** - Returns the name (key) given to the preference data when it is saved in the database.
  - **Value** - Returns the user preference data.
- The **sNode**, **xNode**, **sUser**, and **xUser** objects have had the following methods added:
  - **SetUserPreference** - Saves the user preference data in the database. You must pass in the name of the user for whom you are saving user preference data, and a **vPreference** object, which contains a preference name (key) and the data.
  - **GetUserPreference** - Returns a **vPreference** object containing the user preference data. You must pass in the name of the user and the preference name (key) to identify the user preference data in the database.

- **DeleteUserPreference** - Deletes the specified user preference data from the database.

Users can modify their own user preference data (using the methods on **sUser**/**xUser**). Only administrators can modify user preference data for other users (using the methods on **sNode**/**xNode**).

```
<configuration>
startup useLegacyV2RuntimeActivationPolicy="true">
</startup>
</configuration>
```

This allows your application to use the TIBCO iProcess Server Objects (.NET) libraries that were complied against .NET Framework 2.0.

Changes in Platform Support

Supported platforms are not affected in this release. For a complete list of supported platforms, see the Readme file.

Change in Functionality

No functionality changes have been made in this release of TIBCO iProcess® Server Objects (.NET).

Deprecated and Removed Features

No features have been deprecated or removed in this release of TIBCO iProcess® Server Objects (.NET).

Other Information

Rounding of Large Values May Occur

Because of the data types used in the TIBCO iProcess Server Objects (.NET), rounding may occur in data fields that contain very large numbers. The floating point precision provided in the data types used is 15-17 digits of accuracy.

For more information about this issue, see the following websites:

- http://msdn2.microsoft.com/en-us/library/Aa691146(VS.71).aspx
- http://mindprod.com/jgloss/floatingpoint.html

Encoding Using ICU Conversion Libraries

ICU conversion libraries can be used to specify the desired character encoding.

To use the ICU conversion libraries, you must create the following Registry entry and set it to the name of the converter you wish to use.

- TISOUnicodeConverterName

The Registry entry must be located in the following path:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Staffware plc\Staffware SSO Client\
```

Note:
If the software is installed on a 64-bit machine, the Registry path will include "Wow6432Node" as follows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Staffware plc\...
```

For a list of converter names, and information about each converter, see the Converter Explorer.

Note: When using the ICU libraries, the converter you use must reserve positions 00 through 1F for the standard single-byte ASCII control characters. This ensures that the control characters do not otherwise occur in the byte stream. (The UTF-16 converter, for example, does not satisfy this requirement, and therefore, cannot be used.)

If the **TISOUnicodeConverterName** Registry entry does not exist, or is set to an invalid value, the ICU libraries are not used. In this case, the system looks for the **TISOMultiChar** Registry entry in the following path:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Staffware plc\Staffware SSO Client\
```

Note:
If the software is installed on a 64-bit machine, the Registry path will include "Wow6432Node" as follows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Staffware plc\...
```

If the TISOMultiChar Registry entry exists and is set to 1, UTF-8 (multi-byte) encoding is used, otherwise extended ASCII (single-byte) encoding is used. The system will only look at the TISOMultiChar Registry entry if the TISOUnicodeConverterName Registry entry does not exist or is set to an invalid converter name.

For more information about ICU, see ICU-TC.

Must Rebuild if Upgrading From Pre-Version 10.6.0

If you are upgrading your TIBCO iProcess Server Objects (.NET) from a version prior to 10.6.0 to a version 10.6.0 or newer, you must:

- reference the ICU files (**icuuc36.dll** and **icudt36.dll**) that are new in version 10.6.0, and

- rebuild your application.

During installation, the new ICU files are installed in the TIBCO iProcess Server Objects (.NET) installation directory. For more information about ICU, see Encoding Using ICU Conversion Libraries.

Raw Data Buffers

The **sRawData** object allows you to obtain raw data directly from the message buffers. The **sRawData** object contains the following public constructor:

```
public sRawData(vNodeId aNodeId,
String aUserName,
String aPassword)
```

The **sRawData** object can also be obtained using the **Create_sRawData** method on the **sSession** object.

The **sRawData** object contains the following methods:

- **GetFirstWIBuffer** - This method returns the raw data for the specified number of work items from the specified work queue. There are two signatures for this method, as follows:

  ```
  Byte[] GetFirstWIBuffer(String aWorkQTag, int aMaxWICnt)
  ```

  ```
  Byte[] GetFirstWIBuffer(String aWorkQTag,
                          vWICriteria aCriteria,
                          String[] aCaseFieldNames,
                          int aMaxWICnt)
  ```

  where:

  — **aWorkQTag** is the tag identifying the work queue from which the work items are to be retrieved.

  — **aCriteria** is a **vWICriteria** object, which specifies the filter and sort criteria for the request.

  — **aCaseFieldNames** specifies the case data fields to be returned.

— **aMaxWICnt** is the number of work items to return from the call to **GetFirstWIBuffer**. Passing a -1 in this parameter causes all work items in the buffer to be retrieved.

- **GetNextWIBuffer** - This method returns the next block of work items (the size of which is specified in the *aMaxWICnt* parameter with the **GetFirstWIBuffer** method). This method returns a null if the entire message has been received and is complete. Its syntax is as follows:

```
Byte[] GetNextWIBuffer()
```

- **ReleaseBuffers** - This method releases any segments if the application did not retrieve all of the raw data from the buffer (i.e., the **GetNextWIBuffer** method wasn't called until a null was returned). Its syntax is as follows:

```
void releaseBuffers()
```

Note: If **GetFirstWIBuffer** is called again before all of the work items are retrieved from the buffer, the previous raw data is automatically released from the buffer before the work items are retrieved from the server again.

Note: These methods do not allow for filtering nor sorting the work items. They also return all CDQP fields, but do not return case data.

Raw Data Buffer Format

The following describes the format of the raw data buffers so that they can be parsed after acquiring the raw data with the **GetFirstWIBuffer** and **GetNextWIBuffer** methods.

The message format makes use of the following special byte values:

- ELEMDELIM = Chr(0)
- LISTTERM = Chr(30)
- RECTERM = Chr(29)
- COUNTMARK = Chr(28)

where:

— ELEMDELIM is used to delimit all elements of the message.

- LISTTERM is used to terminate a list.
- RECTERM is used to terminate the data for a record (work item in this case).
- COUNTMARK is used to locate the values of the counters in the message.

The size of each buffer is controlled by the **TCPResponsePages** parameter in the TIBCO iProcess Objects Server. The format of the data in each buffer is a series of data elements. All elements of the message (including the LISTTERM, RECTERM and COUNTMARK) are delimited by the ELEMDELIM character.

All message elements are printable ASCII values. In other words, values of type date, time or numeric are returned as their ASCII string equivalents and not as their native binary values.

The response format of the QQ message is listed below and uses the following syntactical notation.

Syntax	Description
element1, element2	element1 and element2 are separated by NULL (0x00).
{element1, element2, element3, …}	There are 0 or more occurrences (a list) of the ordered tuple in the data stream. The list is terminated with the LISTTERM, (0x1E) and a NULL (0x00). Lists can be embedded (lists can contain lists). Each list is terminated with 0x1E and 0x00.
element	Elements in **bold** are literal values.
RECTERM	This delimits the end of a logical record (a workitem in this case). The value of the delimiter is 0x1D followed by a NULL (0x00).
LISTTERM	This delimits the end of a list. The value of the delimiter is 0x1E followed by a NULL (0x00).
COUNTMARK	This is used to indicate the beginning of the count data. The value of the delimiter is 0x1C followed by a NULL (0x00).

Each buffer (or message segment) returned by the server has as its first element the message code of the request. For the **GetFirstWIBuffer** method, that message code will always be the value QQ. When the application parses the response buffers, it needs to skip the QQ value that is at the beginning of each message segment.

The basic strategy to parsing a list is to grab the first element of the list. If the value is LISTTERM, then you are at the end of the list. If the value is not LISTTERM, then that is the value of the first element; the subsequent elements comprise the remaining elements of the list.

Individual elements and their trailing delimiter never span message segments. However, a logical record (meaning the elements that comprise a work item) can span message segments. This spanning between elements can occur at any point in the logical record.

The format of date and time values is controlled by the TIBCO iProcess Engine settings. You can see the TIBCO iProcess Engine date and time settings via the **DateFormat**, **DateSeparator**, and **TimeSeparator** properties on the **vNodeLocale** object. These are configured in the **staffpms** file. If a date is uninitialized, the value is December 31, 3000.

Some elements of a message are only sent by the TIBCO iProcess Objects Server if both the client and server are at a particular interface level or newer. If this applies to an element, it is notated in the format specification below with the minimum interface level of the client and server that would be needed for the element to appear in the data stream. The current elements this impacts are **IsKeepOnWithdraw**, **IsSuspended**, **ProcMajorVer**, **ProcMinorVer**, **FieldIsArray**, **CDQPName** and **CDQPValue**.

Note:
The data buffer format described here is based on a TIBCO iProcess Objects Server message interface of 3.1.0, and a TIBCO iProcess Server Objects client message interface of 3.1.0. If you upgrade to a newer server and client in the future, you may start receiving new data elements in your message (depending on the message interface versions of the client and server). If this occurs, you will need to adjust your parsing logic to accommodate them. You can determine the TIBCO iProcess Objects Server's interface version by running:

```
$SWDIR/bin/swentobjsv -V
```

You can determine the TIBCO iProcess Server Objects Client's interface version by accessing the **InterfaceVersion** property on **vClientInfo**.

Data Buffer Format

The following is the format of the data stream contained in the buffers:

ComputerName, Status {IsUnopen, IsLocked, IsDeadline, IsDeadlineExp, IsStartStep, IsUndelivered, IsAvailable, IsLongLock, IsUrgent, IsAutoPurge, IsWorkdays, IsNetworked, IsEditable, IsForwardable, IsReleasable, IsDeadlineAWD, IsKeepOnWithdraw, IsSuspended,

ArrivedDate, ArrivedTime, DeadlineDate, DeadlineTime, CaseDescription, StepName, StepDescription, CaseNumber, ProcName, ProcDescription, ProcNumber, CaseReference, ProcNode, WQParam1, WQParam2, WQParam3, WQParam4, Priority, LockedBy, StartedBy, AddrToName, MailID, ExtraPackFile, ExtraStepNum, ExtraCaseCtlRec, ExtraReqID, ExtraPNumCnt, ProcMajorVer, ProcMinorVer, {FieldName, FieldType, FieldIsArray, FieldLength, FieldDecimalPlaceCnt, FieldValue}, {CDQPName, CDQPValue}, RECTERM}, COUNTMARKER, Count, OverMaxCnt, InvalidCnt, ExcludeCnt

where:

ComputerName = Name of the computer where server is running

Status = you will always get a value of **2**

IsUnopen = **1** if item is unopened, **0** if item is not unopened

IsLocked = **1** if item is locked, **0** if item is not locked

IsDeadline = **1** if item has deadline, **0** if no deadline

IsDeadlineEXP = **1** if deadline has expired, **0** if not expired

IsStartStep = **1** if item from case start, **0** if not

IsUndelivered = **1** if item is undelivered, **0** if not

IsAvailable = **1** if item is available, **0** if not (orphaned)

IsLongLock = **1** if item is long locked, **0** if not

IsUrgent = **1** if item is urgent, **0** if not

IsAutoPurge = **1** if item's case is auto-purge, **0** if not

IsWorkdays = **1** if item's case uses workdays, **0** if not

IsNetworked = **1** if item's procedure is networked, **0** if not

IsEditable = **1** if item is editable, **0** if not

IsForwardable = **1** if item is forwardable, **0** if not

IsReleasable = **1** if item is directly releasable, **0** if not

IsDeadlineAWD = **1** if item is auto-withdrawn on deadline expiration,

0 if not

IsKeepOnWithdraw = **1** if deadline NOP, **0** if not (interface 3.0.0)

IsSuspended = **1** if item is suspended, **0** if not (interface 2.0.4)

ArrivedDate = Date item arrived in queue (in server format)

ArrivedTime = Time item arrived in queue (in server format)

DeadlineDate = Deadline date (in server format)

DeadlineTime = Deadline time (in client format)

CaseDescription = Case description for case

StepName = Name of step for work item

StepDescription = Description of step for work item

CaseNumber = Case number of case for work item

ProcName = Name of procedure for work item

ProcDescription = Description of procedure for work item

ProcNumber = Internal number of procedure for work item

CaseReference = Case reference

ProcNode = Hosting node for procedure

WQParam1 = Value for work queue parameter 1

WQParam2 = Value for work queue parameter 2

WQParam3 = Value for work queue parameter 3

WQParam4 = Value for work queue parameter 4

Priority = Priority of work item

LockedBy = User who has item locked

StartedBy = User who started the case

AddrToName = Addressee of work item

MailID = An ID unique to this item in this queue only

ExtraPackfile = Pack file name for work item

ExtraStepNum = Number of step within procedure

ExtraCaseCtlRec = Case control record number

ExtraReqID = Unique Identifier for this item

ExtraPNumCnt = Slot usage for ProcNum

ProcMajorVer = Major version number of case that work item is from

(interface 3.0.0)

ProcMinorVer = Minor version number of case that work item is from

(interface 3.0.0)

FieldName = Name of case data field

FieldType = **D** for date

= **A** for ASCII text

= **R** for real number

= **T** for time

= **X** for attachment

= **F** for memo

= **C** for composite (table)

= **N** for comma separated (numeric)

FieldIsArray = **1** if field is an array field, **0** if not (interface 3.0.0)

FieldLength = Length of field

FieldDecimalPlaceCnt = Number of decimal places (for real number and comma

separated)

FieldValue = Value of field

CDQPName = Name of the CDQP field (interface 1.1.1)

CDQPValue = Value of the CDQP field (interface 1.1.1)

Count = Number of items returned (value will always be 0 if

MaxCount <> -1)

OverMaxCnt = Number of items matching expression not returned

because of Max Count specification

InvalidCnt = Number of items not returned because expression was

invalid in the case context

ExcludeCnt = Number of items not returned because the expression

explicitly excluded them

Buffer Data in the Log File

If you want to see what some sample buffers would look like, you can enable MsgTrace in the TIBCO iProcess Objects Server and look at the trace dumps in the log file for a QQ message (which you can generate by iterating calling the getWorkItems method). If you do this, note that each message segment as traced in the TIBCO iProcess Objects Server log file contains a header. This header is stripped off by the **GetFirstWIBuffer** and **GetNextWIBuffer** methods. This header on each segment is a fixed 17-byte header containing the following elements:

Description	Size	Format
Total Segment Length (not counting this field)	4 bytes fixed	Unsigned Binary in Network Byte Order
Request ID (echoed from client request)	4 bytes fixed	Unsigned Binary in Network Byte Order
Segment ID	1 byte fixed	F = First segment M = Middle segment L = Last segment O = Only segment
Transaction Status (only applicable if Segment ID is L or O). Positive on success, 0 or negative on error.	4 bytes fixed	Signed Binary in Network Byte Order
Total Cumulative Length (total number of bytes in all message fragments for this response). Only applicable if Segment ID is L or O.	4 bytes fixed	Unsigned Binary in Network Byte Order

Large Number of User Logins May Cause Problems

If your client application performs a large number of user logins to the TIBCO iProcess Server Objects (for example, in a stateless or web environment), it may experience problems that are a result of running out of TCP ports, or attempting to connect to a TCP port above 5000. You may see one of the following problems on either the client or TIBCO iProcess Server Objects:

- the system hanging or exhibiting unexpected behavior
- unable to create new connections to the server
- Event ID 2009 “Number of sessions exceeds 2048” may be written to the event log
- the local computer responds with a WSENOBUFS (10055) error

For information about resolving these issues, refer to Microsoft Knowledge Base articles Q149532 and Q196271. These can be found at the Microsoft website.

Must Normalize Case Data to View Cases

When you install the iProcess Engine, there is a check box in the **Configuration** window that is used to enable a feature called **case data normalization**. If you are using an application developed with iProcess Server Objects, you **must** enable case data normalization. If it is not enabled, cases will not appear in the case list.

Case data normalization can be enabled either during the installation, or at a later time using the Case Data Normalization Utility, **swnormcd**.

For more information about case data normalization and using **swnormcd**, see the *TIBCO iProcess Engine Administrator’s Guide*.

Cannot Run in Same Process as iProcess Objects

You cannot run TIBCO iProcess Server Objects in the same process as TIBCO iProcess Objects. Both products in the same process do not run properly, and is not supported.

Known Issues

Uninstall Must be Performed Before Upgrading to Version 11.4.0

If you are upgrading to version 11.4.0 of TIBCO iProcess® Server Objects (.NET) from a previous version, you *must* uninstall the previous version before installing version 11.4.0. For information about uninstalling, see the *Repairing/Removing* TIBCO iProcess Server Objects (.NET) chapter in the *TIBCO iProcess® Server Objects (.NET) Installation* guide.

Closed Issues

The table in this section lists issues that were closed in the named releases of TIBCO iProcess® Server Objects (.NET).

Closed in Release	CR #	Summary
11.4.1	IPSO-242	A means to specify case close and purge priority needs to be added. Implemented. See New Parameter on Close and Purge Case Methods to Specify Priority (IPSO-242).
11.4.1	IPSO-239	The current size limit of 4K for passwords need to be increased. Implemented. See Password Size Limit Has Been Increased (IPSO-239).
11.4.1	IPSO-234	Some audit actions are missing in **SWAuditType**, which results in the Action Processor returning exceptions to TIBCO iProcess Workspace (Browser). Corrected.
11.4.1	IPSO-232	The TIBCO iProcess Objects Server and TIBCO iProcess Server Objects (.NET) message interface versions need to made available through the API. Implemented. See Message Interface Versions are now Available Through the API (IPSO-232).
11.4.1	IPSO-231	The XSD for **MaxCnt** on the **vACaseCriteria** object indicates that the count is required (minOccurs="1"). It should indicate that it is optional (minOccurs="0"). Corrected.
11.4.1	IPSO-230	Support for IPv6 IP addressing needs to be added for connecting iProcess Server Objects to the TIBCO iProcess Objects Server or

Closed in Release	CR #	Summary
		Director. Implemented. See IPv6 IP Addressing is Now Supported (IPSO-230).
11.4.1	IPSO-229	A means to specify that a procedure-level event be triggered when closing and purging cases needs to be added. Implemented. See New Parameter on Close and Purge Case Methods to Trigger Event (IPSO-229).
11.4.1	IPSO-228	When SW_CASENUM is a marking on a form, and the value is greater than 107, the value returned by **xWorkQ.LockItems** is in exponential format. Corrected.
11.4.0	IPSO-205	Installed assemblies should all be strongly named. Implemented. See Only Strongly Named Assemblies Now Installed (IPSO-205).
11.4.0	IPSO-202	Internally, the case number needs to be modified to allow for an unlimited number of possible case numbers. Implemented. See Unlimited Number of Case Numbers Now Possible (IPSO-202).
11.4.0	IPSO-194	A number of methods that contain a list as an argument can cause a stack overflow. Corrected.
11.4.0	IPSO-191	The assembly version number for the 32-bit .NET DLLs is different than the 64-bit .NET DLLs. They should be the same so that a program compiled against the 32-bit .NET DLLs will also run against the 64-bit .NET DLLs, and vice versa. Corrected.

Closed in Release	CR #	Summary
11.4.0	IPSO-188	Some of the legends on configuration values in the client log file are very similar to the environment variable names, but with different casing, which can cause confusion. Corrected. All legends have been changed to be plain English so there is no confusion with environment variable names.
11.4.0	IPSO-182	There is a memory leak when **sWorkQ.GetForwardToWorkQIds** is called. Corrected.
11.4.0	IPSO-178	The username should be included with messages sent to the iProcess Objects Director so that the Director can use the username when making load balancing decisions. Corrected.
11.4.0	IPSO-45	The ability to change the log file name via the Registry should be added. Implemented. See Added Ability to Change Client Log File Name via Registry (IPSO-45).
11.3.0	43171	If there is an error when keeping, releasing, undoing, or unlocking multiple work items, the work item tags in the exception details are *all* for the first work item with an error. Corrected.
11.3.0	43126	If an **sClientLog** or **sNodeManager** object is constructed before **sSession**, the product version in the log is “UNKNOWN”. Corrected.
11.3.0	43116	The 32-bit version of TIBCO iProcess Server Objects (.NET) needs to be moved from .NET Framework 1.1 to 2.0. Implemented.

Closed in Release	CR #	Summary
11.3.0	43037	If you attempt to get *all* procedure definitions with the **GetProcDefs** method (by not passing an *aProcTags* argument), and request procedure audit data (**vProcAudit** objects) in the **vProcDefContent** object, the audit data is not returned with the procedure definitions. (It is returned if you request specific procedure definitions by passing an *aProcTags* argument.) Corrected.
11.2.0	42839	Errors in the client log state that the 'msgtype' of the response did not match the 'msgtype' of the request. This occurs because a mutex added as part of 64-bit support was not initialized at start-up. Corrected.
11.2.0	41348	If the **Value** property of a **vAttribute** object contains an **SWEmptyField** object, XML serialization fails with an exception. Corrected.
11.2.0	41082	In the XML interface, when a **vField** object whose type is ArrayOf... (ArrayOfText, ArrayOfMemo, ArrayOfNumeric, ArrayOfComma, ArrayOfDate, or ArrayOfTime) contains a value such as “one||three”, only two array elements are generated. It should generate three elements. Corrected.
11.2.0	41080	If a **vField** object whose type is ArrayOf... (ArrayOfText, ArrayOfMemo, ArrayOfNumeric, ArrayOfComma, ArrayOfDate, or ArrayOfTime) contains an **SWEmptyField** object, the Java Virtual Machine (and CLR) crashes. Corrected. If the condition described above occurs, an exception will now be thrown.
11.2.0	40962	Numeric value’s DecimalSeparator and CommaSeparator are not converted to the server format before being sent to the server.

Closed in Release	CR #	Summary
		Corrected.
11.2.0	39005	No work items are returned by the **FetchAWorkItemListJMS** method if the requested number of work items (*aReturnCount* parameter) exactly equals the number of available work items. Corrected.
11.2.0	38885	If there are a large number of role names defined on the system, a call to **sUser.GetRoleNames** can cause a stack overflow. Corrected.
11.2.0	38452	The usernames and passwords returned by the **GetDatabaseConfig** method are frequently corrupted. Corrected.
11.2.0	38405	The ability to lock the first available work item needs to be added to TIBCO iProcess Server Objects (.NET). Implemented. See Ability to Lock First Available Work Item Added (38405).
11.1.0	38304	TIBCO iProcess Server Objects (.NET) is missing audit action type **swForwardedBy** (value 41). Corrected. The missing audit action type has been added.
11.1.0	37120	If custom .NET cultures are used that don't have separator characters in the dates or times, invalid DateTime errors are returned. Corrected.
11.1.0	36787	The ability to limit the number of cases returned from the server needs to be added. Implemented. See Ability to Limit the Number of Cases Returned from Server Added (36787).

Closed in Release	CR #	Summary
11.1.0	36348	If a null is passed as an argument on one of the user preference methods (**SetUserPreference**, **GetUserPreference**, or **DeleteUserPreference**) on either **sUser** or **sNode**, it results in a fatal exception. It should return an invalid argument exception. Corrected.
11.1.0	35882	Assertions are enabled in the released version of TIBCO iProcess Server Objects (.NET). Corrected.
11.0.0	34669	Deserialization of custom audit steps results in an **InvalidOperationException**. Corrected.
11.0.0	34259	When the **FetchWorkItemList** or **FetchAWorkItemList** methods are called, the list status always returns “swPLNoChange” even if a refresh was requested and there were changes in the list. (The list status is available in the **PLStatus** property on the returned **vWorkItemListState** object.) Corrected.
11.0.0	34071	New methods are needed to fetch a work item list only if there has been a change in the list since it was originally obtained. Implemented. See New Methods Added to Fetch a Work Item List Only if it Has Changed (34071).
11.0.0	33058	Client applications need the ability to obtain work queue deltas via a subscription to a JMS topic rather than through the TIBCO iProcess Server Objects (.NET) object model. Implemented. See Obtaining Work Queue Deltas Via a Subscription to a JMS Topic (33058).
11.0.0	32432	The ability to set, get, and delete user preference data in the engine

Closed in Release	CR #	Summary
		database needs to be added. Implemented. See Ability to Set, Get, and Delete User Preference Data Added (32432).
11.0.0	31095	The **FetchWorkItemListDelta** and **FetchAWorkItemListDelta** methods are not returning any work items when -1 is passed in the *aMaxCount* parameter, when they should return all delta work items. Corrected.
11.0.0	30952	The information shown in a long list of procedures is corrupted. Corrected.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the TIBCO iProcess® Server Objects (.NET) Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, and iProcess are either registered trademarks or trademarks of Cloud Software Group, 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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software 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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.