



TIBCO iProcess® Workspace (Browser)

Action Processor Reference

Version 11.10.0 | May 2025

Contents

Contents	2
Introduction	4
Overview	4
Actions and Requests	6
Request Groups	8
Schemas	10
Character Encoding	11
Action Processor Session	13
Date and Time Format	13
Responses to Actions	13
Response Status	14
Requests	19
Introduction	19
Authentication	20
Login	20
Logout	21
Utility	22
Ping	22
CaseManager	22
Form	30
RenderForm	30
StartCaseForm	30
KeepForm	31
ReleaseForm	31
UndoForm	31
Node	32

NodeManager	37
ProcManager	38
IPEConfig	41
Session	42
User	43
WorkQ	47
WorkQManager	53
Parameters	58
Parameters Details	58
TIBCO Documentation and Support Services	76
Legal and Third-Party Notices	78

Introduction

This section provides an overview of the TIBCO iProcess Workspace (Browser) Action Processor.

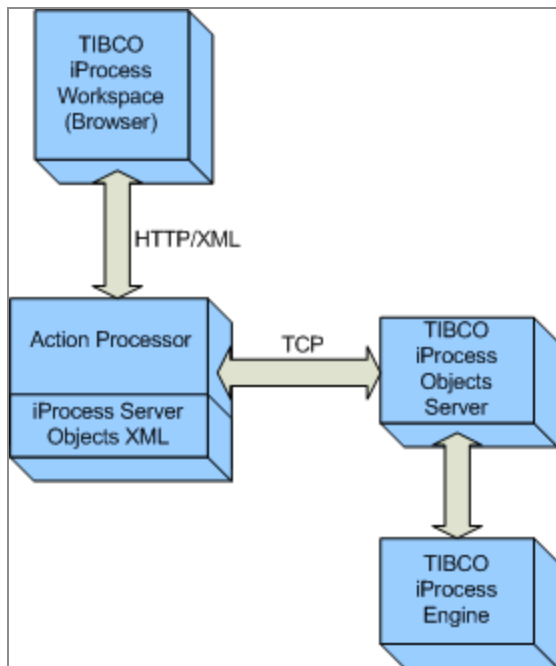
Overview

The **TIBCO iProcess Workspace (Browser) Action Processor** services action requests made by the TIBCO iProcess Workspace (Browser) application (which was developed using TIBCO® General Interface Builder).

Although the Action Processor is the engine that processes requests from iProcess Workspace (Browser), it can also be used directly from any application that can send HTTP requests and receive HTTP responses. The intent of this document is to provide details about each request/reply that is supported by the Action Processor so that it can be used by a custom application.

Each 'action' requested by the TIBCO® General Interface Builder GUI components can contain one or more of the following 'request' types:

- Request for TIBCO iProcess data (e.g., a list of work items)
- Request an action be executed on the TIBCO iProcess Engine (e.g., start a case)



The TIBCO® General Interface Builder GUI components send their action details to the Action Processor via a Form POST over HTTP — the action details are sent as XML on an HTML **<input>** tag.

The Action Processor also accepts actions via an HTTP GET. When doing a GET, the action XML needs to be sent as a URL parameter named 'action'.

Upon receiving an action request (that is not a form request), the Action Processor breaks the action into individual requests and processes each one. The resulting data from each request is then consolidated in XML under one root element (**<ap:ActionResult>**). The Action Processor then returns this XML data to the TIBCO® General Interface Builder GUI components over HTTP.

The Action Processor is implemented as both a Microsoft .NET ASP Web Application and as a Java Servlet. This has been done so as to satisfy both the Microsoft and customer base. The implementation of the two Action Processors is obviously very different but the functionality and configuration is the same.

Form Requests

If the action request is a form request, the Action Processor redirects the HTML to the client application — no XML is returned. Form requests are handled as follows:

1. The Action Processor looks at the **ExternalFormURI** parameter in the Action Processor's configuration file, **apConfig.xml**. If there is a value in this parameter, it

assumes the form is an “External Form”, which means it’s an ASP Form, JSP Form, or FormBuilder Form. The Action Processor uses the value in the **ExternalFormURI** parameter as the “base” URI to the form’s location. The remainder of the URL is obtained from the value in the **Formflow Form** field in the step definition.

2. If there is no value in the **ExternalFormURI** parameter, the Action Processor assumes the “default form” type, i.e., an iProcess Modeler-produced form. In this case, the form to display is defined in the step definition.

In the case of “GI Forms” (which includes both TIBCO® General Interface Forms and TIBCO Forms), the Action Processor does not get involved. These form types are handled by the TIBCO® General Interface code running in the client application.

Actions and Requests

Each action, designated by the **<ap:Action>** element, can contain one or more requests to the Action Processor. The following action example contains two requests; the first request performs a login (**Login** request in the **Authentication** group) and the second request gets the details of three users (**GetUsers** request in the **Node** group).

```

<?xml version="1.0" encoding="UTF-8"?>
<ap:Action xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sso="http://tibco.com/bpm/sso/types"
  xmlns:ap="http://tibco.com/bpm/actionprocessor">

  <ap:Authentication>
    <ap:Login Id="x435t">
      <sso:NodeCtx>
        <sso:NodeId>
          <sso:Name>myNode</sso:Name>
          <sso:ComputerName>Mickey</sso:ComputerName>
          <sso:IPAddress>10.20.30.75</sso:IPAddress>
          <sso:TCPPort>46578</sso:TCPPort>
          <sso:IsDirector>false</sso:IsDirector>
        </sso:NodeId>
        <sso:UserName>bobbie</sso:UserName>
        <sso:Password></sso:Password>
      </sso:NodeCtx>
    </ap:Login>
  </ap:Authentication>

  <ap:Node>
    <ap:GetUsers Id="kk6w4">
      <sso:UserContent>
        <sso:IsWithAttributes>true</sso:IsWithAttributes>
        <sso:IsWithGroups>true</sso:IsWithGroups>
        <sso:IsWithRoles>true</sso:IsWithRoles>
        <sso:IsWithWorkQs>false</sso:IsWithWorkQs>
      </sso:UserContent>
      <sso:UserNames>
        <sso:string>fredu</sso:string>
        <sso:string>jamesd</sso:string>
        <sso:string>maryc</sso:string>
      </sso:UserNames>
    </ap:GetUsers>
  </ap:Node>
</ap:Action>

```

Action requests are submitted to the Action Processor URL, where the URL is in the following format:

```
http://Host:Port/APDir/ActionProcessor.ext
```

where:

- *Host* is the name of the machine hosting the Action Processor. (Note that if you are hosting both the iProcess Workspace (Browser) and the Action Processor on the same machine, and they are both being hosted by Tomcat, you can specify *Host* as "localhost".)
- *Port* is the port number used by the Web Application Server (WAS) that is hosting the Action Processor to communicate with web applications.
- *APDir* is the directory on *Host* in which the Action Processor is installed.

- *ext* is the file name extension. This is "servlet" (for Java servlet) if you are connecting to a Java Action Processor, or "aspx" (for .NET ASP web application) if you are connecting to a .NET Action Processor.

The format for passing the action request to the Action Processor is as follows:

```
[ActionProcessorURL]?action=[ActionRequest]
```

For example:

```
http://myserver/ActionProcessor/ActionProcessor.aspx?action=<?xml%20vers
ion="1.0"%20encoding="UTF-
8"?><ap:Action%20xmlns:ap="http://tibco.com/bpm/actionprocessor"%20xmlns
:sso="http://tibco.com/bpm/sso/types"><ap:Authentication><ap:Login%20Id
="jsx_ipcNS_
23Login"><sso:NodeCtx><sso:NodeId><sso:Name>i2tagtest</sso:Name><sso:Com
puterName>ozquadling</sso:ComputerName><sso:IPAddress>10.97.8.55</sso:IP
Address><sso:TCPPort>58002</sso:TCPPort><sso:IsDirector>>false</sso:IsDir
ector></sso:NodeId><sso:UserName>swadmin</sso:UserName><sso:Password/></
sso:NodeCtx></ap:Login></ap:Authentication></ap:Action>
```

Id Parameter

Note that every request to the Action Processor contains an **Id** parameter. For example:

```
<ap:Authentication>
  <ap:Login Id="x435t">
    .
    .
    .
```

The **Id** parameter can be any string; it is simply echoed back in the response (it is reflected in the **ID** attribute of the **<vResult>** tag in the XML response data). Its purpose is to correlate replies from requests when you make multiple asynchronous requests. You can also pass NULL in the **Id** parameter, which causes the **ID** attribute in the response to be the name of the method call with an initial capital letter.

Request Groups

All Action Processor requests are organized into the following groups:

- Authentication
- Utility
- CaseManager
- Form
- Node
- NodeManager
- ProcManager
- Session
- User
- WorkQ
- WorkQManager

With the exception of **Authentication**, **Utility**, and **Form**, the groups correspond to the XML Server Object of the same name, and contain a request for each method on the XML Server Object. For example, the **ProcManager** group contains 14 requests, one for each of the 14 methods on the **xProcManager** object.



Note: Another exception is the **Session** request group, which only contains a single request — **KeepAliveWorkQDeltaJMSPublications** — whereas the **xSession** object contains that method, as well as additional methods used to create other XML Server Objects.

The **Authentication**, **Utility**, and **Form** groups contain requests that do not appear on any XML Server Object. They contain requests used to login/logout, for working with iProcess Modeler-created forms, and for pinging the Action Processor.

All request elements must be contained within the proper group element, and only one request can be made for each group element. If multiple requests from a group are to be made, multiple group elements must be submitted.

For example, to submit requests for **GetStartProclds** and **GetWorkQs**, both of which are in the **User** group, they must be submitted as follows:

```

...
<ap:User>
  <ap:GetStartProcIds Id="u886a"/>
</ap:User>

<ap:User>
  <ap:GetWorkQs Id="pl07d"/>
  <sso:WorkQTags>
    <sso:string>willey|kellyp|R</sso:string>
  </sso:WorkQTags>
</ap:User>
...

```

Schemas

XML Schema Definition (XSD) files are used to validate the XML action requests and responses as follows:

- **apAction.xsd** - This XSD can be used to validate XML action requests sent to the Action Processor. Note, however, that requests are NOT validated by default — this is configurable using the **<IsValidateActionXML>** element in the Action Processor's configuration file, **apConfig.xml** (there is some overhead incurred when validating). For information about setting this configuration parameter, see *TIBCO iProcess Workspace (Browser) Configuration and Customization*.
- **apActionResult.xsd** - You can use this XSD to validate the XML response from the Action Processor. This can be used for design purposes — when referenced in some XML development tools, it provides Microsoft IntelliSense®-like features.
- **apSSOTypes.xsd** - Both of the XSD files above reference this XSD, which is generated from the TIBCO iProcess Server Objects.

Note that all elements in this XSD are sequenced, requiring that you submit them in the specified order. For example, the **WIFGContent** parameter required by the **LockItems** request (see above) represents the **vWIFGContent** object in the iProcess Server Objects, which contains two properties: **FieldsOption** and **WIFieldNames**. Because these represent properties on an object, they must be submitted in the proper order (they are defined within a **<sequence>** element in **apSSOTypes.xsd**).

```

<xs:complexType name="vWIFGContent">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:vContentRequest">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1"
          name="FieldsOption" type="tns:SWFieldsOptionType" />
        <xs:element minOccurs="0" maxOccurs="1"
          name="WIFieldNames" type="tns:ArrayOfString" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Character Encoding

If you are manually sending requests to the Action Processor, you must:

- ensure that special characters within your XML data are properly escaped so that they are not interpreted as XML reserved characters,
- perform proper URI encoding, and
- ensure that UTF-8 encoding is specified in the XML file.

Escaping Special Characters

The following table shows the reserved XML characters that must be escaped within your XML data:

Character	Description	Include in XML as...
<	Less than symbol	<
>	Greater than symbol	>
&	Ampersand	&

Character	Description	Include in XML as...
'	Single quote (apostrophe)	'
"	Double quote	"

Perform URI Encoding

URI encoding can be performed using the following:

- **Java:**
 - java.net.URLEncoder class
- **C# (.NET):**
 - HTTPUtility.UrlEncode class
 - or
 - HTTPUtility.UrlPathEncode class
- **JavaScript:**
 - encodeURIComponent()
 - or
 - encodeURIComponent()

Specifying UTF-8 Encoding

UTF-8 encoding is specified in your XML file by including the **encoding** attribute, as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```



Note: For information about setting character encoding in the web server (using Tomcat as an example), see the *Application Server Settings* topic in the *TIBCO iProcess Workspace (Browser) Configuration and Customization* guide.

Action Processor Session

The first action the TIBCO iProcess Workspace (Browser) requests is a **Login** request. When the Action Processor receives the **Login** request, it logs into the TIBCO iProcess Objects Server to validate the user's credentials. Then it stores the login details (**vNodeCtx** object) in the session. Subsequent requests to the Action Processor are processed using the login details stored in the session. The **vNodeCtx** remains in the session for the lifetime of the session or until it is replaced by different login credentials from the same client.

Date and Time Format

All dates and times that are sent to the Action Processor must be in the proper XML format. If they are in the wrong format, an error is thrown by the serializer code when the Action Processor passes the date/time to the serializer.

Dates and times must be in the following format:

```
"yyyy-mm-ddThh:mm:ss.00000000"
```

where:

- yyyy-mm-dd = the year, month, and day
- T = a separator between the date and time
- hh:mm:ss = the hour, minute, and second
- .0000000 represents milliseconds, but is always zeros in this context

For example:

```
2005-06-29T10:30:28.00000000
```

Dates and times returned by the Action Processor are also in the format shown above.

Responses to Actions

Once the Action Processor has processed all of the individual requests in an action, it consolidates the data under a single XML root element, **<ap:ActionResult>**. This XML response is then sent back to the client via HTTP.

The format of the XML response is as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<ap:ActionResult xmlns:ap='http://tibco.com/bpm/actionprocessor'>

  <ap:Status>
    <ap:Version></ap:Version>
    <ap:ReturnCode></ap:ReturnCode>
    <ap:ReturnComment></ap:ReturnComment>
    <ap:ReturnDateTime></ap:ReturnDateTime>
    <ap:ExceptionMessage></ap:ExceptionMessage>
    <ap:ExceptionStackTrace></ap:ExceptionStackTrace>
  </ap:Status>

  <ap:SSO>
    <sso:vSSOData xmlns:sso="http://tibco.com/bpm/sso/types">
      <sso:Results>
        <!-- The Results of all Requests in the Action -->
      </sso:Results>
    </sso:vSSOData>
  </ap:SSO>

</ap:ActionResult>
```

Response Status

The XML response to the client contains an **<ap:Status>** element that indicates whether or not the action was successfully processed. It also contains information to help determine the problem if an error does occur.

The **<ap:Status>** element provides the version number of the Action Processor and the date and time the response was returned to the client. For example:

```
<?xml version="1.0"?>
<ap:ActionResult xmlns:ap="http://tibco.com/bpm/sso/types/ActionProcessor">

  <ap:Status>
    <ap:Version>11.3.0</ap:Version>
    <ap:ReturnCode>0</ap:ReturnCode>
    <ap:ReturnComment>The Action was processed successfully. Check the individual
      Request Results for their status.</ap:ReturnComment>
    <ap:ReturnDateTime>2006-03-03T16:10:16.211-0800</ap:ReturnDateTime>
  </ap:Status>
```

i Note: Whether the Action Processor version is displayed in the response status is determined by the **IsReturnVersion** parameter in the **apConfig.xml** file. For more information, see the *TIBCO iProcess Workspace (Browser) Configuration and Customization* guide.

If the **<ap:ReturnCode>** element contains a 0 (zero), the action was processed successfully.

If the **<ap:ReturnCode>** element contains any number other than a 0, an error has occurred.

The following table lists all of the possible return codes, as well as the comment that is returned in the **<ap:ReturnComment>** element for each return code. The comment can be used to help determine the cause of the error.

ReturnCode	Return Comment
0	The action was processed successfully. Check the individual Request Results for their status.
-1	Exception Thrown. ¹
-2	There was no Action specified.
-3	Invalid Action XML.
-4	The Action contains no Requests.
-5	The Request has no Name specified.
-6	There can only ever be a single Request in an Action when that Request is a HTML Request.
-7	There is no node context associated with this session, a Login is required.
-8	Failed to create xSession.

¹This indicates that an error has occurred somewhere other than in the Action Processor, e.g., in the iProcess Objects Server. For more information see [Errors Outside the Action Processor](#).

ReturnCode	Return Comment
-9	Unknown Request type.
-10	Invalid apConfig XML.
-11	Internal error, see Action Processor log for details.
-12	No WorkItem tag was specified.
-13	Unable to determine Procedure.
-14	Unable to determine Step.
-15	Unknown Form type.
-16	This error indicates an XML validation error. The return comment will be a detailed description of the syntax error in the XML.

The value in the **<ap:ReturnDateTime>** element is in the form:

```
yyyy-mm-ddThh:nn:ss.uuu+/-oooo
```

where:

- yyyy-mm-dd = year, month, and day the XML response was sent
- T = separator between the date and time
- hh:nn:ss.uuu = hour, minute, second, and microsecond the XML response was sent
- +/- = indicates whether the offset from GMT is plus (+) or minus (-)
- oooo = offset (in 2400-hour format) from Greenwich Mean Time (GMT)

For example:

```
2006-04-26T08:02:02.146-0700
```

In this example, the “-0700” indicates minus 7 hours from GMT.

Errors Outside the Action Processor

If an error occurs somewhere outside of the Action Processor — for instance, in the iProcess Objects Server — the Action Processor will pass the error through and show it in the **<ap:Status>** element.

If an error occurs outside the Action Processor, a -1 is returned in the **<ap:ReturnCode>** element, and two additional elements are shown under **<ap:Status>**. They are shown below:

```
<ap:Status>
  <ap:Version></ap:Version>
  <ap:ReturnCode></ap:ReturnCode>
  <ap:ReturnComment></ap:ReturnComment>
  <ap:ReturnDateTime></ap:ReturnDateTime>
  <ap:ExceptionMessage></ap:ExceptionMessage>
  <ap:ExceptionStackTrace></ap:ExceptionStackTrace>
</ap:Status>
```

where:

- **<ap:ExceptionMessage>** is the exception message that is being passed through by the Action Processor.
- **<ap:ExceptionStackTrace>** provides information on the execution history of the current thread, displaying the names of the classes and methods within those classes that had been called at the point when the exception occurred. Due to security reasons, this section will remain empty in the HTML response but the stack trace can be seen in the Action Processor log.

The following shows an example response when an exception occurs outside of the Action Processor:

```
<?xml version='1.0' encoding='UTF-8'?>
<ap:ActionResult xmlns:ap='http://tibco.com/bpm/actionprocessor'>

  <ap:Status>
    <ap:Version>11.3.0</ap:Version>
    <ap:ReturnCode>-1</ap:ReturnCode>
    <ap:ReturnComment>Exception Thrown.</ap:ReturnComment>
    <ap:ReturnDateTime>2006-03-09T14:55:53.747-0800</ap:ReturnDateTime>
    <ap:ExceptionMessage>
      <![CDATA[ Unknown username or bad password. ]]>
    </ap:ExceptionMessage>
    <ap:ExceptionStackTrace>
      <![CDATA[]]>
    </ap:ExceptionStackTrace>
  </ap:Status>
```

Requests

This chapter provides a list of all Action Processor requests, including the parameters required for each request.

Note that you will need to use the TIBCO iProcess Server Objects (Java or .NET) on-line help system in conjunction with this reference guide. The on-line help provides details about the parameters that need to be passed with each request.

Introduction

Action Processor requests are organized in this chapter by request group.

The parameters required for each request are listed in the **Parameter** column of the table shown for each request. The details of these parameters are provided in the [Parameters](#) chapter on [Parameters](#). You can cross-reference the parameter names with the details of the parameters by clicking on the parameter name in the **Parameters** column (note that some requests allow multiple sets of parameters).

With the exception of **Authentication**, **Utility**, and **Form**, the request groups correspond to the XML Server Object of the same name, and contain a request for each method on the XML Server Object. For example, the **ProcManager** group contains 14 requests, one for each of the 14 methods on the **xProcManager** object.

 **Note:** Another exception is the **Session** request group, which only contains a single request — **KeepAliveWorkQDeltaJMSPublications** — whereas the **xSession** object contains that method, as well as additional methods used to create other XML Server Objects.

The **Authentication**, **Utility**, and **Form** groups contain requests that do not appear on any XML Server Object. They contain requests used to login/logout, for working with iProcess Modeler-created forms, and for pinging the Action Processor.

i Note: This document does not provide details about the operation of each method/request. For that information, see the TIBCO iProcess Server Objects (Java or .NET) on-line help system.

Authentication

The following table lists the requests available in the **Authentication** group. Click the request name to view request parameter information.

Request Name	Description
Login	Logs user into the TIBCO iProcess Objects Server
Logout	Logs user out of the TIBCO iProcess Objects Server

The Action Processor requests listed in this section must be enclosed in an **<Authentication>** element.

Login

The **Login** request logs the user into the TIBCO iProcess Objects Server, using the credentials provided in the **vNodeCtx** object. (Note that there is no Login method on the XML Server Objects because there is an implied login each time an XML Server Object is constructed — user credentials are passed on the object constructor.)

Request Name	Parameter
Login	NodeCtx

**Note:**

It is a good practice to enable security by adding a two character unique code per Application, xId (for example, xId="AB"), to the login requests as an attribute in the Login Id tag above.

Using <ap:Login Id="x435t" xId="AB"> instead of <ap:Login Id="x435t"> generates "acrostic" value that is returned in Response Headers.

Further, all subsequent requests to ActionProcessor, must contain "acrostic" parameter and value as Request Header or Query Parameter.

Login Request Example:

```
<ap:Authentication>
  <ap:Login Id="x435t">
    <sso:NodeCtx>
      <sso:NodeId>
        <sso:Name>myNode</sso:Name>
        <sso:ComputerName>Mickey</sso:ComputerName>
        <sso:IPAddress>10.20.30.75</sso:IPAddress>
        <sso:TCPPort>46578</sso:TCPPort>
        <sso:IsDirector>>false</sso:IsDirector>
      </sso:NodeId>
      <sso:UserName>bobbie</sso:UserName>
      <sso>Password></sso>Password>
    </sso:NodeCtx>
  </ap:Login>
</ap:Authentication>
```

Logout

The **Logout** request logs the user out of the TIBCO iProcess Objects Server.

Request Name	Parameter
Logout	None
	ReleaseAllResources

ReleaseAllResources = "false" causes the "user session" to be closed, which closes the TCP connection between the client and the iProcess Objects Server.

ReleaseAllResources = "true" (the default) causes the "user session" to be closed, which closes the TCP connection between the client and the iProcess Objects Server. This also

closes the “SAL session”, which releases all lists held on the iProcess Objects Server — this releases all client-side and server-side resources.

Utility

The following table lists the requests available in the **Utility** group. Click the request name to view request parameter information.

Request	Description
Ping	Returns True if the Action Processor is running.

The Action Processor requests listed in this section must be enclosed in a **<Utility>** element.

Ping

The **Ping** request allows you to determine if the Action Processor is running. It returns True if the Action Processor is running, False if it is not running.

Request Name	Parameter
Ping	None

CaseManager

The following table lists the requests available in the **CaseManager** group. Click the request name to view request parameter information.

Request Name	Description
GetProcIds	Get procedure IDs

Request Name	Description
GetACases	Get cases
MakeACaseList	
FetchACaseList	
GetAuditSteps	Manage cases
SetCaseData	
SetCaseSuspended	
GetCaseCnt	
GetOutstandingItems	
GetSubProcCases	
JumpTo	
TriggerEvent	
ReleaseEAllItem	
SetGraftTaskCnt	Control graft steps
StartGraftTask	
DeleteGraftTask	
ExternalGraftProcessComp	
PredictCase	Case prediction
MakePredictedItemList	
FetchPredictedItemList	
CloseCases	Close cases
CloseCasesByCriteria	
PurgeCases	Purge cases
PurgeCasesByCriteria	

Request Name	Description
PurgeAndReset	

The Action Processor requests listed in this section must be enclosed in a **<CaseManager>** element.

CaseManager Group	
Request Name	Parameters
GetProcIds	None
GetACases	CaseTags ACaseContent
MakeACaseList	ProcTag ACaseCriteria ACaseContent StartIndex ReturnCount Hold
FetchACaseList	HeldId StartIndex ReturnCount Hold
GetAuditSteps	CaseTag SortAscending
SetCaseData	CaseTag Fields

CaseManager Group	
Request Name	Parameters
SetCaseSuspended	CaseTags
	Suspend
	Wait
	CaseTag
	Suspend
	Wait
	Fields
	UpdateOutstanding
GetCaseCnt	ProcTag
	FilterExpression
GetOutstandingItems	CaseTag
	OutstandingItemContent
	IncludeSubProcs
GetSubProcCases	CaseTag
	IncludeSubProcs
JumpTo	CaseTag
	WithdrawList
	SendList
	Reason
	Wait
	CaseTag
	WithdrawList

CaseManager Group	
Request Name	Parameters
	SendList
	Reason
	Wait
	Fields
	UpdateOutstanding
TriggerEvent	CaseTag
	EventName
	CaseTag
	EventName
	Fields
	UpdateOutstanding
	DeadlineCalculate
	CaseTag
	EventName
	Resurrect
	Fields
ReleaseEAllItem	ExternalId
	Fields
	ExternalId
	Fields
	NextStep
	DoActions

CaseManager Group	
Request Name	Parameters
SetGraftTaskCnt	CaseTag GraftId StepName TaskCnt
StartGraftTask	CaseTag GraftId StepName GraftSubTasks
DeleteGraftTask	CaseTag GraftId StepName
ExternalGraftProcessComp	CaseTag GraftId StepName GraftSubTask
PredictCase	CaseTag CaseTag Fields CaseTag MaxSubProc MaxStepLoop CaseTag

CaseManager Group	
Request Name	Parameters
	MaxSubProc
	MaxStepLoop
	Fields
MakePredictedItemList	PredictionCriteria StartIndex ReturnCount Hold
FetchPredictedItemList	HeldId StartIndex ReturnCount Hold
CloseCases	CaseTags CaseTags DoEvent Priority
	ProcTag ProcTag DoEvent Priority
CloseCasesByCriteria	CloseCriteria ProcTags

CaseManager Group	
Request Name	Parameters
	CloseCriteria
	ProcTags
	DoEvent
	Priority
PurgeCases	CaseTags
	CaseTags
	DoEvent
	Priority
	ProcTag
	ProcTag
	DoEvent
	Priority
PurgeCasesByCriteria	PurgeCriteria
	ProcTags
	PurgeCriteria
	ProcTags
	DoEvent
	Priority
PurgeAndReset	ProcTags
	ProcTags
	DoEvent
	Priority

Form

The following table lists the requests available in the **Form** group. Click the request name to view request parameter information.

Request Name	Description
RenderForm	Locks the work item and displays the HTML representation of the TIBCO iProcess Modeler-generated form
StartCaseForm	Displays the first form of the specified procedure
KeepForm	Closes the form, keeps the work item, and saves data
ReleaseForm	Closes the form, releases the work item, and saves data
UndoForm	Closes the form, keeps the work item, but does not save data

The Action Processor requests listed in this section must be enclosed in a **<Form>** element.

RenderForm

The **RenderForm** request locks the work item and displays the HTML representation of the TIBCO iProcess Modeler form. The work item is identified by a work item tag.

Request Name	Parameter
RenderForm	WorkItemTag

StartCaseForm

The **StartCaseForm** request displays the HTML representation of the TIBCO iProcess Modeler form for the first step in the specified procedure.

Request Name	Parameters
StartCaseForm	ProcTag Description ProcTag Description SubProcPrecedence

KeepForm

The **KeepForm** request closes the form, unlocks the work item, and keeps the work item in the work queue. The *WIFieldData* parameter identifies the work item to keep, as well as provides updated work item data to save.

Request Name	Parameter
KeepForm	WIFieldData

ReleaseForm

The **ReleaseForm** request closes the form, unlocks the work item, then releases the work item. The *WIFieldData* parameter identifies the work item to release, as well as provides updated work item data to save.

Request Name	Parameter
ReleaseForm	WIFieldData

UndoForm

The **UndoForm** request closes the form, unlocks the work item. and keeps the work item in the work queue. The *WIFieldData* parameter identifies the work item to unlock. Work item

data is not saved.

Request Name	Parameter
UndoForm	WIFieldData

Node

The following table lists the requests available in the **Node** group. Click the request name to view request parameter information.

Request Name	Description
GetANode	Get vANode object
GetUserAttributes	Manage user attributes
GetGroupAttributes	
ChangeAttributes	
GetAttributeDefs	
CreateAttributeDefs	
DeleteAttributeDefs	
GetAGroups	Manage groups
MakeAGroupList	
FetchAGroupList	
CreateGroups	
DeleteGroups	
AddUsersToGroups	
RemoveUsersFromGroups	
GetOSUsers	Manage OS users

Request Name	Description
MakeOSUserList	
FetchOSUserList	
GetUsers	Manage users
MakeUserList	
FetchUserList	
CreateUser	
DeleteUsers	
GetRoles	Manage roles
CreateRoles	
DeleteRoles	
GetTables	Manage tables and lists
GetListValidations	
MakeWorkQIdList	Get work queue lists
FetchWorkQIdList	
MoveSysInfo	Update system information

The Action Processor requests listed in this section must be enclosed in a **<Node>** element.

Node Group	
Request Name	Parameters
GetANode	None
GetUserAttributes	UserName
	UserName

Node Group	
Request Name	Parameters
	AttributeNames
GetGroupAttributes	GroupName GroupName AttributeNames
ChangeAttributes	UserNames Attributes
GetAttributeDefs	None AttributeNames
CreateAttributeDefs	AttributeDefs
DeleteAttributeDefs	AttributeNames
GetAGroups	GroupNames AGroupContent
MakeAGroupList	AGroupContent StartIndex ReturnCount Hold
FetchAGroupList	HeldId StartIndex ReturnCount Hold

Node Group	
Request Name	Parameters
CreateGroups	Groups
DeleteGroups	GroupNames
AddUsersToGroups	GroupNames UserNames
RemoveUsersFromGroups	GroupNames UserNames
GetOSUsers	OSUserNames
MakeOSUserList	StartIndex ReturnCount Hold
FetchOSUserList	HeldId StartIndex ReturnCount Hold
GetUsers	UserNames UserContent
MakeUserList	UserContent StartIndex ReturnCount Hold

Node Group	
Request Name	Parameters
FetchUserList	HeldId StartIndex ReturnCount Hold
CreateUser	UserName Attributes GroupNames
DeleteUsers	UserNames
GetRoles	None RoleNames
CreateRoles	Roles
DeleteRoles	RoleNames
GetTables	None TableNames
GetListValidations	None ListValidationNames
MakeWorkQIdList	StartIndex ReturnCount Hold
FetchWorkQIdList	HeldId

Node Group	
Request Name	Parameters
	StartIndex ReturnCount Hold
MoveSysInfo	None

NodeManager

The following table lists the requests available in the **NodeManager** group. Click the request name to view request parameter information.

Request Name	Description
VerifyNode GetNodes GetANode	iProcess Objects Server/Director discovery.
SetSrvLogOptions ResetSrvLog	Setting server log options.

The Action Processor requests listed in this section must be enclosed in a **<NodeManager>** element.

NodeManager Group	
Request Name	Parameters
VerifyNode	ComputerId

NodeManager Group	
Request Name	Parameters
	NodeName UDPPortNumbers Instance
GetNodes	UDPPortNumbers
GetANode	NodeCtx
SetSrvLogOptions	NodeCtx Categories LogLevel Trace MaxSize
ResetSrvLog	NodeCtx

ProcManager

The following table lists the requests available in the **ProcManager** group. Click the request name to view request parameter information.

Request Name	Description
GetProcIds GetProcs GetAProcs	Get procedures
GetProcDefs	Get procedure definitions

Request Name	Description
GetProcVersions	
GetStepIds	Get step/field definitions
GetSteps	
GetPublicSteps	
GetFieldDefs	
GetForm	Get form definitions
GetFormMarkings	
GetExternalForm	
GetPluginForm	
SimulateCase	Case simulation

The Action Processor requests listed in this section must be enclosed in a **<ProcManager>** element.

ProcManager Group	
Request Name	Parameters
GetProcIds	None ProcTags
GetProcs	None ProcTags
GetAProcs	AProcContent ProcTags AProcContent

ProcManager Group	
Request Name	Parameters
GetProcDefs	ProcDefContent ProcTags ProcDefContent
GetProcVersions	ProcName AProcContent
GetStepIds	ProcTag
GetSteps	ProcTag StepContent ProcTag StepNames StepContent
GetPublicSteps	ProcTag StepContent
GetFieldDefs	ProcTag
GetForm	ProcTag StepName
GetFormMarkings	ProcTag StepName
GetExternalForm	ProcTag StepName

ProcManager Group	
Request Name	Parameters
GetPluginForm	ProcTag StepName
SimulateCase	ProcTag ProcTag Fields StepNames ProcTag MaxStepLoop MaxSubProc Fields StepNames

IPEConfig

The following table lists the requests available in the **IPEConfig** group. Click the request name to view request parameter information.

Request	Description
GetActivityPub	Returns an XML string that represents a Message Event Request (MER) message. The MER specifies how activity monitoring occurs in the iProcess Engine. If the request with no parameter is used, the MER will include activity monitoring configuration information for all procedures for which it has been defined.
SetActivityPub	Used to set the activity monitoring (event publication) configuration for the iProcess Engine.

Request	Description
	You must pass a Message Event Request (MER) message in the method call in the form of an XML string. The MER message can be written with any available XML tool. The MER message must conform to the SWMonitorList.xsd schema (which is written to the SWDIR\schemas directory when the iProcess Engine is installed).

The Action Processor requests listed in this section must be enclosed in a **<IPEConfig>** element.

Request Name	Parameter
GetActivityPub	None
	ProcName
SetActivityPub	String

Session

The following table lists the requests available in the **Session** group. Click the request name to view request parameter information.

Request	Description
KeepAliveWorkQDeltaJMSPublications	Notifies the WIS that the application is still active and wants the WIS to continue to publish deltas to JMS.
GetClientMsgInterfaceVer	Returns the iProcess Server Objects Client Message Interface version. This version can be used by GUIs and applications to conditionally show functionality based on the services available in the iProcess Server Objects

Request	Description
	Client.
GetServerMsgInterfaceVer	Returns the iProcess Objects Server Message Interface version. This version can be used by GUIs and applications to conditionally show functionality based on the services available in the iProcess Objects Server.

The Action Processor requests listed in this section must be enclosed in a **<Session>** element.

Request Name	Parameter
KeepAliveWorkQDeltaJMSPublications	None
GetClientMsgInterfaceVer	None
GetServerMsgInterfaceVer	None

User

The following table lists the requests available in the **User** group. Click the request name to view request parameter information.

Request Name	Description
StartCase	Start case
GetAuditProcIds	Determine user's audit/start permissions
GetStartProcIds	
GetWorkItems	Get user-related information

Request Name	Description
GetUserId	
GetProcs	
GetSupervisedQIds	
GetWorkQIds	
GetWorkQs	
GetAttributes	
GetGroups	
GetRoleNames	
ChangePassword	Manage passwords
IsPasswordExpired	
AddCaseAuditEntry	Manage case audit trail
GetCustomAuditMsgDefs	
SetUserPreference	Manage user preference data
GetUserPreference	
DeleteUserPreference	

The Action Processor requests listed in this section must be enclosed in a **<User>** element.

User Group	
Request Name	Parameters
StartCase	ProcTag
	Description
	ProcTag
	Description

User Group	
Request Name	Parameters
	ReleaseItem
	ValidateFields
	Fields
	ProcTag
	Description
	SubProcPrecedence
	StartStepName
	ReleaseItem
	ValidateFields
	Fields
GetAuditProcIds	None
GetStartProcIds	None
GetWorkItems	WorkItemTags WIContent
GetUserId	None
GetProcs	ProcTags
GetSupervisedQIds	None
GetWorkQIds	None
GetWorkQs	None WorkQTags

User Group	
Request Name	Parameters
GetAttributes	None AttributeNames
GetGroups	GroupContent
GetRoleNames	None
ChangePassword	OldPassword NewPassword
IsPasswordExpired	None
AddCaseAuditEntry	CaseTag StepName StepDescription UserName MessageId
GetCustomAuditMsgDefs	None
SetUserPreference	UserName Preference
GetUserPreference	UserName PreferenceName
DeleteUserPreference	UserName PreferenceName

WorkQ

The following table lists the requests available in the **WorkQ** group. Click the request name to view request parameter information.

Request Name	Description
GetWorkQ	Get the work queue
GetWorkItems	Get work items from the queue
MakeWorkItemList	
FetchWorkItemList	
FetchWorkItemListDelta	
FetchWorkItemListIfChanged	
GetAWorkItems	
MakeAWorkItemList	
FetchAWorkItemList	
FetchAWorkItemListDelta	
FetchAWorkItemListIfChanged	
MakeAWorkItemListJMS	
StartWorkQDeltaJMSPublish	
FetchAWorkItemListJMS	
StartCase	Start a case
LockItems	Process work items
LockFirstWorkItem	
LockFirstAWorkItem	
KeepItems	
ReleaseItems	

Request Name	Description
UndoItems	
UnlockItems	
GetForwardToWorkQIds	Forward work items
ForwardItems	
GetDefaultCriteria	Manage default filter / sort criteria
ChangeDefaultCriteria	
ClearDefaultCriteria	
GetForm	Get iProcess form information
AddCaseAuditEntry	Manage case audit trail
GetCustomAuditMsgDefs	

The Action Processor requests listed in this section must be enclosed in a **<WorkQ>** element.

WorkQ Group	
Request Name	Parameters
GetWorkQ	WorkQTag
GetWorkItems	WorkQTag
	WorkItemTags
	WContent
MakeWorkItemList	WorkQTag
	WCriteria
	WContent

WorkQ Group	
Request Name	Parameters
	StartIndex
	ReturnCount
	Hold
FetchWorkItemList	WorkQTag
	HeldId
	StartIndex
	ReturnCount
	Hold
	Refresh
FetchWorkItemListDelta	WorkQTag
	HeldId
	MaxCount
	Hold
FetchWorkItemListIfChanged	WorkQTag
	HeldId
	StartIndex
	ReturnCount
	Hold
GetAWorkItems	WorkQTag
	WorkItemTags
	AWIContent

WorkQ Group	
Request Name	Parameters
MakeAWorkItemList	WorkQTag WICriteria AWIContent StartIndex ReturnCount Hold
FetchAWorkItemList	WorkQTag HeldId StartIndex ReturnCount Hold Refresh
FetchAWorkItemListDelta	WorkQTag HeldId MaxCount Hold
FetchAWorkItemListIfChanged	WorkQTag HeldId StartIndex ReturnCount Hold
MakeAWorkItemListJMS	WorkQTag

WorkQ Group	
Request Name	Parameters
	WICriteria
	TopicName
StartWorkQDeltaJMSPublish	WICriteria
	HeldId
	WorkQDeltaId
	WorkQSyncId
FetchAWorkItemListJMS	WICriteria
	HeldId
	StartIndex
	ReturnCount
StartCase	WorkQTag
	ProcTag
	Description
	SubProcPrecedence
	StartStepName
	Fields
	ReleaseItem
	ValidateFields
LockItems	WorkQTag
	WorkItemTags
	WIFGContent

WorkQ Group	
Request Name	Parameters
LockFirstWorkItem	HeldId StartIndex WIFGContent
LockFirstAWorkItem	HeldId StartIndex WIFGContent
KeepItems	WorkQTag WIFieldData ValidateFields
ReleaseItems	WorkQTag WIFieldData ValidateFields
UndoItems	WorkQTag WorkItemTags
UnlockItems	WorkQTag WorkItemTags
GetForwardToWorkQIds	WorkQTag WorkItemTag
ForwardItems	WorkQTag WorkItemTags DestWorkQTag

WorkQ Group	
Request Name	Parameters
GetDefaultCriteria	WorkQTag
ChangeDefaultCriteria	WorkQTag WICriteria
ClearDefaultCriteria	WorkQTag
GetForm	WorkQTag ProcTag StepName
AddCaseAuditEntry	WorkQTag CaseTag StepName StepDescription UserName MessageId
GetCustomAuditMsgDefs	WorkQTag

WorkQManager

The following table lists the requests available in the **WorkQManager** group. Click the request name to view request parameter information.

Request Name	Description
GetWorkQIds	Get work queue IDs

Request Name	Description
MakeWorkQIdList	
FetchWorkQIdList	
GetWorkQs	Get work queues
MakeWorkQList	
FetchWorkQList	
GetAWorkQs	
MakeAWorkQList	
FetchAWorkQList	
GetSupervisorNames	Manage work queue supervisors
AddSupervisors	
RemoveSupervisors	
GetRedirection	Manage redirection
ChangeRedirection	
CancelRedirection	
GetParticipations	Manage participation
CreateParticipations	
ChangeParticipation	
RemoveParticipations	
GetCDQPDefs	Get Case Data Queue Parameters (CDQPs)
GetClientMsgInterfaceVer	Get message interface versions
GetServerMsgInterfaceVer	

The Action Processor requests listed in this section must be enclosed in a **<WorkQManager>** element.

WorkQManager Group	
Request Name	Parameters
GetWorkQIds	WorkQTags
MakeWorkQIdList	StartIndex ReturnCount Hold
FetchWorkQIdList	HeldId StartIndex ReturnCount Hold
GetWorkQs	WorkQTags
MakeWorkQList	StartIndex ReturnCount Hold
FetchWorkQList	HeldId StartIndex ReturnCount Hold
GetAWorkQs	WorkQTags AWorkQContent
MakeAWorkQList	AWorkQContent StartIndex ReturnCount

WorkQManager Group	
Request Name	Parameters
	Hold
FetchAWorkQList	HeldId StartIndex ReturnCount Hold
GetSupervisorNames	WorkQTag
AddSupervisors	SupervisorNames WorkQTags
RemoveSupervisors	SupervisorNames WorkQTags
GetRedirection	WorkQTag
ChangeRedirection	WorkQTag Redirection
CancelRedirection	WorkQTags
GetParticipations	WorkQTag
CreateParticipations	WorkQTag Participations
ChangeParticipation	WorkQTag ExistingParticipation ChangedParticipation

WorkQManager Group	
Request Name	Parameters
RemoveParticipations	WorkQTag Participations
GetCDQPDefs	WorkQTag

Parameters

This chapter provides an alphabetical list of the parameters that are submitted with Action Processor requests.

You may need to use the TIBCO iProcess Server Objects (Java or .NET) on-line help system in conjunction with this chapter to find details about the data that is passed in these parameters.

i Note: The parameters listed in this chapter may be validated using **apSSOTypes.xsd** (validation can be turned on or off — for more information, see [Schemas](#)). Within this XSD, the parameter elements are specified in **<sequence>** elements, which means that you submit them in the specified order. They are listed in the required order in this document.

Parameters Details

Parameter Name	Format
ACaseContent	<pre> <sso:ACaseContent> <sso:IsReturnAllFields /> <sso:CaseFieldNames /> <sso:IsWithAuditData /> <sso:IsAuditAscending /> <sso:AuditFilterExpression /> </sso:ACaseContent> </pre>
ACaseCriteria	<pre> <sso:ACaseCriteria> <sso:FilterExpression /> <sso:SortFields> <sso:vSortField> <sso:FieldName /> </pre>

Parameter Name	Format
	<pre> <sso:IsAscending /> <sso:SortTypeAs /> </sso:vSortField> </sso:SortFields> <sso:MaxCnt /> </sso:ACaseCriteria> </pre>
AGroupContent	<pre> <sso:AGroupContent> <sso:IsWithAttributes /> <sso:IsWithUserNames /> </sso:AGroupContent> </pre>
AProcContent	<pre> <sso:AProcContent> <sso:IsWithProcSummary /> </sso:AProcContent> </pre>
AttributeDefs	<pre> <sso:AttributeDefs> <sso:vAttributeDef> <sso:Name /> <sso:Value /> <sso:Type /> <sso:Length /> <sso:DecimalPlaceCnt /> </sso:vAttributeDef> </sso:AttributeDefs> </pre>
AttributeNames	<pre> <sso:AttributeNames> <sso:string /> </sso:AttributeNames> </pre>
Attributes	<pre> <sso:Attributes> <sso:vAttribute /> </sso:Attributes> </pre>

Parameter Name	Format
AWIContent	<pre> <sso:AWIContent> <sso:IsReturnAllFields /> <sso:CaseFieldNames /> <sso:IsReturnAllCDQPs /> <sso:CDQPNames /> <sso:IsWithForwardToWorkQIds /> </sso:AWIContent> </pre>
AWorkQContent	<pre> <sso:AWorkQContent> <sso:IsWithParticipation /> <sso:IsWithRedirection /> <sso:IsWithSupervisorNames /> <sso:IsWithCDQPDefs /> </sso:AWorkQContent> </pre>
CaseTag	<pre> <sso:CaseTag /> </pre>
CaseTags	<pre> <sso:CaseTags> <sso:string /> </sso:CaseTags> </pre>
Categories	<pre> <sso:Categories /> </pre>
ChangedParticipation	<pre> <sso:ChangedParticipation> <sso:Name /> <sso:Index /> <sso:Qname /> <sso:StartingDate> <sso:IsValueSet /> <sso:Date /> </sso:StartingDate> <sso:EndingDate> </pre>

Parameter Name	Format
	<pre> <sso:IsValueSet /> <sso:Date /> </sso:EndingDate> <sso:StartingTime> <sso:IsValueSet /> <sso:Time /> </sso:StartingTime> <sso:EndingTime> <sso:IsValueSet /> <sso:Time /> </sso:EndingTime> <sso:IsSunday /> <sso:IsMonday /> <sso:IsTuesday /> <sso:IsWednesday /> <sso:IsThursday /> <sso:IsFriday /> <sso:IsSaturday /> <sso:UserNames> <sso:string /> </sso:UserNames> </sso:ChangedParticipation> </pre>
CloseCriteria	<pre> <sso:CloseCriteria /> </pre>
ComputerId	<pre> <sso:ComputerId /> </pre>
DeadlineCalculate	<pre> <sso:DeadlineCalculate /> </pre>
Description	<pre> <sso:Description /> </pre>
DestWorkQTag	<pre> <sso:DestWorkQTag /> </pre>

Parameter Name	Format
DoActions	<code><sso:DoActions /></code>
DoEvent	<code><sso:DoEvent /></code>
Priority	<code><sso:Priority /></code>
EventName	<code><sso:EventName /></code>
ExistingParticipation	<pre> <sso:ExistingParticipation> <sso:Name /> <sso:Index /> <sso:Qname /> <sso:StartingDate> <sso:IsValueSet /> <sso:Date /> </sso:StartingDate> <sso:EndingDate> <sso:IsValueSet /> <sso:Date /> </sso:EndingDate> <sso:StartingTime> <sso:IsValueSet /> <sso:Time /> </sso:StartingTime> <sso:EndingTime> <sso:IsValueSet /> <sso:Time /> </sso:EndingTime> <sso:IsSunday /> <sso:IsMonday /> <sso:IsTuesday /> <sso:IsWednesday /> <sso:IsThursday /> <sso:IsFriday /> </pre>

Parameter Name	Format
	<pre> <sso:IsSaturday /> <sso:UserNames> <sso:string /> </sso:UserNames> </sso:ExistingParticipation> </pre>
ExternalId	<pre> <sso:ExternalId /> </pre>
Fields	<pre> <sso:Fields> <sso:vField> <sso:Name /> <sso:FieldType /> <sso:Value /> </sso:vField> </sso:Fields> </pre>
FilterExpression	<pre> <sso:FilterExpression /> </pre>
GraftId	<pre> <sso:GraftId /> </pre>
GraftSubTask	<pre> <sso:GraftSubTask> <sso:Fields> <sso:vField> <sso:Name /> <sso:FieldType /> <sso:Value /> </sso:vField> </sso:Fields> <sso:Name /> <sso:Type /> <sso:Status /> </sso:GraftSubTask> </pre>

Parameter Name	Format
GraftSubTasks	<pre> <sso:GraftSubTasks /> <sso:GraftSubTask> <sso:Fields> <sso:vField> <sso:Name /> <sso:FieldType /> <sso:Value /> </sso:vField> </sso:Fields> <sso:Name /> <sso:Type /> <sso:Status /> </sso:GraftSubTask> </sso:GraftSubTasks> </pre>
GroupContent	<pre> <sso:GroupContent> <sso:IsWithAttributes /> </sso:GroupContent> </pre>
GroupName	<pre> <sso:GroupName /> </pre>
GroupNames	<pre> <sso:GroupNames> <sso:string /> </sso:GroupNames> </pre>
Groups	<pre> <sso:Groups> <sso:vAGroup> <sso:vGroup> <sso:vGroupId> <sso:Name /> <sso:Description /> </sso:vGroupId> <sso:Attributes> <sso:vAttribute /> </sso:Attributes> </sso:vGroup> </sso:vAGroup> </sso:Groups> </pre>

Parameter Name	Format
	<pre> </sso:Attributes> </sso:vGroup> <sso:UserNames> <sso:string /> </sso:UserNames> </sso:vAGroup> </sso:Groups> </pre>
HeldId	<code><sso:HeldId /></code>
Hold	<code><sso:Hold /></code>
IncludeSubProcs	<code><sso:IncludeSubProcs /></code>
Instance	<code><sso:Instance /></code>
ListValidationNames	<pre> <sso:ListValidationNames> <sso:string /> </sso:ListValidationNames> </pre>
LogLevel	<code><sso:LogLevel /></code>
MaxCount	<code><sso:MaxCount /></code>
MaxSize	<code><sso:MaxSize /></code>

Parameter Name	Format
MaxStepLoop	<code><sso:MaxStepLoop /></code>
MaxSubProc	<code><sso:MaxSubProc /></code>
MessageId	<code><sso:MessageId /></code>
NewPassword	<code><sso:NewPassword /></code>
NextStep	<code><sso:NextStep /></code>
NodeCtx	<pre> <sso:NodeCtx> <sso:NodeId> <sso:Name /> <sso:ComputerName /> <sso:IPAddress /> <sso:TCPPort /> <sso:IsDirector /> </sso:NodeId> <sso:UserName /> <sso:Password /> </sso:NodeCtx> </pre>
NodeId	<pre> <sso:NodeId> <sso:Name /> <sso:ComputerName /> <sso:IPAddress /> <sso:TCPPort /> <sso:IsDirector /> </sso:NodeId> </pre>

Parameter Name	Format
NodeName	<pre><sso:NodeName /></pre>
OldPassword	<pre><sso:OldPassword /></pre>
OSUserNames	<pre><sso:OSUserNames> <sso:string /> </sso:OSUserNames></pre>
OutstandingItemContent	<pre><sso:OutstandingItemContent> <sso:IsWithNormalItems /> <sso:IsWithEventItems /> <sso:IsWithEAIItems /> <sso:IsWithSubProcCallItems /> <sso:IsWithDynamicSubProcItems /> <sso:IsWithGraftItems /> <sso:IsWithTransactionControlItems /> </sso:OutstandingItemContent></pre>
Participations	<pre><sso:Participations> <sso:vParticipation> <sso:Name /> <sso:Index /> <sso:Qname /> <sso:StartingDate> <sso:IsValueSet /> <sso:Date /> </sso:StartingDate> <sso:EndingDate> <sso:IsValueSet /> <sso:Date /> </sso:EndingDate> <sso:StartingTime> <sso:IsValueSet /> <sso:Time /> </sso:StartingTime> </sso:vParticipation> </sso:Participations></pre>

Parameter Name	Format
	<pre> </sso:StartingTime> <sso:EndingTime> <sso:IsValueSet /> <sso:Time /> </sso:EndingTime> <sso:IsSunday /> <sso:IsMonday /> <sso:IsTuesday /> <sso:IsWednesday /> <sso:IsThursday /> <sso:IsFriday /> <sso:IsSaturday /> <sso:UserNames> <sso:string /> </sso:UserNames> </sso:vParticipation> </sso:Participations> </pre>
PredictionCriteria	<pre> <sso:PredictionCriteria> <sso:FilterExpression /> <sso:SortFields> <sso:vSortField> <sso:FieldName /> <sso:IsAscending /> <sso:SortTypeAs /> </sso:vSortField> </sso:SortFields> <sso:MaxSubProc /> <sso:MaxStepLoop /> </sso:PredictionCriteria> </pre>
Preference	<pre> <sso:Preference> <sso:Name /> <sso:Value /> </sso:Preference> </pre>
PreferenceName	<pre> <sso:PreferenceName /> </pre>

Parameter Name	Format
ProcDefContent	<pre> <sso:ProcDefContent> <sso:IsWithAdminBy /> <sso:IsWithStartBy /> <sso:IsWithNetworkNodes /> <sso:IsWithStepIds /> <sso:IsWithFieldDefs /> <sso:IsWithAuditData /> </sso:ProcDefContent> </pre>
ProcName	<pre> <sso:ProcName /> </pre>
ProcTag	<pre> <sso:ProcTag /> </pre>
ProcTags	<pre> <sso:ProcTags> <sso:string /> </sso:ProcTags> </pre>
PurgeCriteria	<pre> <sso:PurgeCriteria /> </pre>
Reason	<pre> <sso:Reason /> </pre>
Redirection	<pre> <sso:Redirection> <sso:StartingDateTime> <sso:IsValueSet /> <sso:DateTime /> </sso:StartingDateTime> <sso:EndingDateTime> <sso:IsValueSet /> <sso:DateTime /> </sso:EndingDateTime> <sso:WorkQName /> </pre>

Parameter Name	Format
	<code></sso:Redirection></code>
Refresh	<code><sso:Refresh /></code>
ReleaseAllResources	<code><sso:ReleaseAllResources /></code>
ReleaseItem	<code><sso:ReleaseItem /></code>
Resurrect	<code><sso:Resurrect /></code>
ReturnCount	<code><sso:ReturnCount /></code>
RoleNames	<pre> <sso:RoleNames> <sso:string /> </sso:RoleNames> </pre>
Roles	<pre> <sso:Roles> <sso:vRole> <sso:Name /> <sso:UserName /> </sso:vRole> </sso:Roles> </pre>
SendList	<pre> <sso:SendList> <sso:string /> </sso:SendList> </pre>

Parameter Name	Format
SortAscending	<code><sso:SortAscending /></code>
StartIndex	<code><sso:StartIndex /></code>
StartStepName	<code><sso:StartStepName /></code>
StepContent	<pre> <sso:StepContent> <sso:IsWithRouting /> <sso:IsWithMarkings /> <sso:IsWithPublicFields /> </sso:StepContent> </pre>
StepDescription	<code><sso:StepDescription /></code>
StepName	<code><sso:StepName /></code>
StepNames	<pre> <sso:StepNames> <sso:string /> </sso:StepNames> </pre>
SubProcPrecedence	<code><sso:SubProcPrecedence /></code>
SupervisorNames	<pre> <sso:SupervisorNames> <sso:string /> </sso:SupervisorNames> </pre>

Parameter Name	Format
Suspend	<pre><sso:Suspend /></pre>
TableNames	<pre><sso:TableNames> <sso:string /> </sso:TableNames></pre>
TaskCnt	<pre><sso:TaskCnt /></pre>
TopicName	<pre><sso:TopicName /></pre>
Trace	<pre><sso:Trace /></pre>
UpdateOutstanding	<pre><sso:UpdateOutstanding /></pre>
UDPPortNumbers	<pre><sso:UDPPortNumbers> <sso:string /> </sso:UDPPortNumbers></pre>
UserContent	<pre><sso:UserContent> <sso:IsWithAttributes /> <sso:IsWithGroups /> <sso:IsWithRoles /> <sso:IsWithWorkQs /> </sso:UserContent></pre>
UserName	<pre><sso:UserName /></pre>

Parameter Name	Format
UserNames	<pre> <sso:UserNames> <sso:string /> </sso:UserNames> </pre>
ValidateFields	<pre> <sso:ValidateFields /> </pre>
Wait	<pre> <sso:Wait /> </pre>
WIContent	<pre> <sso:WIContent> <sso:IsReturnAllFields /> <sso:CaseFieldNames> <sso:string /> </sso:CaseFieldNames> <sso:IsReturnAllCDQPs /> <sso:CDQPNames> <sso:string /> </sso:CDQPNames> </sso:WIContent> </pre>
WICriteria	<pre> <sso:WICriteria> <sso:FilterExpression /> <sso:SortFields> <sso:vSortField> <sso:FieldName /> <sso:IsAscending /> <sso:SortTypeAs /> </sso:vSortField> </sso:SortFields> </sso:WICriteria> </pre>
WIFGContent	<pre> <sso:WIFGContent> <sso:FieldsOption /> <sso:WIFieldNames> </pre>

Parameter Name	Format
	<pre> <sso:string /> </sso:WIFieldNames> </sso:WIFGContent> </pre>
WIFieldData	<pre> <sso:WIFieldData> <sso:vWIFieldGroup> <sso:WorkItemTag /> <sso:WorkItemFields> <sso:vField> <sso:Name /> <sso:FieldType /> <sso:Value /> </sso:vField> </sso:WorkItemFields> </sso:vWIFieldGroup> </sso:WIFieldData> </pre>
WithdrawList	<pre> <sso:WithdrawList> <sso:string /> </sso:WithdrawList> </pre>
WorkItemTag	<pre> <sso:WorkItemTag /> </pre>
WorkItemTags	<pre> <sso:WorkItemTags> <sso:string /> </sso:WorkItemTags> </pre>
WorkQDeltald	<pre> <sso:WorkQDeltaId /> </pre>
WorkQSyncId	<pre> <sso:WorkQSyncId /> </pre>

Parameter Name	Format
WorkQTag	<pre><sso:WorkQTag /></pre>
WorkQTags	<pre><sso:WorkQTags> <sso:string /> </sso:WorkQTags></pre>

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

Documentation for TIBCO iProcess® Workspace (Browser) is available on the [TIBCO iProcess® Workspace \(Browser\) 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.

This document 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 Cloud Software Group, Inc.

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>.

Copyright © 2005-2025. Cloud Software Group, Inc. All Rights Reserved.