

# CA Process Automation

## Web Services Reference

Service Pack 04.0.01



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2012 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA Technologies products:

- CA Catalyst for CA Service Desk Manager (CA Catalyst Connector for CA SDM)
- CA Client Automation (formerly CA IT Client Manager)
- CA Configuration Automation (formerly CA Cohesion® Application Configuration Manager)
- CA Configuration Management Database (CA CMDB)
- CA eHealth®
- CA Embedded Entitlements Manager (CA EEM)
- CA Infrastructure Insight (formerly Bundle: CA Spectrum IM & CA NetQoS Reporter Analyzer combined)
- CA NSM
- CA Process Automation (formerly CA IT Process Automation Manager)
- CA Service Catalog
- CA Service Desk Manager (CA SDM)
- CA Service Operations Insight (CA SOI) (formerly CA Spectrum® Service Assurance)
- CA SiteMinder®
- CA Workload Automation AE

# Contact CA Technologies

## Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

## Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

## Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- The following topics about Web Service methods have been updated:
  - [checkStartRequestStatus](#) (see page 19)
  - [controlProcess](#) (see page 22)
  - `executeProcess`
  - `executeStartRequest`
  - [getProcessLogs](#) (see page 58)
  - [getProcessStatus](#) (see page 65)
  - [ImportObject](#) (see page 74)
  - [controlInstance](#) (see page 21) (new)
- Various examples have been updated:
  - [Example \(<attachments> tag\)](#) (see page 14)
  - [Example \(<params> tag\)](#) (see page 15)
  - [Example \(<options> tag\)](#) (see page 15)



# Contents

---

## Chapter 1: Web Services Introduction 9

|  |   |
|--|---|
| Username/Password Authentication and Authorization ..... | 9 |
|--|---|

## Chapter 2: Tags 11

|  |    |
|--|----|
| Common Tags for Web Services Methods ..... | 11 |
| The <page> and <pages> Tags .....          | 12 |
| Example .....                              | 13 |
| The <attachments> Tag .....                | 14 |
| Example .....                              | 14 |
| The <params> Tag .....                     | 15 |
| Example .....                              | 15 |
| The <options> Tag .....                    | 15 |
| Example .....                              | 15 |

## Chapter 3: Web Services Methods 17

|  |    |
|--|----|
| AsyncSoapResponse .....                | 17 |
| checkServerStatus .....                | 18 |
| checkStartRequestStatus .....          | 19 |
| controlInstance .....                  | 21 |
| controlProcess .....                   | 22 |
| deleteArchivedInstances .....          | 25 |
| deleteAttachments .....                | 26 |
| executePendingInteraction .....        | 27 |
| executeProcess .....                   | 30 |
| executeStartRequest .....              | 33 |
| exportObject .....                     | 37 |
| generateEvent .....                    | 40 |
| getAttachments .....                   | 42 |
| getITPamVersionInfo .....              | 44 |
| getMatchingEvents .....                | 45 |
| getPendingInteractionRequestForm ..... | 48 |
| getPendingUserInteractions .....       | 51 |
| getProcessLogs .....                   | 58 |
| getProcessStatus .....                 | 65 |
| getStartRequestForm .....              | 68 |
| getStartRequestForms .....             | 70 |

---

ImportObject.....74

**Index** **79**



# Chapter 1: Web Services Introduction

---

Web services are application programming interfaces (APIs) that provide users with the ability to communicate with CA Process Automation from external sources.

The WSDL for the CA Process Automation web services interface can be accessed at:

`http(s)://<CA Process Automation Server Name>:<port>/itpam/soap?wsdl`

**Note:** Incoming CA Process Automation Web services only support SOAP version 1.1.

## Username/Password Authentication and Authorization

You can specify a userid and password for authentication.

### Example

```
<auth>
  <user>Joe</ user >
  <password>thisismypassword</ password >
</auth>
```



# Chapter 2: Tags

---

This section contains the following topics:

[Common Tags for Web Services Methods](#) (see page 11)

[The <page> and <pages> Tags](#) (see page 12)

[The <attachments> Tag](#) (see page 14)

[The <params> Tag](#) (see page 15)

[The <options> Tag](#) (see page 15)

## Common Tags for Web Services Methods

The following common tags apply to numerous Web services methods:

### [The <pages> Tag](#) (see page 12)

This tag is used in the following Web services:

- [getPendingInteractionRequestForm Method](#) (see page 48)
- [getStartRequestForm Method](#) (see page 68)

### [The <attachments> Tag](#) (see page 14)

This tag is used in the following Web services:

- [deleteAttachments Method](#) (see page 26)
- [executeStartRequest Method](#) (see page 33)
- [getAttachments Method](#) (see page 42)
- [excuteProcess Method](#) (see page 30)

### [The <params> Tag](#) (see page 15)

This tag is used in the following Web services:

- [checkStartRequestStatus Method](#) (see page 19)
- [executePendingInformation Method](#) (see page 27)
- [executeProcess Method](#) (see page 30)
- [executeStartRequest Method](#) (see page 33)
- [getProcessStatus Method](#) (see page 65)

### [The <options> Tag](#) (see page 15)

This tag is used in the following Web services:

- [executeProcess Method](#) (see page 30)
- [executeStartRequest Method](#) (see page 33)

## The <page> and <pages> Tags

In a SOAP response, the <pages> tag represents the pages in a start request form/interaction request form. For every page in the start request form/interaction request form, there is a <page> tag under the <pages> tag.

The parameters in that page are shown under the <page> tag. The <page> tag contains an attribute name which shows the name of the page.

A corresponding tag is under <Page> for most of the types of variables in the form dataset.

Mapping between dataset variables and tags under <Page>.

- Boolean\itpamBoolean
- Date\itpamDate
- Double\itpamDouble
- Integer\itpamInt
- Long\itpamLong
- Password\itpamPassword
- String\itpamString

**Note:** CA Technologies does not support ValueMaps, Object References or Arrays through Web services.

The attributes of the parameter element include:

- Validation attributes of the parameter
- Name of the parameter

The following appear under this node:

- Label
- Value (not returned in case of password)
- Description (returned if not blank)
- Default value (returned if not blank)

The <option> tag displays with the options you specify (if you specify predefined values for a parameter in a start request form/interaction request form). The isallowOtherValue attribute of <option> tag is true if the user can specify a value other than what is provided under option.

## Example

```
<pages hasComplexType="false">
  <page name="Parameters">
    <itpamString isReadOnly="false" maxLength="254" minLength="0"
name="Var_0" sequenceNo="0">
      <label>Var_0</label>
      <description/>
      <value>pink</value>
<defaultValue>op10</defaultValue>
      <options isallowOtherValue="false">
        <op name="option1">op1</op>
        <op name=" option2">op2</op>
      </options>
    </itpamString>
    <itpamInt isReadOnly="false" maxval="2147483647"
minval="-2147483648" name="Var_4" sequenceNo="4">
      <label>Var_4</label>
      <description/>
      <value>0</value>
      <defaultValue>0</defaultValue>
    </itpamInt>
    <itpamLong isReadOnly="false" maxval="9223372036854775807"
minval="-9223372036854775808" name="Var_1" sequenceNo="1">
      <label>Var_1</label>
      <description/>
      <value>0</value>
      <defaultValue>0</defaultValue>
    </itpamLong>
    <itpamDouble isReadOnly="false" maxval="1.7976931348623157E308"
minval="-1.7976931348623157E308" name="Var_2" sequenceNo="2">
```

```
        <label>Var_2</label>
        <description/>
        <value>0.0</value>
        <defaultValue>0.0</defaultValue>
    </itpamDouble>
    <itpamDate isReadOnly="false" name="Var_3" sequenceNo="3">
        <label>Var_3</label>
        <description/>
    </itpamDate>
    <itpamBoolean isReadOnly="false" name="Var_5" sequenceNo="5">
        <label>Var_5</label>
        <description/>
        <value>>false</value>
        <defaultValue>>false</defaultValue>
    </itpamBoolean>
</page>
<page name="User Prompt"/>
<page name="System"/>
</pages>          <page name="System"/>
</pages>
```

## The <attachments> Tag

The <attachments> tag provides meta information about attachment. CA Process Automation supports adding multiple attachments to the executeProcess and executeStartRequestForm methods.

### Example

```
<attachments attachmentsParamName="?">
  <attachment
    <attachmentID> </attachmentID>
    <name>?</name>
  </attachment>
```

<attachmentID> provides the content Id of attachment.

<name> specifies the name of attachment.

<attachmentsParamName = "paramName"> specifies the name of variable which appears in the process dataset of the executed process instance. The attachment can be accessed using the paramName name variable.

**Important!** When integrating into CA Process Automation using Web services, the Web service stubs generated using common third-party tools (such as wsdl2java) do not cleanly generate code for handling attachments. To integrate into CA Process Automation using Web services that manipulate attachments, see the "CA IT PAM Web Utilities" section of the "CA IT PAM Best practices" page. This page is linked from the CA IT PAM home page on [support.ca.com](http://support.ca.com).

## The <params> Tag

To provide a parameter, create one <params> tag. You can provide a list of parameters under the <params> tag in the request.

### Example

```
<params>
  <param name="vendorName">CA </param>
  <param name="quantity"> 4</param>
  <param name="start Date">10/30/2000</param>
  <param name="approvalRequired">false</param>
</params>
```

The name attribute specifies the name of the parameter.

*Value* specifies the value of the parameter in the value attribute.

- If the parameter exists in the dataset of the CA Process Automation object, the value of the parameter is adjusted with new value.
- If the parameter does not exist, a new variable is created with the name and value provided. The type of the variable depends upon the value provided.

## The <options> Tag

The <options> tag contains the scheduling parameters.

### Example

```
<options>
  <startDate> 2002-09-24</startDate>
  <startTime> 09:00:00 </startTime>
</options>
```

The process executes according to the start date and start time provided.



# Chapter 3: Web Services Methods

---

This section contains the following topics:

- [AsyncSoapResponse](#) (see page 17)
- [checkServerStatus](#) (see page 18)
- [checkStartRequestStatus](#) (see page 19)
- [controllInstance](#) (see page 21)
- [controlProcess](#) (see page 22)
- [deleteArchivedInstances](#) (see page 25)
- [deleteAttachments](#) (see page 26)
- [executePendingInteraction](#) (see page 27)
- [executeProcess](#) (see page 30)
- [executeStartRequest](#) (see page 33)
- [exportObject](#) (see page 37)
- [generateEvent](#) (see page 40)
- [getAttachments](#) (see page 42)
- [getITPamVersionInfo](#) (see page 44)
- [getMatchingEvents](#) (see page 45)
- [getPendingInteractionRequestForm](#) (see page 48)
- [getPendingUserInteractions](#) (see page 51)
- [getProcessLogs](#) (see page 58)
- [getProcessStatus](#) (see page 65)
- [getStartRequestForm](#) (see page 68)
- [getStartRequestForms](#) (see page 70)
- [ImportObject](#) (see page 74)

## AsyncSoapResponse

Use this method to complete an Invoke SOAP Method Async operator that is running in any process instance.

### Example

```
<AsyncSoapResponse xmlns="http://www.ca.com/itpam">  
<MessageID>514514e3-e8c8-4b1d-be42-3ee85e9d8d37</MessageID>  
<!-- Any valid XML fragment -->  
</AsyncSoapResponse>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<AsyncSoapResponse xmlns="http://www.ca.com/itpam">
<MessageID>514514e3-e8c8-4b1d-be42-3ee85e9d8d37</MessageID>
<!--Any valid XML fragment-->
</AsyncSoapResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

<MessageID> specifies the UUID of the Invoke SOAP Method Async operator that is running.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<AsyncSoapResponseResponse xmlns="http://www.ca.com/itpam">
<status>SOAP_ASYNC_SUCCESS</status>
</AsyncSoapResponseResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## checkServerStatus

Use this method to see the status of the server.

### Example

```
<checkServerStatus xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
</checkServerStatus>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body><checkServerStatus xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
</checkServerStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<checkServerStatusResponse xmlns="http://www.ca.com/itpam">
<serverStatus>Server status ok.</serverStatus>
</checkServerStatusResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<serverStatus>* shows the status of the server.

## checkStartRequestStatus

Use this method to see the status of a start request.

### Example

```
<checkStartRequestStatus xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<interactionId>51</interactionId>
</checkStartRequestStatus>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<checkStartRequestStatus xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<interactionId>51</interactionId>
<tagId>z34dsf5c7-xagb-4g5d-74g5-bd5c4we2378f</tagId>
</checkStartRequestStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

You can include one of the following:

*<interactionId>*: The user passes the ID of the start request form to view.

*<tagId>*: The user passes the tagId of the start request form to view.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<checkStartRequestStatusResponse xmlns="http://www.ca.com/itpam">
<startRequestStatus>
<interactionId>51</interactionId>
<state>Completed</state>
<earliest-start-time>2012-03-01T12:51:25.690+05:30</earliest-start-time>
<start-time>2012-03-01T12:51:25.893+05:30</start-time>
<end-time>2012-03-01T12:51:38.033+05:30</end-time>
<refProcess>/TENSU03/Process</refProcess>
<processID>52</processID>
</startRequestStatus>
</checkStartRequestStatusResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response contains tags under which the user can see the result.

`<state>` describes the state of the start request.

`<earliest-start-time>` states the time when the start request is queued.

`<start-time>` states the actual start time of the start request.

`<end-time>` states the end time of the start request.

`<refProcess>` shows the reference path of the process that is attached with the start request.

`<processID>` shows the Instance Id of the process (only returned if a process instance has been created).

## controlInstance

Use this method to create a request to control an instance.

controlInstance specifies archiving for those process/start request forms (and all its child instances) that are not triggered in detached mode. This method does not send a fault if a process/start request form is already marked for archiving. controlInstance sends a fault when a process/start request form is not found with the specified ID.

## SOAP Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:itp="http://www.ca.com/itpam">
  <soapenv:Header/>
  <soapenv:Body>
    <controlInstance>
      <auth>
        <user>Joe</ user >
        <password>thisismypassword</ password >
      </auth>
      <rootUUID>b939f5c7-7aeb-4a5f-8435-bfbc4c79ac8f</rootUUID>
      <instanceId>3186</instanceId>
      <tagId>Srf4</tagId>
      <action>archive</action>
    </controlInstance>
  </soapenv:Body>
</soapenv:Envelope>
```

<rootUUID> specifies the rootUUID of the process/start request form.

<instanceId> specifies the instance ID of the process/start request form.

<tagId> specifies the tagId of the process/start request form.

<action> specifies the action to perform on the process/start request form instance. Currently, only archive is supported.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <controlProcessResponse xmlns="http://www.ca.com/itpam">
      <actionStatus>The archive action was successfully executed.</actionStatus>
    </controlProcessResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## controlProcess

Use this method to create a request to control a process.

### Request

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Header/><SOAP-ENV:
Body><controlProcess xmlns="http://www.ca.com/itpam">
  <ProcessID>47</ProcessID>
  <action>suspend</action>
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
</controlProcess></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

## Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<controlProcessResponse xmlns="http://www.ca.com/itpam">
<actionStatus>The suspend action for Process ID "47" was queued.</actionStatus>
</controlProcessResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Actions

#### Suspend

You can suspend a single process instance through a web services method.

If you cannot suspend a process instance (for example, it already completed or aborted), an error message returns containing details about the request.

Here is an example of a fault when you try to suspend a completed process:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">
<faultcode>SOAP-ENV:Server</faultcode>
<faultstring>The process with Process ID "208" is in completed state and
therefore cannot be suspended.</faultstring>
</Fault>
```

#### Resume

You can resume a single process instance that was previously suspended through a web services method.

If you cannot resume a process instance, an error message returns containing details about the request. For example, a process instance cannot resume because it was already completed, aborted, or not currently suspended.

Here is an example of a fault when you try to resume a completed process:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">
<faultcode>SOAP-ENV:Server</faultcode>
<faultstring>The process with Process ID "208" is in completed state and
therefore cannot be resumed.</faultstring>
</Fault>
```

#### Abort

You can abort a single process instance through a web services method.

If you cannot abort a process instance (for example, it already completed or aborted), an error message returns containing details about the request.

Here is an example of a fault when you try to abort a completed process:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>The process with Process ID "486" is in completed state and
  therefore cannot be Aborted.</faultstring>
</Fault>
```

### Example 1

```
<controlProcess xmlns="http://www.ca.com/itpam">
  <ProcessID>ProcessID </ProcessID>
  <action>action</action>
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  <auth></controlProcess>
```

"*ProcessedID*" is the ROID of the process instance that you must control (suspend/resume/abort). The "ROID" of a running process is in the response when executing a process through web services.

### Example 2

```
<ExecuteC20FlowResponse xmlns="http://www.ca.com/itpam">
  <ExecuteC20FlowResult>
    <ROID>567</ROID>
    ...
    ...
  </ExecuteC20FlowResult>
</ExecuteC20FlowResponse>
```

The value for action is to suspend, resume, or abort. Only an authorized user can perform this action; otherwise, a fault is returned.

### Example 3

```
<SOAP-ENV:Fault xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>Invalid user/password or token.</faultstring>
</SOAP-ENV:Fault>
```

A fault returns if you try to perform an operation that you cannot perform on a process in its current state. For example, trying to suspend a process that already completed.



## Example 4

```
<SOAP-ENV:Fault xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>Process with Process Id "576" is in completed state therefore
cannot be suspended.</faultstring>
</SOAP-ENV:Fault>
```

## deleteArchivedInstances

Use this method to delete the archived instances of a CA Process Automation Orchestrator.

### Example

```
<deleteArchivedInstances xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<dateRange>
<fromDate>2012-02-02T02:00:00.320+05:30</fromDate>
<toDate>2012-02-02T03:00:00.320+05:30</toDate>
</dateRange>
</deleteArchivedInstances>
```

### SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/><SOAP-ENV:Body>
<deleteArchivedInstances xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<dateRange>
<fromDate>2012-02-02T02:00:00.320+05:30</fromDate>
<toDate>2012-02-02T03:00:00.320+05:30</toDate>
</dateRange>
</deleteArchivedInstances>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

You can specify the `<dateRange>` in the `<fromDate>` and `<toDate>` tags to delete archived instances. Use the Standard XSD format 2002-05-30T09:00:00.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<deleteArchivedInstancesResponse xmlns="http://www.ca.com/itpam">
<successMessage>4 archived instances were deleted successfully.</successMessage>
</deleteArchivedInstancesResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<successMessage>* shows the successful deletion of instances and the number of instances deleted.

When there are no instances in the given period, the response will be:

```
<deleteArchivedInstancesResponse xmlns="http://www.ca.com/itpam">
<successMessage>No archived instance is available for the specified date
range.</successMessage>
</deleteArchivedInstancesResponse>
```

## deleteAttachments

Use this method to delete the attachments which were uploaded using the executeStartRequest/ExecuteProcess method.

### Example

```
<deleteAttachments xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<attachments>
<attachmentId>111</attachmentId>
</attachments>
</deleteAttachments>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body><deleteAttachments xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<attachments>
<attachmentId>111</attachmentId>
</attachments>
</deleteAttachments>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<deleteAttachmentsResponse xmlns="http://www.ca.com/itpam">
<successMessage>The specified attachments were deleted
successfully.</successMessage>
</deleteAttachmentsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The `<successMessage>` tag shows results of the request.

## executePendingInteraction

Use this method to execute the pending user interactions.

## Example

```
<executePendingInteraction xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <params>
    <param name="Var_0">hello</param>
    <param name="Var_1">world</param>
  </params>
  <userInteractionID>22141</userInteractionID>
  <isApprove>true</isApprove>
</executePendingInteraction>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body><executePendingInteraction xmlns="http://www.ca.com/itpam">
    <auth>
      <user>Joe</ user >
      <password>thisismypassword</ password >
    </auth>
    <params>
      <param name="Var_0">hello</param>
      <param name="Var_1">world</param>
    </params>
    <userInteractionID>22141</userInteractionID>
    <isApprove>true</isApprove>
  </executePendingInteraction>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### To provide a parameter

In the previous request, the user can provide a list of parameters under the <params> tag. The user must create one <param> tag which includes the following parameters:

#### Name attribute of this tag

Provide the name of the parameter.

#### Value of this tag

Provide the value of the parameter.

*<userInteractionID>* is the ID of the Task Id of the user interaction execute.

*<isApprove>* specifies to approve or reject the task. This value can be true or false. This tag is optional.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<executePendingInteractionResponse xmlns="http://www.ca.com/itpam">
<userInteractionID>22141</userInteractionID>
<interactionTitle>Task_MyTasks</interactionTitle>
<state>Completed</state>
<startTime>2012-02-29T18:11:28.817+05:30</startTime>
<endTime>2012-02-29T18:17:27.707+05:30</endTime>
<parmittedUserGroup>pamadmin</parmittedUserGroup>
<closedBy>pamadmin</closedBy>
</executePendingInteractionResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<userInteractionID>* specifies the Id of the interaction request.

*<interactionTitle>* specifies the title of the interaction request.

*<state>* specifies the state of the interaction request.

*<startTime>* specifies the start time of the interaction request.

*<endTime>* specifies the end time of the interaction request.

*<parmittedUserGroup>* shows the list of assigned users and groups.

*<closedBy>* shows the name of the user who responded to the task.

**Note:** The "IsApprove flag" was being ignored for this method in CA Process Automation r2.1. To correct this issue, yet allow retention of the existing behavior for existing CA Process Automation instances, a new install has "oasis.reject.unnecessary.approval" set to true in the OasisConfig.properties configuration file. Existing installations have oasis.reject.unnecessary.approval added to set to the OasisConfig.properties configuration file and set to false.

If "oasis.reject.unnecessary.approval" is true, then the server considers the isApprove flag. A fault is returned if the user tries to approve or reject a form which does not require approval. A fault is also returned if the SOAP message does not include the isApprove flag for a form which requires approval.

## executeProcess

Use this method to execute a process.

### Example

```
<executeProcess xmlns="http://www.ca.com/itpam">
  <flow>
    <name>/MIMETEST</name>
    <action>start</action>
    <auth>
      <user>Joe</ user >
      <password>thisismypassword</ password >
    </auth>
    <params>
      <param name="name_6__">param__</param>
    </params>
    <options>
      <startDate>startDate__</startDate>
      <startTime>startTime__</startTime>
      <tagId> 06a4d113-0333-4aba-8cce-8781c18647c9</tagId>
      <isAutoArchive>true</isAutoArchive>
      <priority>priority__</priority>
    </options>
    <attachments attachmentsParamName="name">
      <attachment>
        <attachmentID>123</attachmentID>
        <name>MIMETEST</name>
        <localSourceLocation>"C:\\setupdir.log"</localSourceLocation>
      </attachment>
    </attachments>
  </flow>
</executeProcess>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<executeProcess xmlns="http://www.ca.com/itpam">
<flow>
<name>/MIMETEST</name>
<action>start</action>
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<params>
<param name="name_6__">param__</param>
</params>
<options>
<startDate>startDate__</startDate>
<startTime>startTime__</startTime>
<tagId> 06a4d113-0333-4aba-8cce-8781c18647c9</tagId>
<isAutoArchive>>true</isAutoArchive>
<priority>priority__</priority>
</options>
<attachments attachmentsParamName="TENSU03">
<attachment>
<attachmentID>123</attachmentID>
<name>MIMETEST</name>
<localSourceLocation>"C:\\setupdir.log"</localSourceLocation>
</attachment>
</attachments>
</flow>
</executeProcess>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following tags are under the <flow> tag:

- <name> specifies the name of the process including absolute path which the user wants to execute.
- <action> Specifies "start" to start the process.
- <tagId> Adds a unique ID to a process/start request form. A fault returns if there is a process or start request form instance with same tag. The <tagId> tag monitors the status of any process/start request form that you specify the tag for.
- <IsAutoArchive> tag can be set to false to skip process/Start Request Form instances (along with their child instances) from archiving under archival policy. Processes that are run in detached mode are not affected by this tag. The default value for <IsAutoArchive> is true.



## SOAP Response

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ExecuteC20FlowResponse xmlns="http://www.ca.com/itpam">
<ExecuteC20FlowResult>
<ROID>150</ROID>
<flow-name>/MIMETEST</flow-name>
<flow-action>start</flow-action>
<auth-user>pamadmin</auth-user>
<submission-time>2012-01-18T09:26:36.453+05:30</submission-time>
<sender-address>none</sender-address>
<connection-id>none</connection-id>
<message>Document accepted for processing</message>
<params>
<param name="RuntimeROID">150</param>
<param name="ServerId">1b4bda3d-bc5c-4a89-a2ad-23e2d4e71bce</param>
<param name="effectiveUser">pamadmin</param>
<param name="InstanceUUID">659322db-deca-4827-b622-14f69d09b33c</param>
<param name="name_6__">param__</param>
<param name="UserName">pamadmin</param>
<param name="ProcessName">/MIMETEST</param>
<param name="ServerName">lodivsa204.ca.com</param>
<param name="DisplayName">MIMETEST</param>
<param name="ProcessPriority">priority__</param>
<param name="TouchpointName">Orchestrator</param>
<param name="CallerUser">pamadmin</param>
<param name="ScheduledStartTime">2012-01-18 00:00:00</param>
<param name="rootUUID">659322db-deca-4827-b622-14f69d09b33c</param>
<param name="ProcessAction">start</param>
<param name="InstanceName">MIMETEST_150</param>
</params>
</ExecuteC20FlowResult>
</ExecuteC20FlowResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

In this response, the user can see the details of the instance of the process that executes.

<ROID> specifies the ROID of the instance of the process.

## executeStartRequest

Use this method to start a start request form from the CA Process Automation library.

## Example

```
<executeStartRequest xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <objLocation>
    <name>MYSRF</name>
    <path>/myfolder</path>
  </objLocation>
  <params>
    <param name="Var_0">joe</param>
    <param name="Var_1">smith</param>
  </params>
  <options>
    <startDate>startDate__</startDate>
    <startTime>startTime__</startTime>
    <tagId> 06a4d113-0333-4aba-8cce-8781c18647c9</tagId>
    <isAutoArchive>>true</isAutoArchive>
    <priority>priority__</priority>
  </options>
  <attachments attachmentsParamName="attachmentsParamName__">
    <attachment>
      <attachmentID>attachmentID__</attachmentID>
      <name>name_11__</name>
      <localSourceLocation>localSourceLocation__</localSourceLocation>
    </attachment>
  </attachments>
</executeStartRequest>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<executeStartRequest xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<objLocation>
<name>MYSRF</name>
<path>/TENSU03/</path>
</objLocation>
<params>
<param name="Var_0">user</param>
<param name="Var_1">joe</param>
</params>
<options>
<startDate>startDate__</startDate>
<startTime>startTime__</startTime>
<tagId>06a4d113-0333-4aba-8cce-8781c18647c9</tagId>
<isAutoArchive>true</isAutoArchive>
<priority>priority__</priority>
</options>
<attachments attachmentsParamName="attachmentsParamName__">
<attachment>
<attachmentID>attachmentID__</attachmentID>
<name>name_11__</name>
<localSourceLocation>localSourceLocation__</localSourceLocation>
</attachment>
</attachments>
</executeStartRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

`<objLocation>` specifies the location of the start request form in the CA Process Automation library. Two tags are under the `<objLocation>` tag.

`<name>` specifies the name of the start request form.

`<path>` provides the absolute path of the start request form object in the CA Process Automation Library.

`<tagId>` Adds a unique ID to a process/Start Request Form. A fault returns if there is a process or start request form instance with same tag. The `<tagId>` tag monitors the status of any process/start request form that you specify the tag for.

`<IsAutoArchive>` tag can be set to false to skip process/start request form instances (along with their child instances) from archiving under archival policy. Processes that are run in detached mode are not affected by this tag. The default value for `<IsAutoArchive>` is true.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<executeStartRequestResponse xmlns="http://www.ca.com/itpam">
<startRequestStatus>
<interactionId>114</interactionId>
<state>Running</state>
<earliest-start-time>2012-03-01T00:00:00.000+05:30</earliest-start-time>
<start-time>2012-03-01T13:54:13.523+05:30</start-time>
<refProcess>/TENSU03/Process</refProcess>
<params>
<param name="DisableSchedulingDialog">True</param>
<param name="FlowChartPath">/TENSU03/Process</param>
<param name="userId">pamadmin</param>
<param name="CALLER_TYPE">ITPAM_WS</param>
<param name="Var_1">tentu</param>
<param name="Var_0">sudhakar</param>
</params>
</startRequestStatus>
</executeStartRequestResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

`<interactionId>` specifies the interaction Id of the instance of start request form started.

`<state>` shows the state of the instance.

`<refProcess>` is the name of the process with the absolute path attached with the start request form.

## exportObject

Use this method to export a library using a Web service call.

### Example

```
<exportObject xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <folderName>/SOAPTest/ObjectsToExport</folderName>
  <level>2</level>
  <isAbsolute>true</isAbsolute>
  <downloadFileName>
    <downloadLocation>C:\</downloadLocation>
    [set the File Name variable]SOAPExportedObjects.xml</fileName>
  <overwriteFile>true</overwriteFile>
</downloadFileName>
  <filter>
    <exportObjectTypes>
      <objectType>Agenda</objectType>
      <objectType>Calendar</objectType>
      <objectType>CustomIcon</objectType>
      <objectType>CustomOperator</objectType>
      <objectType>CustomSensor</objectType>
      <objectType>Dataset</objectType>
      <objectType>InteractionRequestForm</objectType>
      <objectType>Package</objectType>
      <objectType>Process</objectType>
      <objectType>ProcessWatch</objectType>
      <objectType>Resources</objectType>
      <objectType>StartRequestForm</objectType>
      <objectType>StatePolicy</objectType>
      <objectType>System</objectType>
    </exportObjectTypes>
  </filter>
</exportObject>
```

## SOAP Request

The request contains:

- FolderName (the folder or object that you want to export)
- Folder recursion level
- Absolute/relative paths in export output
- Object types
- Download file name and the download location, where the exported library has to save.

**Note:** All parameters are optional with the exception of FolderName.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body><exportObject xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<folderName>/SOAPTest/ObjectsToExport</folderName>
<level>2</level>
<isAbsolute>true</isAbsolute>
<downloadFileName>
<downloadLocation>C:\</downloadLocation>
[set the File Name variable]SOAPExportedObjects.xml</fileName>
<overwriteFile>true</overwriteFile>
</downloadFileName>
<filter>
<exportObjectTypes>
<objectType>Agenda</objectType>
<objectType>Calendar</objectType>
<objectType>CustomIcon</objectType>
<objectType>CustomOperator</objectType>
<objectType>CustomSensor</objectType>
<objectType>Dataset</objectType>
<objectType>InteractionRequestForm</objectType>
<objectType>Package</objectType>
<objectType>Process</objectType>
<objectType>ProcessWatch</objectType>
<objectType>Resources</objectType>
<objectType>StartRequestForm</objectType>
<objectType>StatePolicy</objectType>
<objectType>System</objectType>
</exportObjectTypes>
</filter>
</exportObject>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ExportLibraryResponse xmlns="http://www.ca.com/itpam">
<exportLibraryResponse>Successfully uploaded the exported library to provided
download file name:C:\SOAPExportedObjects.xml</exportLibraryResponse>
<downloadedServer>PAM40-W2K8-17</downloadedServer>
</ExportLibraryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- If you do not provide the Download File Name in the request, then the serialized library (exported library object) sends it as an attachment to the response.
- If you do not provide an accessible shared location and file name in the request, then the error response is returned to the user.
- If you provide the accessible shared location and the Download File Name, then the exported library is saved at the given location with the specified file name. The shared location is returned in response.
- If a Download File Name that you provide exists in the shared location, then the error response is returned to the user.

## generateEvent

Use this method to post an event.

### Example

```
<generateEvent xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<eventName>test</eventName>
<eventType>eventType__</eventType>
<eventSource>testsource</eventSource>
<eventDestination>testdestination</eventDestination>
<eventExpDuration>130</eventExpDuration>
<payload>test</payload>
<params>
<param name="name__">param__</param>
</params>
<toSingleSubscriber>>false</toSingleSubscriber>
</generateEvent>
```



## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body><generateEvent xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<eventName>test</eventName>
<eventType>eventType__</eventType>
<eventSource>testsource</eventSource>
<eventDestination>testdestination</eventDestination>
<eventExpDuration>130</eventExpDuration>
<payload>test</payload>
<params>
<param name="name__">param__</param>
</params>
<toSingleSubscriber>>false</toSingleSubscriber>
</generateEvent>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<eventName>* identifies the name of the event (mandatory).

*<eventType>* identifies the type of event (optional).

*<eventSource>* identifies the source of the event (optional).

*<eventDestination>* identifies the destination of the event (optional).

*<eventExpDuration>* identifies the expiration duration of the event.

*<payload>* and *<params>* identify additional event data. If it is a single value, use *payload*; otherwise, use *params*.

*<toSingleSubscriber>* identifies whether the event can be delivered to single or multiple subscribers.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <generateEventResponse xmlns="http://www.ca.com/itpam">
      <event>
        <eventId>b1a3c218-0a7d-4667-9b7c-9eaddc7408</eventId>
        <eventName>test</eventName>
        <eventType>eventType__</eventType>
        <eventSource>testsource</eventSource>
        <eventDestination>testdestination</eventDestination>
        <payload>test</payload>
        <toSingleSubscriber>>false</toSingleSubscriber>
        <eventCreationTime>2012-03-01T14:05:14.692+05:30</eventCreationTime>
        <eventExpirationTime>2012-03-01T14:07:24.692+05:30</eventExpirationTime>
        <user>pamadmin</user>
      </event>
    </generateEventResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## getAttachments

Use this method to view attachments.

### Example

```
<getAttachments xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <filter>
    <contentId>111</contentId>
    <contentType>text/html</contentType>
  </filter>
</getAttachments>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body><getAttachments xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<filter>
<contentId>111</contentId>
<contentType>text/html</contentType>
</filter>

</getAttachments></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

*<filter>* tag filters the response that is based on the content ID and content type of the attachment (optional).

*<contentId>* specifies the content ID of the attachment (optional).

*<contentType>* specifies the content Type of the attachment (optional).

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getAttachmentsResponse xmlns="http://www.ca.com/itpam">
<attachments>
<attachment>
<attachmentID>1</attachmentID>
<name>test</name>
<contentId>111</contentId>
<contentType>text/html</contentType>
</attachment>
</attachments>
</getAttachmentsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In the response for every attachment found, there is an `<attachment>` tag.

For each attachment tag, there are the following tags:

`<attachmentID>` specifies the Attachment ID of the attachment.

`<name>` specifies the name of the attachment.

`<contentId>` specifies the content ID of the attachment.

`<contentType>` specifies the content type of the attachment.

## getITPamVersionInfo

Use this method to obtain CA Process Automation version details.

### Example

```
<getITPamVersion xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
</getITPamVersion>
```

### SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getITPamVersion xmlns="http://www.ca.com/itpam">
      <auth>
        <user>Joe</ user >
        <password>thisismypassword</ password >
      </auth>
    </getITPamVersion>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<servicePack> identifies the current CA Process Automation service pack (optional).

<patchNumber> identifies the current CA Process Automation patch number (optional).

<buildDetails> identifies any build details being used with the current CA Process Automation version (optional).

<majorVersion> identifies the current CA Process Automation version.

<minorVersion> identifies the current CA Process Automation minor version.

<buildNumber> identifies the current CA Process Automation build number.

<buildDate> identifies the current CA Process Automation build date.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ITPamVersionDetails xmlns="http://www.ca.com/itpam">
  <majorVersion>4</majorVersion>
  <minorVersion>0</minorVersion>
  <patchNumber>0</patchNumber>
  <buildNumber>335</buildNumber>
  <buildDate>2012-03-01T06:08:09.000+05:30</buildDate>
</ITPamVersionDetails>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<majorVersion> identifies the current CA Process Automation version.

<minorVersion> identifies the current CA Process Automation minor version.

<patchNumber> identifies the current CA Process Automation patch number (optional).

<buildNumber> identifies the current CA Process Automation build number.

<buildDate> identifies the current CA Process Automation build date.

<buildDetails> provides additional details about the build (optional).

## getMatchingEvents

Use this method to obtain the list of matching events for a given subscriber.

## Example

### Allow partial Match True

```
<getMatchingEvents xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<eventName>abc</eventName>
<eventType></eventType>
<eventSource></eventSource>
<eventDestination></eventDestination>
<eventExpression></eventExpression>
<enablePatternMatch></enablePatternMatch>
<allowPartialMatch>true</allowPartialMatch>
<consumeEvents></consumeEvents>
<retrieveAllMatchingEvents>true</retrieveAllMatchingEvents>
<uniqueId></uniqueId>
</getMatchingEvents>
```

### Enable Pattern Match True

```
<getMatchingEvents xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<eventName>a.*f</eventName>
<eventType></eventType>
<eventSource></eventSource>
<eventDestination></eventDestination>
<eventExpression></eventExpression>
<enablePatternMatch>true</enablePatternMatch>
<allowPartialMatch></allowPartialMatch>
<consumeEvents></consumeEvents>
<retrieveAllMatchingEvents>true</retrieveAllMatchingEvents>
<uniqueId></uniqueId>
</getMatchingEvents>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getMatchingEvents xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<eventName>abc</eventName>
<eventType/>
<eventSource/>
<eventDestination/>
<eventExpression/>
<enablePatternMatch/>
<allowPartialMatch/>
<consumeEvents/>
<retrieveAllMatchingEvents>true</retrieveAllMatchingEvents>
<uniqueId/>
</getMatchingEvents>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<eventName>* identifies the name of the event (optional).

*<eventType>* identifies the type of event (optional).

*<eventSource>* identifies the source of the event (optional).

*<eventDestination>* identifies the destination of the event (optional).

*<eventExpression>* is a Boolean expression for additional event parameters. This expression is evaluated against the payload or event parameters that are sent with an event (optional). For example:

```
Event.eventid==1234.
```

*<enablePatternMatch>* enables pattern matching for the parameters like event name, type, source and destination.

*<allowPartialMatch>* allows a partial match for the parameters like event name, type, source and destination.

<uniqueId> sends a unique ID with the request so that the SOAP call consumes the event only once. If the send event has an option to deliver to multiple subscribers, multiple calls of getMatchingEvents with same <uniqueId> consumes the event only once.

<consumeEvents> lets the request consume or not consume the matching events.

<retrieveAllMatchingEvents> lets the request retrieve all or first matched send event.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getMatchingEventsResponse xmlns="http://www.ca.com/itpam">
      <events>
        <event>
          <eventId>fefce4c1-ad8e-42a2-8964-362704ddb24</eventId>
          <eventName>abc</eventName>
          <eventType>eventType__</eventType>
          <eventSource>eventSource__</eventSource>
          <eventDestination>eventDestination__</eventDestination>
          <payload>payload__</payload>
          <toSingleSubscriber>>false</toSingleSubscriber>
          <eventCreationTime>2012-03-06T17:04:32.562+05:30</eventCreationTime>
          <eventExpirationTime>2012-03-06T17:06:42.562+05:30</eventExpirationTime>
          <user>pamadmin</user>
        </event>
      </events>
    </getMatchingEventsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## getPendingInteractionRequestForm

Use this method to retrieve information for a pending interaction request form that is based on the Task ID.

### Example

```
<getPendingInteractionRequestForm xmlns="http://www.ca.com/itpam"
  getApprovalRequired="false" getParamSequence="false">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <userInteractionID>924</userInteractionID>
</getPendingInteractionRequestForm>
```



## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getPendingInteractionRequestForm xmlns="http://www.ca.com/itpam"
getApprovalRequired="false" getParamSequence="false">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<userInteractionID>924</userInteractionID>
</getPendingInteractionRequestForm>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<userInteractionID>* specifies the Task ID of the pending interaction request form.

*<getPendingInteractionRequestForm>* retrieves the sequence number of the parameters and sets the *getParamSequence* attribute value to "true".

## SOAP Response

### With proper values:

```
<getPendingInteractionRequestFormResponse xmlns="http://www.ca.com/itpam">
  <processID>913</processID>
  <isApprovalRequired>true</isApprovalRequired>
  <description/>
  <pages hasComplexType="false">
    <page name="Parameters">
      <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_0"
        sequenceNo="0">
        <label>Var_0</label>
        <description/>
        <value/>
      </itpamString>
      <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_1"
        sequenceNo="1">
        <label>Var_1</label>
        <description/>
        <value/>
      </itpamString>
    </page>
    <page name="System"/>
    <page name="User Prompt"/>
  </pages>
</getPendingInteractionRequestFormResponse>
```

*<pages>* specifies the details of the pages in the interaction request form.

**When provided a task in taken state:**

```

<getPendingInteractionRequestFormResponse xmlns="http://www.ca.com/itpam">
  <processID>7</processID>
  <isApprovalRequired>true</isApprovalRequired>
  <description/>
  <pages hasComplexType="false">
    <page name="Parameters">
      <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_0"
        sequenceNo="0">
        <label>Var_0</label>
        <description/>
        <value/>
      </itpamString>
      <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_1"
        sequenceNo="1">
        <label>Var_1</label>
        <description/>
        <value/>
      </itpamString>
    </page>
    <page name="System"/>
    <page name="User Prompt"/>
  </pages>
</getPendingInteractionRequestFormResponse>

```

**When provided a task in completed state:**

```

<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>"UserInteraction with Task Id "<<taskid>>" is in completed state
  therefore this action is not allowed"
</faultstring>
</Fault>

```

**Note:** Starting with CA Process Automation r2.2, "getApprovalRequired" is added to "getPendingInteractionRequestForm" request. If it is set to true, then the "isApprovalRequired" flag returns, indicating if the form needs approval. To enable this behavior in CA Process Automation instances that were upgraded from r2.1, have your CA Process Automation Administrator perform the changes to the CA Process Automation configuration (described in the Note in "[executePendingInformation Method](#) (see page 27)").

## getPendingUserInteractions

Use this method to view all pending interactions that are assigned to the user.

## Return all interactions that are pending in the Orchestrator

### Inline text

```
<getPendingUserInteractions xmlns="http://www.ca.com/itpam" getRootUUID="false">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<rootUUID></rootUUID>
<processID></processID>
</getPendingUserInteractions>
```

### Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getPendingUserInteractions xmlns="http://www.ca.com/itpam" getRootUUID="false">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<rootUUID></rootUUID>
<processID></processID>
</getPendingUserInteractions>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Response**

```

<getPendingUserInteractionsResponse xmlns="http://www.ca.com/itpam">
  <userInteractions>
    <userInteraction id="116" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="117" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="118" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="119" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="120" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="121" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
  </userInteractions>
</getPendingUserInteractionsResponse>

```

**Return all pending interactions that are invoked from a start request form****Inline text**

Provide the rootUUID of the process that the start request form is calling:

```

<getPendingUserInteractions xmlns="http://www.ca.com/itpam"
  getRootUUID="false">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <rootUUID>076be822-7514-4f04-a333-3e8b73693a2b</rootUUID>
  <processID></processID>
</getPendingUserInteractions>

```

### Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getPendingUserInteractions xmlns="http://www.ca.com/itpam" getRootUUID="false">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<rootUUID>6e8faae6-77a4-4461-8feb-9e5ca4d572d5</rootUUID>
<processID></processID>
</getPendingUserInteractions>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Response**

```
<getPendingUserInteractionsResponse xmlns="http://www.ca.com/itpam">
  <userInteractions>
    <userInteraction id="207" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="208" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="209" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="210" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="211" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="212" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="213" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="214" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="215" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="216" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
  </userInteractions>
</getPendingUserInteractionsResponse>
```

## Parent process calling an attached child process

### Inline text

Provide the rootUUID of the parent process from the dataset:

```
<getPendingUserInteractions xmlns="http://www.ca.com/itpam"
  getRootUUID="false">
  <auth>
  <user>Joe</ user >
  <password>thisismypassword</ password >
  <auth>
  <rootUUID>6e8faae6-77a4-4461-8feb-9e5ca4d572d5</rootUUID>
  <processID></processID>
</getPendingUserInteractions>
```

### Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
  <getPendingUserInteractions xmlns="http://www.ca.com/itpam" getRootUUID="false">
  <auth>
  <user>Joe</ user >
  <password>thisismypassword</ password >
  <auth>
  <rootUUID>6e8faae6-77a4-4461-8feb-9e5ca4d572d5</rootUUID>
  <processID></processID>
  </getPendingUserInteractions>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



**Response**

```
<getPendingUserInteractionsResponse xmlns="http://www.ca.com/itpam">
  <userInteractions>
    <userInteraction id="176" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="177" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="178" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="179" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="180" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="181" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="182" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="183" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="184" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
    <userInteraction id="185" title="Task by B">
      <description>Task by B</description>
    </userInteraction>
  </userInteractions>
</getPendingUserInteractionsResponse>
```

*<userInteractions>* specifies the ID and title of every pending user interaction.

## getProcessLogs

Use this method to get the logs of a process instance.

### Message Levels

CA Process Automation process instance logs display the following message levels:

- Error =4
- Warning =3
- Notice =2
- Normal =1

### Message Categories

CA Process Automation process instances have the following message categories:

#### Process

Logs messages for a process instance.

#### Operator

Logs messages for operators in process.

#### Handler

Logs messages when lane change or exception handler is invoked.

#### Response

Logs messages when agents send a response back to the Orchestrator.

#### Custom

When you log messages without any category, these messages are grouped in the Custom category. You can provide the category as Custom to retrieve such messages.

### Custom Messages

You can define your own categories for logging. For example:

```
logEvent(3, "SD_MESSAGES", "this messages is logged for Service Desk operators  
"
```

To retrieve such messages, specify the category as SD\_MESSAGES in the request.

### To retrieve log messages

Use the following SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getProcessLogs xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
</auth>
<processID>72</processID>
<filter>
<level>2</level>
<categories>
<category>process</category>
<category>operator</category>
<category>handler</category>
<category>response</category>
<category> SD_MESSAGES </category>
</categories>
</filter>
</getProcessLogs>
```

`<level>` `</level>` can have a value of error, warning, notice, and normal.

If the user specifies the level as “normal”, all messages with a level of normal and above are retrieved.

When the error is at the top-most level and the user specifies the level as “error”, only messages with the “error” level are retrieved.

If the user specifies an invalid level, it is treated as normal (the lowest level).

To retrieve logs with multiple categories:

```
<categories>
<category>userMsg</category>
<category>process</category>
</categories>
```

In this example, all messages with the category “userMsg” and “process” are retrieved. Only log messages that match the level and one of the categories are returned.

## Example

```
<getProcessLogs xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <processID>72</processID>
  <filter>
    <level>2</level>
    <categories>
      <category>process</category>
      <category>operator</category>
      <category>handler</category>
      <category>response</category>
    </categories>
  </filter>
</getProcessLogs>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getProcessLogs xmlns="http://www.ca.com/itpam">
      <auth>
        <user>Joe</ user >
        <password>thisismypassword</ password >
      </auth>
      <processID>72</processID>
      <filter>
        <level>2</level>
        <categories>
          <category>process</category>
          <category>operator</category>
          <category>handler</category>
          <category>response</category>
        </categories>
      </filter>
    </getProcessLogs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getProcessLogsResponse xmlns="http://www.ca.com/itpam">
      <processLogs>
        <processlog category="Process" level="Notice">
          <logMessage>'LaneChangeHandler_30_72' instance was created.</logMessage>
          <time>2012-03-01T13:18:43.248+05:30</time>
        </processlog>
        <processlog category="Process" level="Notice">
          <logMessage>'LaneChangeHandler_30_72' is in 'Queued' state.</logMessage>
          <time>2012-03-01T13:18:43.248+05:30</time>
        </processlog>
        <processlog category="Process" level="Notice">
          <logMessage>Process started at '03/01/2012 13:18:43' by 'pamadmin'.</logMessage>
          <time>2012-03-01T13:18:43.592+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Start_Script_1' is enabled following 'Start_1'.</logMessage>
          <time>2012-03-01T13:18:43.639+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Start_1' is 'Completed' on 'Current Server'.</logMessage>
          <time>2012-03-01T13:18:43.639+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>A service request was sent for 'Start_Script_1'.</logMessage>
          <time>2012-03-01T13:18:44.123+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Start_Script_1' is 'Running' on 'Current Server'.</logMessage>
          <time>2012-03-01T13:18:44.139+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Start_Script_1' is 'Completed'.</logMessage>
          <time>2012-03-01T13:18:47.904+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Start_Script_1_1' is enabled following 'Start_Script_1'.</logMessage>
          <time>2012-03-01T13:18:47.935+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Lane_Change_1' was reset.</logMessage>
          <time>2012-03-01T13:18:47.982+05:30</time>
        </processlog>
        <processlog category="Operator" level="Notice">
          <logMessage>'Calculation_1' was reset.</logMessage>
          <time>2012-03-01T13:18:47.998+05:30</time>
        </processlog>
      </processLogs>
    </getProcessLogsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_2' was reset.</logMessage>
<time>2012-03-01T13:18:48.014+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' was reset.</logMessage>
<time>2012-03-01T13:18:48.029+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_3' was reset.</logMessage>
<time>2012-03-01T13:18:48.029+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' was reset.</logMessage>
<time>2012-03-01T13:18:48.045+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1' is enabled following 'Lane_Change_1'.</logMessage>
<time>2012-03-01T13:18:48.045+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_1' is 'Completed' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:48.060+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Calculation_1'.</logMessage>
<time>2012-03-01T13:18:48.092+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:48.092+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1' is 'Completed'.</logMessage>
<time>2012-03-01T13:18:48.576+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Start_Script_1_1'.</logMessage>
<time>2012-03-01T13:18:48.639+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_1' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:48.654+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_1' is 'Completed'.</logMessage>
<time>2012-03-01T13:18:49.060+05:30</time>
</processlog>
```

```
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_2' is enabled following
'Start_Script_1_1'.</logMessage>
<time>2012-03-01T13:18:49.076+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_1' was reset.</logMessage>
<time>2012-03-01T13:18:49.154+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1' was reset.</logMessage>
<time>2012-03-01T13:18:49.170+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_2' was reset.</logMessage>
<time>2012-03-01T13:18:49.201+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' was reset.</logMessage>
<time>2012-03-01T13:18:49.217+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_3' was reset.</logMessage>
<time>2012-03-01T13:18:49.217+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' was reset.</logMessage>
<time>2012-03-01T13:18:49.248+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' is enabled following 'Lane_Change_2'.</logMessage>
<time>2012-03-01T13:18:49.248+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_2' is 'Completed' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:49.248+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Calculation_1_1'.</logMessage>
<time>2012-03-01T13:18:49.279+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:49.295+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' is 'Completed'.</logMessage>
<time>2012-03-01T13:18:49.639+05:30</time>
</processlog>
```

```
<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Start_Script_1_2'.</logMessage>
<time>2012-03-01T13:18:49.701+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_2' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:49.701+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_2' is 'Completed'</logMessage>
<time>2012-03-01T13:18:50.123+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_3' is enabled following
'Start_Script_1_2'.</logMessage>
<time>2012-03-01T13:18:50.154+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_1' was reset.</logMessage>
<time>2012-03-01T13:18:50.217+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1' was reset.</logMessage>
<time>2012-03-01T13:18:50.248+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_2' was reset.</logMessage>
<time>2012-03-01T13:18:50.264+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_1' was reset.</logMessage>
<time>2012-03-01T13:18:50.279+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_3' was reset.</logMessage>
<time>2012-03-01T13:18:50.310+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' was reset.</logMessage>
<time>2012-03-01T13:18:50.326+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' is enabled following 'Lane_Change_3'.</logMessage>
<time>2012-03-01T13:18:50.342+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Lane_Change_3' is 'Completed' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:50.342+05:30</time>
</processlog>
```



```

<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Calculation_1_2'.</logMessage>
<time>2012-03-01T13:18:50.373+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:50.373+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Calculation_1_2' is 'Completed'</logMessage>
<time>2012-03-01T13:18:50.842+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>A service request was sent for 'Start_Script_1_3'.</logMessage>
<time>2012-03-01T13:18:51.107+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_3' is 'Running' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:51.123+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Start_Script_1_3' is 'Completed'</logMessage>
<time>2012-03-01T13:18:55.982+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Normal_Stop_1' is enabled following 'Start_Script_1_3'.</logMessage>
<time>2012-03-01T13:18:56.014+05:30</time>
</processlog>
<processlog category="Operator" level="Notice">
<logMessage>'Normal_Stop_1' is 'Completed' on 'Current Server'.</logMessage>
<time>2012-03-01T13:18:56.060+05:30</time>
</processlog>
<processlog category="Process" level="Notice">
<logMessage>Process is 'Completed'.</logMessage>
<time>2012-03-01T13:18:56.060+05:30</time>
</processlog>
</processLogs>
</getProcessLogsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## getProcessStatus

Use this method to view the state of a process instance.

## Example

```
<getProcessStatus xmlns="http://www.ca.com/itpam">
  <flow>
    <ROID>489</ROID>
    <action>check</action>
    <auth>
      <user>Joe</ user >
      <password>thisismypassword</ password >
    </auth>
  </flow>
</getProcessStatus>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getProcessStatus xmlns="http://www.ca.com/itpam">
      <flow>
        <ROID>489</ROID>
        <tagId>z34dsf5c7-xagb-4g5d-74g5-bd5c4we2378f</tagId>
        <action>check</action>
        <auth>
          <user>Joe</ user >
          <password>thisismypassword</ password >
        </auth>
      </flow>
    </getProcessStatus>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

You can include one of the following:

- `<ROID>` specifies the ROID of the process instance that you are viewing the state of.
- `<tagId>`: The user passes the tagId of the start request form to view.

`<action>` specifies the action to perform on the specified instance; the value of this tag must be "check".

## SOAP Responses

### Completed Process

```
<CheckFlowResponse xmlns="http://www.ca.com/itpam">
  <CheckFlowResult>
    <ROID>489</ROID>
    <flow-state>Completed</flow-state>
    <submission-time>2012-02-01T20:15:41.515+05:30</submission-time>
    <sender-address>none</sender-address>
    <connection-id>none</connection-id>
    <message>Flow state received</message>
    <params>
      <param name="RuntimeROID">489</param>
      <param name="Result">1</param>
      <param name="ServerId">b9c4b8de-b74b-44b5-b2d2-1deb6e3dc872</param>
      <param name="effectiveUser">pamadmin</param>
      <param name="InstanceUUID">f120811e-3895-41a2-abbb-3458c19a51c8</param>
      <param name="UserName">pamadmin</param>
      <param name="ProcessName">/SOAPTTest/P1</param>
      <param name="ServerName">lodivsa204.ca.com</param>
      <param name="DisplayName">P1</param>
      <param name="TouchpointName">Orchestrator</param>
      <param name="CallerUser">pamadmin</param>
      <param name="EndDate">2012-02-01 16:30:24</param>
      <param name="StartDate">2012-02-01 16:30:24</param>
      <param name="ScheduledStartTime">2012-02-01 00:00:00</param>
      <param name="rootUUID">f120811e-3895-41a2-abbb-3458c19a51c8</param>
      <param name="EndTime">2012-02-01 16:30:24</param>
      <param name="StartTime">2012-02-01 16:30:24</param>
      <param name="ProcessAction">start</param>
      <param name="InstanceName">P1_489</param>
    </params>
  </CheckFlowResult>
</CheckFlowResponse>
```

*<flow-state>* returns the state of the process instance.

### Nonexistent or Archived Process

The operator fails with a fault and returns this response:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>The instance does not exist. It may have been archived. Refresh the
  screen.</faultstring>
</Fault>
```

### Invalid Action

The operator fails with this fault:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">  
<faultcode>SOAP-ENV:Server</faultcode>  
<faultstring>Invalid flow action: c</faultstring>  
</Fault>
```

### Invalid ROID

The operator fails with this fault:

```
<Fault xmlns="http://schemas.xmlsoap.org/soap/envelope/">  
<faultcode>SOAP-ENV:Server</faultcode>  
<faultstring>error retrieving flow state</faultstring>  
</Fault>
```

## getStartRequestForm

Use this method to retrieve information about a start request form that is based on the name and location of the start request form from the CA Process Automation library.

### Example

```
<getStartRequestForm xmlns="http://www.ca.com/itpam" getParamSequence="true">  
<auth>  
<user>Joe</ user >  
<password>thisismypassword</ password >  
</auth>  
<objLocation>  
<name>MYSRF</name>  
<path>/myfolder</path>  
</objLocation>  
</getStartRequestForm>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/><SOAP-ENV:Body>
<getStartRequestForm xmlns="http://www.ca.com/itpam" getParamSequence="true">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<objLocation>
<name>MYSRF</name>
<path>/TENSU03/</path>
</objLocation>
</getStartRequestForm>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<name>* is the name of the start request form.

*<path>* is the absolute path of the folder where the start request form resides.

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getStartRequestFormResponse xmlns="http://www.ca.com/itpam">
      <refProcess>/Joe/Process</refProcess>
      <pages hasComplexType="false">
        <page name="Page">
          <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_0"
            sequenceNo="0">
            <label>Text Field</label>
            <description/>
            <value/>
          </itpamString>
          <itpamString isReadOnly="false" maxLength="2147483647" minLength="0" name="Var_1"
            sequenceNo="1">
            <label>Text Field</label>
            <description/>
            <value/>
          </itpamString>
        </page>
        <page name="System"/>
      </pages>
    </getStartRequestFormResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<refProcess>* shows the name and path of the process that is attached with this start request form.

## getStartRequestForms

Use this method to search the start request forms in a CA Process Automation library. A search is based on the path and keyword provided.

## Example

### With /root as the lookup path

```
<getStartRequestForms xmlns="http://www.ca.com/itpam">
  <auth>
    <user>Joe</ user >
    <password>thisismypassword</ password >
  </auth>
  <filter>
    <lookupPath isRecursive="true"></lookupPath>
    [set the value for HTML meta tag keywords]
    <keyword>ABC</keyword>
  </filter>
</getStartRequestForms>
```

### To obtain all the start request forms in the environment

Provide / and remove the Keyword\_\_ from the Keyword tags:

```
<getStartRequestFormsResponse xmlns="http://www.ca.com/itpam">
<startRequests>
<startRequest name="Start Purchase Request Form" refPath="/PAM Hardware
Procurement/">
<description/>
</startRequest>
<startRequest name="Asset_Install_SW_SRF" refPath="/sinra22/CA ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="Asset_MAC_HW_SRF" refPath="/sinra22/CA ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="Asset_Transfer_SRF" refPath="/sinra22/CA ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="Asset_Disposal_HW_SRF" refPath="/sinra22/CA ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="Asset_Install_HW_Transferred_SRF" refPath="/sinra22/CA
ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="Asset_Install_HW_SRF" refPath="/sinra22/CA ITAM/SRF/">
<description>Database Object</description>
</startRequest>
<startRequest name="~Start Request Form" refPath="/Package Demo/">
<description/>
</startRequest>
<startRequest name="~Start Request Form" refPath="/Lokesh/">
<description/>
</startRequest>
</startRequests>
</getStartRequestFormsResponse>
```



### Perform a keyword search

You can search start request forms with keywords that are associated with them. For example, you can perform a search to find if *SRF1* and *SRF2* are associated with the keyword "ABC" in the folder */myfolder*.

```
<getStartRequestForms xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<filter>
<lookUpPath isRecursive="true">/myfolder</lookUpPath>
[set the value for HTML meta tag keywords]
<keyword>ABC</keyword>
</keywords>
</filter>
</getStartRequestForms>
```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<getStartRequestForms xmlns="http://www.ca.com/itpam">
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<filter>
<lookUpPath isRecursive="true"/></lookUpPath>
[set the value for HTML meta tag keywords]
<keyword>ABC</keyword>
</keywords>
</filter>
</getStartRequestForms>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This tag is present in the `<filter>` tag:

- `<lookUpPath>` provides the path under which the user wants to search the start request forms. One attribute in this tag is recursive. If a user wants to search recursively under the provided path, then set this attribute true. If a user wants to search only under the provided path, set this attribute as false. Provide the complete path (as `/Folder/Folder1`) as the value of this tag in which user wants to search.

## SOAP Response

```
<getStartRequestFormsResponse xmlns="http://www.ca.com/itpam">
  <startRequests>
    <startRequest name="SRF1" refPath="/TENSU03/">
      <description/>
    </startRequest>
    <startRequest name="SRF2" refPath="/TENSU03/">
      <description/>
    </startRequest>
  </startRequests>
</getStartRequestFormsResponse>
```

For every start request form that fulfills the criteria for the search, there is a `<startRequest>` under the `<startRequests>` tag. This tag contains the name, reference path, and description of the start request form in the SOAP response.

## ImportObject

Use this method to import objects into a CA Process Automation Library.

### Example

```
<ImportObject xmlns="http://www.ca.com/itpam">
  <importFileParam>
    <auth>
      <user>Joe</ user >
      <password>thisismypassword</ password >
    </auth>
    <importLocation>importLocation__</importLocation>
    <localSourceLocation>localSourceLocation__</localSourceLocation>
    <isSetCurrent>isSetCurrent__</isSetCurrent>
    <isMakeAvailable>isMakeAvailable__</isMakeAvailable>
    <overwriteAction>overwriteAction__</overwriteAction>
  </importFileParam>
  <!-- This SOAP request can have MIME contents -->
</ImportObject>
```

`<isSetCurrent>` specifies whether to mark the imported objects as the current versions (optional).

`<isMakeAvailable>` specifies whether to mark imported custom operators as available (optional).

After you provide values in the inline text, create a MIME attachment for the file you specify in the `<localSourceLocation>` tag in the SOAP request. To add a MIME attachment, in the Properties panel under the MIME attachments, click Add, then provide appropriate values for:

#### Content Type

Type of content the MIME attachment is carrying. For example, text/xml.

#### Content ID

ID that the attachment is uniquely identified with. For example, 11111.

#### File URL

Path of the attachment. This path can be the path of the .xml to be imported. For example, C:\\SOAPExportedObjects.xml.

Save and run the process.

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ImportObject xmlns="http://www.ca.com/itpam">
<importFileParam>
<auth>
<user>Joe</ user >
<password>thisismypassword</ password >
<auth>
<importLocation></importLocation>
<localSourceLocation>"C:\\SOAPExportedObjects.xml"</localSourceLocation>
<isSetCurrent>true</isSetCurrent>
<isMakeAvailable>true</isMakeAvailable>
<overwriteAction>incrementObjectVersion</overwriteAction>
</importFileParam>
<!-- This request can have MIME contents -->
</ImportObject>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

`<importLocation>` specifies the location in the CA Process Automation library where you want to import the objects.

`<isSetCurrent>` specifies whether to mark the imported objects as the current versions (optional). The default value is false.

`<isMakeAvailable>` specifies whether to mark imported custom operators as available (optional). The default value is false.

#### `<overwriteAction>`

If an object with the same name already exists in the Library, you can select one of the following options for `<overwriteAction>`:

- `incrementObjectVersion` - Import as a new version and keep the existing object.
- `replaceObject` - Import and replace the existing object.
- `skipImport` - Do not import objects with the same name as an existing object.

For example:

```
<ImportObject xmlns="http://www.ca.com/itpam">
  <importFileParam>
    <auth>
      <user>Joe</ user >
      <password>thisismypassword</ password >
    </auth>
    <importLocation>/ABC</importLocation>
    <localSourceLocation>"C://abc.xml"</localSourceLocation>
    <isSetCurrent>true</isSetCurrent>
    <isMakeAvailable>true</isMakeAvailable>
    <overwriteAction>skipImport</overwriteAction>
  </importFileParam>
  <!-- This SOAP request can have MIME contents-->
</ImportObject>
```

Result:

```
<importFileResponse xmlns="http://www.ca.com/itpam">
  <messages>
    <message>Could not import object CustomIcon. Another object with the same name
as "CustomIcon" already exists.</message>
    <message>Could not import object P1. Another object with the same name as "P1"
already exists.</message>
    <message>Could not import object Process. Another object with the same name
as "Process" already exists.</message>
    <message>Could not import object Agenda. Another object with the same name
as "Agenda" already exists.</message>
    <message>Could not import object Resources. Another object with the same name
as "Resources" already exists.</message>
    <message>Could not import object SOAPMethods. Another object with the same
name as "SOAPMethods" already exists.</message>
    <message>Could not import object ProcessWatch. Another object with the same
name as "ProcessWatch" already exists.</message>
    <message>Could not import object Calendar. Another object with the same name
as "Calendar" already exists.</message>
    <message>Could not import object SOAPModule_NegativeCases. Another object
with the same name as "SOAPModule_NegativeCases" already exists.</message>
```

```
<message>Could not import object IRF. Another object with the same name as
"IRF" already exists.</message>
<message>Could not import object SOAPOperators_DynamicParams. Another object
with the same name as "SOAPOperators_DynamicParams" already
exists.</message>
<message>Could not import object Interaction Request Form. Another object
with the same name as "Interaction Request Form" already exists.</message>
<message>Could not import object SRF. Another object with the same name as
"SRF" already exists.</message>
<message>Could not import object SOAPTest_Trigger. Another object with the
same name as "SOAPTest_Trigger" already exists.</message>
<message>Could not import object CustomOperator. Another object with the same
name as "CustomOperator" already exists.</message>
<message>Could not import object Process_2. Another object with the same name
as "Process_2" already exists.</message>
<message>Could not import object Dataset. Another object with the same name
as "Dataset" already exists.</message>
<message>Could not import object Start Request Form. Another object with the
same name as "Start Request Form" already exists.</message>
<message>Could not import object GlobalVars. Another object with the same name
as "GlobalVars" already exists.</message>
<message>Could not import object Package. Another object with the same name
as "Package" already exists.</message>
</messages>
</importFileResponse>
```

## SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<importFileResponse xmlns="http://www.ca.com/itpam">
<successMessage>The specified object imported successfully.</successMessage>
</importFileResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*<successMessage>* displays a successful import message.



# Index

---

## <

- <attachments> tag
  - defined • 14
  - example • 14
- <options> tag
  - defined • 15
  - example • 15
- <page> and <pages> tags
  - defined • 12
  - example • 13
- <params> tag
  - defined • 15
  - example • 15

## A

- AsyncSoapResponse • 17
  - completing an asynchronous SOAP client call operator • 17
- authentication
  - username/password • 9

## C

- checkServerStatus • 18
- checkStartRequestStatus • 19
- controlInstance • 21
- controlProcess Web service method • 22

## D

- deleteArchivedInstances • 25

## E

- executePendingInteraction
  - defined • 27
- executeProcess • 30
- executeStartRequest • 33
- exportObject • 37

## G

- generateEvent • 40
- getAttachments • 42
- getITPamVersionInfo • 44
- getMatchingEvents • 45
- getPendingInteractionRequestForm • 48

- getPendingUserInteractions • 51
- getProcessLogs • 58
- getProcessStatus • 65
- getStartRequestForm • 68
- getStartRequestForms • 70

## I

- ImportObject • 74

## T

- tags
  - common for Web services methods • 11