Progress®



PAS for OpenEdge Administration Tools Reference

OpenEdge®

Copyright

© 2021 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Icenium, Inspec, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), SpeedScript, Stylus Studio, Stylized Design (Arrow/3D Box logo), Styleized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Habitat, Chef WorkStation, Corticon.js, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Everywhere, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, JustAssembly, JustDecompile, JustMock, KendoReact, NativeScript Sidekick, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Insight, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

July 2021

Product version: Progress OpenEdge 12.4

Updated: 2021/06/30

Table of Contents

Learn about PAS for OpenEdge administration tools.....11

TCMAN Reference.....13

Using TCMAN.....13
The tcman command.....14
Extending TCMAN.....16
Manager actions.....18
List deployed applications (list).....19
Display OS and server information (info).....20
Deploy a web application (deploy).....21
Undeploy a web application (undeploy).....23
Reload a web application (reload).....24
Display detailed server status (status).....25
Display memory leaks (leaks).....26
Start a web application (enable).....27
Stop a web application (disable).....28
Display global server resources (resources).....29
Display web application HTTP sessions (sessions).....30
Refresh agents in an ABL application (jmx refreshagents).....30
Export an ABL application (export).....31
Import an ABL application (import).....33
Refresh web handlers for an ABL web application (jmx refreshWeb).....34
Server actions.....35
Create an instance (create).....35
Delete an instance (delete).....37
Display and manage an instance's configuration (config).....38
Display or modify the server features of an instance (feature).....40
Clean up or archive server log files (clean).....42
Display server instances (instances).....43
List process ids (plist).....44
Register an instance for tracking (register).....45
Register and manage an instance as a Windows service (service).....47
Show Windows process information (showproc).....49
Start an instance (start).....50
Stop an instance (stop).....51
Display server, OS, and runtime version information (version).....52
Test a server configuration (test).....53
Stop tracking an instance (unregister).....54

Create a Tomcat worker configuration file (workers)....55
General actions....58
Display help (help)....58
Display runtime environment information (env)....59

REST API Reference for oemanager.war....61

Session management....62
Get session information....62
Get client information....64
Get current request information....65
Get configuration properties....67
Update configuration properties....69
Get runtime metrics....70
Reset runtime metrics....72
Cancel a request....73
Dump a session stack....74
Terminate a session....76
Terminate idle ABL sessions....77
Get dynamic session limit....78
Update dynamic session limit....80
Reset dynamic session limit....81
Agent management....83
List agents....83
Check ABL object tracking status....84
Enable or disable ABL object tracking....85
Get a report of tracked ABL objects....87
Get agent status information....89
Get connection information....90
Get request information....91
Get agent configuration properties....93
Update agent configuration properties....95
Stop a multi-session agent....96
Add a multi-session agent....98
Get session metrics....99
Refresh all ABL sessions of one or all agents....100
Flush deferred log buffer to agent log file....102
Reset deferred log buffer....104
Transport management....105
Get transport configuration properties....106
Update properties of a transport (SOAP/REST)....107
Enable or disable a transport (SOAP/REST)....112
Enable or disable a transport (APSV)....114
Get runtime metrics of a transport....115
Reset runtime metrics for a transport....119

List deployed services (SOAP/REST)....121
Get information about a service (SOAP/REST)....124
Get runtime properties of a service (SOAP)....126
Deploy a new service (REST)....127
Deploy a new service (SOAP)....129
Update runtime properties of a service (SOAP)....131
Reset runtime properties of a service (SOAP)....132
Get runtime metrics for a service (SOAP)....133
Reset runtime metrics for a service (SOAP)....135
Enable or disable a service (SOAP)....136
Remove a service (SOAP/REST)....137
Application management....139
List ABL application information....140
Get information about a specific ABL application....142
List web applications deployed to an ABL application....144
Get information about a specific web application....147

OEJMX Query Reference....151

Get agents query....152
Get active session information query....153
Get a stack trace for a session query....153
Get metrics for an ABL application query....155
Reset metrics for an ABL application query....155
Get hung client connections for an ABL application query....156
Get agent metrics query....157
Get agent session metrics query....158
Get active threads from Tomcat thread pool query....159
Get transport metrics for an ABL web application query....160
Reset transport metrics for an ABL web application query....163
Refresh agent query....166
Refresh all agents query....167
Flush deferred log buffer to agent log file query....168
Reset deferred log buffer query....168
Get dynamic ABL session limit query....169
Update dynamic ABL session limit query....170
Reset dynamic ABL session limit query....171
Enable or disable ABL object tracking query....171
Check ABL object tracking status query....172
Get a report of tracked ABL objects query....173

Preface

Purpose

The purpose of this content is to consolidate the reference materials required to manage a Progress Application Server for OpenEdge.

Audience

The audience includes developers managing test application servers and system administrators managing develop, test, staging and production application servers.

Organization

The organization of this content is the reference material for application server administration tools including TCMAN Reference and REST API documentation:

- TCMAN Reference on page 13
- REST API Reference for oemanager.war on page 61
- OEJMX Query Reference on page 151

To view a handy chart of the TCMAN reference guide, see the TCMAN Quick Reference Guide.

Documentation Conventions

See Documentation Conventions for an explanation of the terminology, format, and typographical conventions used throughout the OpenEdge content.

1

Learn about PAS for OpenEdge administration tools

The following topics comprise the reference material for PAS for OpenEdge administration tools including TCMAN Reference and REST API documentation.

- TCMAN Reference on page 13
- REST API Reference for oemanager.war on page 61
- OEJMX Query Reference on page 151

To view a handy chart of the TCMAN reference guide, see the TCMAN Quick Reference Guide.

2

TCMAN Reference

TCMAN is a command-line utility for managing and administering the Progress Application Server. TCMAN extends the basic Tomcat scripts for starting, stopping, and managing server instances.

This *TCMAN Reference* contains usage information for the `tcman` command as well as syntax information on all of the TCMAN actions.

For details, see the following topics:

- Using TCMAN
- Manager actions
- Server actions
- General actions

Using TCMAN

You use the TCMAN utility by running the `tcman` command, with specified actions and options, from either the core PAS (`$CATALINA_HOME/bin`) or from an instance (`$CATALINA_BASE/bin`). You can run `tcman` from a command line, or in a script. Both Windows (`tcman.bat`) and UNIX (`tcman.sh`) versions of TCMAN are implemented in the Progress Application Server.

See also

The tcman command on page 14
Extending TCMAN on page 16

Manager actions on page 18
Server actions on page 35
General actions on page 58

The tcman command

Purpose

TCMAN is a command-line utility for managing and administering PAS. On UNIX systems, you run the `tcman.sh` script followed by appropriate TCMAN actions and options. On Windows systems, you run the `tcman.bat` batch file, which is identical syntactically and functionally with `tcman.sh`.

Note: For the sake of brevity, all the syntax statements and examples in this reference show the `tcman.sh` script.

Syntax

```
{$CATALINA_HOME | $CATALINA_BASE}/bin/tcman.sh action [general_options]
[action_options ]
```

Parameters

`$CATALINA_HOME|$CATALINA_BASE`

Specify whether to run TCMAN from the root directory of the installed PAS (`$CATALINA_HOME`) or from the root directory of an instance (`$CATALINA_BASE`). The context of where you run TCMAN (whether from the `/bin` directory of the parent, or the `/bin` directory of an instance) affects which server the utility acts on.

Note: TCMAN automatically determines the value of `CATALINA_BASE` from the directory where you start it. When you run it from the `/bin` directory of an instance, the value of `CATALINA_BASE` is the root directory of the instance. If you run it from the `/bin` directory of the installed Progress Application Server, the value of `CATALINA_BASE` is the root directory of the installed server (which is the same value as `CATALINA_HOME`).

`action`

Specify which TCMAN action to invoke.

`general_options`

Specify one or more of the TCMAN common options that can apply to most actions. Note that one or more of the general options may be required by a specific action. For example, the `list` action requires `–u` in order to pass a user name and password.

The output of `tcman.sh help` *`action`* includes a list of general options that are applicable to a particular action.

The following table is a list of the common options:

Table 1: TCMAN general options

Common options	Function
`-u` *`user_name`*`:`*`password`*	Pass a valid user name and a password for HTTP Basic access authentication.
`-v`	Display verbose output.
`-M` *`URL`*	Override the default manager that manages Web applications by specifying the URL of an alternative manager. *`URL`* is expressed in the following format: `{http|https}://`*`host`*`:`*`port/manager_application`*
`-B`	Override default `CATALINA_BASE` environment settings.
`-n`	Debug the TCMAN action but do not execute changes.
`-I` *`instance_name`*	Run TCMAN from the `/bin` directory of the specified instance.

`action_options`

Specify an option that applies to the selected *`action`*. These options are explained in the topics that describe each action.

Example

Run the `help` action from the core server (`/psc/pashome`) to display a list of available TCMAN actions:

```
/psc/pashome/bin/tcman.sh help
usage:  tcman action [options...]
  manager actions:
      list        list deployed applications
      info        list server info
      deploy      deploy application
      undeploy    undeploy application
      reload      reload application
      status      show server status
      leaks       show server memory leaks
      enable      start web application running
      disable     stop running web application
      resources   list server global resources
      sessions    list a web application's sessions
  server actions:
      create      create a new server instance
      delete      delete server instance
      config      dump CATALINA_BASE configuration
      clean       clean/archive log files
      instances   list tracked server instances
      register    manually register an instance
      unregister  manually unregister an instance
      start       start this server
      stop        stop this server
      version     show the server version information
      test        test the server's configuration
  general actions:
      env         show tcman execution environment
      help        show this information
```

See also

Manager actions on page 18
Server actions on page 35
Extending TCMAN on page 16
General actions on page 58

Extending TCMAN

You can extend TCMAN by adding your own actions to the built-in actions of the TCMAN utility. You implement actions as UNIX shell scripts, Windows batch files, or Windows PowerShell scripts. Customizing TCMAN with your own actions allows you to unify all of your administrative tasks under a single utility.

File name format

Name your TCMAN action scripts using the following convention:

action_name`Cmd{.sh | .bat | .ps1}`

where file name extensions `.sh`, `.bat`, and `.ps1` are used for UNIX scripts, Windows batch files, and PowerShell scripts respectively.

UNIX action scripts

UNIX action scripts must be executable. That is, the file must have execution permissions for the current user. Each script must exit with a zero (0) status when successful, or a positive non-zero value to indicate an error.

When the action script is in the core server's bin directory (`$CATALINA_HOME/bin`), it is available to administrators of any instance. When the action script is in an instance's bin directory (`$CATALINA_BASE/bin`), it is only available to administrators of that instance.

The search order is:

1. `$CATALINA_BASE/bin`
2. `$CATALINA_HOME/bin`

Windows batch and PowerShell action scripts

On Windows, action scripts can be either DOS batch scripts (`.bat`) or PowerShell scripts (`.ps1`). Batch scripts should return zero (0) for success, or a positive non-zero integer to indicate an error level. Powershell scripts should return `\$true` for success or `\$false` to indicate an error.

When the action script is in the core server's bin directory (`$CATALINA_HOME/bin`), it is available to administrators of any instance. When the action script is in an instance's bin directory (`$CATALINA_BASE/bin`), it is only available to administrators of that instance.

If you have both batch and PowerShell scripts, TCMAN gives precedence to batch files. The search order is:

1. `$CATALINA_BASE/bin/`*`action_name`*`Cmd.bat`
2. `$CATALINA_BASE/bin/`*`action_name`*`Cmd.ps1`
3. `$CATALINA_HOME/bin/`*`action_name`*`Cmd.bat`
4. `$CATALINA_HOME/bin/`*`action_name`*`Cmd.ps1`

Command-line Syntax

You invoke an action script by specifying the *`action-name`* segment of a script's file name in a TCMAN command line.

```
tcman.sh action-name [general_options] [ action-name_options]
```

Note: You do not include the `Cmd.`*`extension`* portion of the file name on the TCMAN command line.

TCMAN first looks for action scripts in the instance (`$CATALINA_BASE/bin`) and then in the core server (`$CATALINA_HOME/bin`). When it finds the action script, TCMAN passes all command line options and parameters during execution.

Example

The following command line runs a script file named `viewCmd.sh` and passes the `-v` option and the `logs/catalina.out` argument.

```
tcman.sh view -v  logs/catalina.out
```

Notes

- When TCMAN does not recognize *action-name* as one of the built-in actions, it looks for an action script . If the specified action is not a built-in action or external action script it will exit with an error.
- When the external action script exits, TCMAN uses its return status to either issue an error or to exit with success. Therefore, extension action scripts should always return a specific value according to the platform type.
- Before executing an external action script, TCMAN exports these environment variables for consumption by the action script file:
  - JAVA_HOME : the root directory of the JAVA JDK (optional)
  - JRE_HOME : the root directory of the JAVA JRE (optional)
  - CATALINA_HOME : the root directory of where PAS was installed
  - CATALINA_BASE : the root directory of the PAS instance being managed (can be the same as CATALINA_HOME)
  - CATALINA_TMPDIR : the temporary file directory for temp files
  - CATALINA_PID : the file where the PAS process-id is stored
  - tcman_osshell : The name of the OS and command line shell TCMAN is running in
  - tcman_instancebase : the name of the new instance base directory when executing a create action
  - tcman_httpport : the HTTP network port command line option (`-p`)
  - tcman_httpsport : the HTTPS network port command line option (`-P`)
  - tcman_cfghttpport : the current instance's configured HTTP network port
  - tcman_cfghttpsport : the current instance's configuredHTTPS network port
  - tcman_cfgshutport : the current instance's shutdown network port (can be `-1` if there is no network shutdown port)
  - tcman_alias : the instance's alias name
  - tcman_type : the type of instance (home, service (on Windows), instance)
  - tcman_secmodel : the security model to apply (`default`, `production`, or `development`)
  - tcman_umask : the UNIX umask to use when creating new directories and files
  - tcman_verbose : the state of the TCMAN `-v` command line option
  - tcman_debug : The state of the TCMAN `-g` command line option

Manager actions

This section details the actions available for deploying, running, and monitoring web applications on a server instance.

See also

The tcman command on page 14

List deployed applications (list)

Purpose

Display all the web applications that are deployed on an instance.

Note: This command may be used whether the instance is online or offline. However, the output differs. When used offline, TCMAN simply shows a list of deployed application directories in the instance's web applications directory. When used online, it provides additional run-time details about the deployed web applications.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance if the instance is online. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh list [general_options] [-u user_id:password]
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help list` to see which general options are appropriate.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

Note: This option is required if the server is online. It is not required if the server is offline.

Example

Show the Web applications deployed to `acme1` when the instance is online:

```
/psc/acme1/bin/tcman.sh list -u tomcat:tomcat
OK - Listed applications for virtual host localhost
/:running:0:ROOT
/manager:running:4:manager
/oemanager:running:0:oemanager
/oeadapters:running:0:oeabl
```

Show the Web applications deployed to `acme1` when the instance is offline:

```
 /psc/acme1/bin/tcman.sh list
OK - Listing directories for /psc/acme1/webapps
/manager:stopped:0:manager
/oeadapters:stopped:0:oeabl
/oemanager:stopped:0:oemanager
/:stopped:0:ROOT
```

See also

Display web application HTTP sessions (sessions) on page 30
Deploy a web application (deploy) on page 21
Undeploy a web application (undeploy) on page 23
Start a web application (enable) on page 27
Stop a web application (disable) on page 28
The tcman command on page 14

Display OS and server information (info)

Purpose

Display server and OS information for a running instance.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Use the `test` action to show configuration information about a server that is not running.

Syntax

```
tcman.sh info [general_options]  -u user_name:password
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help info` to see which general options are appropriate.

`-u` *user_name*`:`*password*

Pass a valid user name and a password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

Example

Display the OS and server information for the running instance named `acme1`:

```
$: /psc/pashome/tcman.sh info -I acme1 -u tomcat:tomcat
OK - Server info
Tomcat Version: Apache Tomcat/7.0.42
OS Name: Linux
OS Version: 2.6.18-164.el5
OS Architecture: amd64
JVM Version: 1.7.0_02-b13
JVM Vendor: Oracle Corporation
```

See also

Display server, OS, and runtime version information (version) on page 52
Display detailed server status (status) on page 25
Test a server configuration (test) on page 53
The tcman command on page 14
Display or modify the server features of an instance (feature) on page 40
The tcman command on page 14

Deploy a web application (deploy)

Purpose

Deploy an ABL web application (`.war` file) to a PAS for OpenEdge instance, online (when the server is running) or offline (when the server is not running). TCMAN copies the web application to the server's web application directory.

Syntax

```
tcman.sh deploy [general_options] [-l] [-u user_id:password] [-a app_name]
war_file_path [abl_app_name]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help deploy` to see which general options are appropriate.

`-l`

Use this switch to deploy an ABL web application online without having to restart the instance. The switch loads the context necessary for the successful online deployment of a web application.

Note: The Tomcat manager (oemanager.war) must already be deployed for the `-l` switch to work. In the PAS for OpenEdge development server, the Tomcat manager is by default deployed in the default instance, `oepas1`. For all new instances in both the development and the production servers, you have to deploy the Tomcat manager first before you can use this switch. Also note that the `-l` switch can be used only in conjunction with the `-u` parameter.

`-u` *`user_id`*`:`*`password`*

Specify a valid user name and password for HTTP Basic access authentication.

Note: This parameter is required if the server is online. It is not required if the server is offline.

`-a` *`app _name`*

Specify a name for the web application. If you do not use this option, the application name will be the same as the `.war` file name.

`war_file_path`

Specify the location of the web application `.war` file that you want to deploy.

`abl_app_name`

It is the name of the ABL application under which you want to deploy your ABL web application.

An ABL application is a logical grouping of ABL web applications. It consists of a session manager, one or more multi-session agents, and one or more deployed ABL web applications.

An instance can contain more than one ABL application. You can deploy an ABL web application to either the default ABL application, which is named after the instance, or to a new ABL application. If you want to deploy an ABL web application under a new ABL application, then specify a name for the ABL application here.

Example

Deploy and rename `oeabl.war` (a web application that implements OpenEdge adapters) to the `acme1` instance of the core `pashome` server online:

```
/psc/acme1/bin/tcman.sh deploy -l -u tomcat:tomcat -a oeadapters
/psc/pashome/extras/oeabl.war
OK - deployed /psc/pashome/extras/oeabl.war to local directory /psc/acme1/webapps
```

Note: The `$CATALINA_HOME/extras` directory (`/psc/pashome/extras` in the example above) also contains number of instance management applications, including `host-manager.war`, `manager.war`, and `oemanager.war`.

See also

List deployed applications (list) on page 19
Undeploy a web application (undeploy) on page 23
Reload a web application (reload) on page 24
Start a web application (enable) on page 27

Stop a web application (disable) on page 28
The tcman command on page 14

Undeploy a web application (undeploy)

Purpose

Remove a Web application from running (online) or stopped (offline) instances. If the instance's autodeploy option is off, you must stop and restart a running server to complete removal. Note that the autodeploy option is set in the `.../conf/appserver.properties` file and is off by default.

Syntax

```
tcman.sh undeploy [general_options] [-u user_id:password]  app _name
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help undeploy` to see which general options are appropriate.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.) This option is required if you are accessing an online instance.

app_name

Specify the name of the web application to remove.

Example

Remove the oemanager application from the `acme1` instance:

```
/psc/acme1/bin/tcman.sh undeploy -u tomcat:tomcat oemanager
              OK - Undeployed application at context path /oemanager
```

See also

Deploy a web application (deploy) on page 21
List deployed applications (list) on page 19
Reload a web application (reload) on page 24
The tcman command on page 14

Reload a web application (reload)

Purpose

Restart a deployed, running Web application so that the application can pick up changes to its classes or libraries.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Note: The reload action does not reload the web application's `web.xml` file. To begin using changes to `web.xml`, you must stop and restart the web application.

Syntax

```
tcman.sh reload [general_options] -u user_id:password app_name
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help reload` to see which general options are appropriate.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

Note: This option is required if the server in online. It is not required if the server is offline.

app_name

Specify the name of the web application to restart.

Example

Reload the `oemanager` web application running on the `acme1` instance:

```
 /psc/acme1/bin tcman.sh reload  -u tomcat:tomcat oemanager
OK - Reloaded application at context path /oemanager
```

See also

Start a web application (enable) on page 27
Stop a web application (disable) on page 28
List deployed applications (list) on page 19
The tcman command on page 14

Display detailed server status (status)

Purpose

List information from the core server's memory, including web application statistics. Information includes memory pool usage, connector thread status, and connector status. Output is in XML format. (Note that redirecting the output to an XML viewer makes it more readable.)

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh status [general_options]  -u user_name:password [-f]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help status` to see which general options are appropriate.

`-u` *user_name*`:`*password*

Pass a valid user name and a password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

`-f`

Return full status information.

Example

Display core server's memory and web application statistics and use `xmllint` to format for readability:

```
$: tcman.sh status -u tomcat:tomcat | xmllint --format -
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/manager/xform.xsl" ?>
<status>
  <jvm>
    <memory free="453196832" total="520028160" max="1051394048"/>
    <memorypool name="PS Eden Space" type="Heap memory" usageInit="50331648"
usageCommitted="48758784" usageMax="55967744" usageUsed="1525560"/>
    <memorypool name="PS Old Gen" type="Heap memory" usageInit="469762048"
usageCommitted="469762048" usageMax="1006632960" usageUsed="63861584"/>
    <memorypool name="PS Survivor Space" type="Heap memory" usageInit="8388608"
usageCommitted="1507328" usageMax="1507328" usageUsed="1444184"/>
    <memorypool name="Code Cache" type="Non-heap memory" usageInit="2555904"
usageCommitted="3407872" usageMax="50331648" usageUsed="3303104"/>
    <memorypool name="PS Perm Gen" type="Non-heap memory" usageInit="67108864"
usageCommitted="67108864" usageMax="134217728" usageUsed="47406400"/>
  </jvm>
  <connector name="&quot;http-bio-8601&quot;">
    <threadInfo maxThreads="150" currentThreadCount="0" currentThreadsBusy="0"/>
    <requestInfo maxTime="0" processingTime="0" requestCount="0" errorCount="0"
bytesReceived="0" bytesSent="0"/>
    <workers/>
  </connector>
  <connector name="&quot;http-bio-8501&quot;">
    <threadInfo maxThreads="300" currentThreadCount="10" currentThreadsBusy="1"/>
    <requestInfo maxTime="2008" processingTime="2116" requestCount="10" errorCount="0"
 bytesReceived="0" bytesSent="5838"/>
    <workers>
      <worker stage="S" requestProcessingTime="2" requestBytesSent="0"
requestBytesReceived="0" remoteAddr="127.0.0.1" virtualHost="localhost" method="GET"
currentUri="/manager/status" currentQueryString="XML=true" protocol="HTTP/1.1"/>
    </workers>
  </connector>
</status>
```

See also

Display OS and server information (info) on page 20
Display memory leaks (leaks) on page 26
The tcman command on page 14

Display memory leaks (leaks)

Purpose

List Web applications with potential memory leaks.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh leaks [general_options]  -u user_name:password
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help leaks` to see which general options are appropriate.

`-u` *user_name*`:`*password*

Pass a valid user name and a password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

Example

Display memory leaks for web applications deployed on the `acme1` server instance:

```
/psc/acme1/bin/tcman.sh leaks -u tomcat:tomcat
OK - Found potential memory leaks in the following applications:
/warehouse
```

See also

List deployed applications (list) on page 19
The tcman command on page 14

Start a web application (enable)

Purpose

Start a web application that is deployed but not running.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh enable [general_options] -u user_id:password app_name
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help start` to see which general options are appropriate.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

app_name

Specify the name of the web application to start.

Note: To start the ROOT web application, you can specify `/` or `ROOT`.

Example

Start the `oeabl` application deployed on the `acme1` instance:

```
tcman.sh enable -u tomcat:tomcat oeabl
OK - Started application at context path /oeabl
```

See also

Stop a web application (disable) on page 28
The tcman command on page 14
The tcman command on page 14

Stop a web application (disable)

Purpose

Stop a running Web application.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh disable [general_options] [-u user_id:password] app_name
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help disable` to see which general options are appropriate.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

app_name

Specify the name of the web application to disable.

Note: To disable the ROOT web application, you can specify `/` or `ROOT`.

Example title

Disable the `oeabl` application running on the `acme1` instance:

```
/psc/acme1/bin/tcman.sh disable -u tomcat:tomcat oeabl
OK - Stopped application at context path /oeabl
```

See also

Start a web application (enable) on page 27
The tcman command on page 14

Display global server resources (resources)

Purpose

List the global resources used by the core server.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh resources [general_options]  -u user_name:password
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help resources` to see which general options are appropriate.

`-u` *user_name*`:`*password*

Pass a valid user name and a password for HTTP Basic access authentication.

(The default is `-u tomcat:tomcat`.)

Example

Display global resources for the running instance, `acme1`:

```
$: /psc/acme1/bin/tcman.sh resources -u tomcat:tomcat
OK - Listed global resources of all types
ServiceRegistry/ServiceRegistryFactory:com.progress.appserv.services.naming.ServiceRegistry
UserDatabase:org.apache.catalina.users.MemoryUserDatabase
```

Display web application HTTP sessions (sessions)

Purpose

Display how many sessions are active for the specified Web application, categorized by their duration.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.sh sessions [general_options] -u user_id:password app_name
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action.

`-u` *user_id*`:`*password*

Specify a valid user name and password for HTTP Basic access authentication. (The default is `-u tomcat:tomcat`.)

app_name

Specify the name of the web application to analyze for session information.

Example

Show the active sessions for the `manager` application deployed on the `acme1` instance:

```
/psc/acme1/bin/tcman.sh sessions -u tomcat:tomcat manager
OK - Session information for application at context path /manager
Default maximum session inactive interval 30 minutes
<1 minutes: 1 sessions
8 - <9 minutes: 2 sessions
9 - <10 minutes: 1 sessions
```

See also

List deployed applications (list) on page 19

Refresh agents in an ABL application (jmx refreshagents)

Purpose

Refresh Agents terminates all sessions within an ABL application, thereby refreshing the multi-session agent or agents so that changes to the application can be picked-up by the agent or agents.

To use this action, the Tomcat manager (`manager.war`) must be deployed on the instance and the instance must be running. You can deploy `manager.war` from `$CATALINA_HOME/extras`.

Syntax

```
tcman.{sh|bat} jmx refreshagents -appname application-name [-v]
```

Parameters

`-appname`

Specify the ABL application running in the PAS for OpenEdge instance.

`-v`

Provides a verbose output.

Example

Refresh agents in the running instance, `oepas1`:

```
tcman jmx refreshagents -appname oepas1 -v
info: TCManager 4.1.0
{
 "refreshAgents":{
  "agents":[
   {
    "agentId":"LdS8phtPR0GIrhRrrSL3Xg",
    "pid":"7672",
    "sessions":[
     {
      "sessionId":4,
      "terminateResult":"success"
     },
     {
      "sessionId":7,
      "terminateResult":"success"
     }
    ]
   }
  ]
 }
}
```

Export an ABL application (export)

Purpose

Exports an ABL application from an existing PAS for OpenEdge instance into an OpenEdge Archive (`.oear`) file.

Syntax

```
tcman.sh export -appName ABL_app_name [general_options] [-archiveBaseName
name_of_Archive] [-archiveClassifier classifier_name] [-archiveVersion
archive_version] [-destDir /path_of_archive/] [-instanceDir
/path_of_instance/] [-full] [-help]
```

Parameters

`-appName` *`ABL_app _name`*

The name of the ABL Application that will be exported into an OpenEdge Archive (`.oear`) file. An ABL application is a logical grouping of ABL web applications. It consists of a session manager, one or more multi-session agents, and one or more deployed ABL web applications.

`general_options`

Specify one or more of the general TCMAN options. Run `tcman.sh help export` to see which general options are appropriate.

`-archiveBaseName`

Base name of the `.oear` file. The default is the specified ABL application name

`-archiveClassifier`

Classifier used in archive name.

`-archiveVersion`

Version used in manifest and archive name.

`-destDir`

The output directory for the `.oear` file. The default is the current directory.

`-instanceDir`

Location of the instance. The default is `$CATALINA_BASE`.

`-full`

Specifies a full export of oeabl.war-based web applications as WAR files. If present, the oeabl.war-based web applications are exported in full as WAR files (for example, ROOT.war). Otherwise, the `tcman export` script runs a diff of an oeabl.war-based web application against oeabl.war and only includes files modified or added into a zip file. When the OEAR package is deployed to a new instance, the `tcman import` script applies

`-help`

Prints the help page for the command.

Example

Export the ABL application `oepas1` as an `.oear` file:

```
/oepas1/bin/tcman.sh export -appName oepas1
BUILD SUCCESSFUL
Total time: 1 second
OK - executed ANT target export in build file /users/devp/user1/oepas1/temp/export.xml

jar -xvf oepas1.oear
 inflated: conf/MANIFEST.MF
 inflated: tlr/merge.properties
 inflated: conf/oeablSecurity.properties
 inflated: webapps/ROOT.war
```

Import an ABL application (import)

Purpose

Imports an existing OpenEdge Archive (`.oear`) file into a PAS for OpenEdge instance.

Syntax

```
tcman.sh import [general_options] [-v] [-g] [-h] [-help] /path-to-archive/
```

Parameters

/path-to-archive/

Specify the path to the OpenEdge Archive (`.oear`) file.

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help import` to see which general options are appropriate.

`-v`

Output verbose execution information.

`-g`

Output debug execution information.

-h

Print usage information.

-help

Print usage information.

Example

Import the OpenEdge Archive file (`oepas1.oear`) into the PAS for OpenEdge instance (`devinst122`):

```
proenv>importInstance\bin\tcman import .\oepas1.oear
Importing archive C:\OpenEdge\WRK122\oepas1.oear
Buildfile: C:\OpenEdge\WRK122\importInstance\temp\appimport\tlr\build.xml

ablapp_deploy:

BUILD SUCCESSFUL
Total time: 0 seconds
OK - deployed C:\Progress\OPENED~2\servers\pasoe\extras\oeabl.war to local directory
C:\OpenEdge\WRK122\importInstance\webapps as application ROOT
Buildfile: C:\OpenEdge\WRK122\importInstance\ablapps\oepas1\tlr\build.xml

ablapp_deployed:

BUILD SUCCESSFUL
Total time: 0 seconds
Successfully imported oepas1 to C:\OpenEdge\WRK122\importInstance
```

Refresh web handlers for an ABL web application (jmx refreshWeb)

Purpose

The JMX-based refresh web handlers TCMAN command, `jmx refreshWeb`, reloads the web handler context to a given OEABL web application. This API dynamically reloads the web transport context and makes web services available without restarting the PAS for OpenEdge instance.

For more information, see 'Deploy web handler services' in *Manage Progress Application Server (PAS) for OpenEdge*.

Syntax

```
tcman.{sh|bat} jmx refreshWeb -appname application-name -webappname webapp-name
[-v]
```

Parameters

-appname

Specify the ABL application running in the PAS for OpenEdge instance. The default ABL application has the same name as the PAS for OpenEdge instance (for example, `oepas1`).

-webappname

Specify the ABL web application running in the PAS for OpenEdge instance. The default ABL web application deployed to an instance is `ROOT`.

`-v`

Provides a verbose output.

Example

This example shows the response of running the refresh web handlers command in the running instance, `oepas1`:

```
tcman.sh jmx refreshWeb -appname oepas1 -webapp ROOT -v
{
   "refreshWebHandlers":{
      "handlers":[
         {
            "service":"",
            "type":"DEFAULT",
            "class":"OpenEdge.Web.CompatibilityHandler",
            "enabled":true,
            "url":"/"
         }
      ],
      "operation":"REFRESH WEB HANDLERS",
      "outcome":"SUCCESS"
   }
}
```

Server actions

This section details the actions available for creating and monitoring server instances.

See also

The tcman command on page 14

Create an instance (create)

Purpose

Create a new instance of PAS for OpenEdge by running this action from `/bin` directory of the PAS for OpenEdge installation (`$CATALINA_HOME/bin/tcman.sh create`).

Syntax

```
tcman.sh create [general_options] [-f] [-p port_num] [-P port_num]
   [-s port_num] [-j port_num] [-m uid:pwd] [-N instance_name]
   [-U user_id ] [-G group_id] [-Z {prod | dev}] base_path
```

Parameters

`general_options`

Specify one or more of the general TCMAN options. Run `tcman.sh help create` to see which general options are appropriate.

`-f`

Copy all deployed web application archives (`.war` files) from `$CATALINA_HOME` to the new instance.

`-p` *`port_num`*

Specify the TCP port that listens for HTTP messages. The default is 8080.

`-P` *`port_num`*

Specify the TCP port that listens for HTTPS messages. The default is 8443.

`-s` *`port_num`*

Specify the TCP port to use to stop an instance. On Windows systems, you must specify a shutdown port. On UNIX, shutdown ports are optional.

`-j` *`port_num`*

Specify the TCP port that listens for AJP13 messages (an Apache protocol for handling requests from a web server to an application server). The default is 8009.

`-m` *`uid`*`:`*`pwd`*

Specify a user name and password that will be required to access Tomcat container-level security, which includes the `manager` and `oemanager` web applications. Replaces the defaults (`tomcat`:`tomcat`) in `/conf/tomcat-users.xml`.

`-N` *`instance_alias`*

Specify an alias for the instance. If you do not specify an alias, the instance name will be the name of the directory where the instance is created.

Note:

All instances are automatically registered for tracking when they are created. However, for tracking to function, the instance name must not contain spaces or any of the following characters: `"[ . # | ] $ ? + = { / , }"`

`-U` *`user_id`*

Specify the user-id of the owner of all the files and directories of the instance. The default is the user-id of the current process. `-G` must be specified if you use this option.

`-G` *`group_id`*

Specify the group-id of the owner of all the files and directories of the instance. The default is the group-id of the current process. `-U` must be specified if you use this option.

`-Z` **{**`dev` **|** `prod`**}**

Specify the security model of the instance to development (`dev`) or secure (`prod`).

A typical use of this option is for testing web applications in a secure server environment before packaging and deploying.

Note: The `-Z prod` option does not create a production server. To actually create a production server, you must have a production server license.

base_path

Specify the pathname where you will create the instance.

Example

Create an instance of `/psc/pashome` in `/psc/acme1`:

```
$: /psc/pashome/bin/tcman.sh create -p 8501 -P 8601 -s 8701 /psc/acme1
Server instance acme1 created at /psc/acme1
```

See also

Delete an instance (delete) on page 37
Display and manage an instance's configuration (config) on page 38
Register an instance for tracking (register) on page 45
Start an instance (start) on page 50
Stop an instance (stop) on page 51

Delete an instance (delete)

Purpose

Remove the directory tree and all of the files in an instance. Alias tracking is disabled for servers that are removed.

To execute this action, the instance cannot be running.

Note: You cannot recover any files or directories removed by the `delete` action. Backup anything you want to save before launching this action.

Also note that you cannot use delete to remove the installed, root server (`$CATALINA_HOME`).

Syntax

```
tcman.sh delete [general_options] [-y][base_path|alias_name]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help delete` to see which general options are appropriate.

`-y`

Delete everything without prompting for confirmation.

`base_path`

Specify the pathname of the instance that you intend to delete.

`alias_name`

Refer to the instance that you intend to delete by its alias rather than its pathname.

Example

Delete the instance of `pashome` that was created in `/psc/acme3`:

```
$: /psc/pashome/bin/tcman.sh delete /psc/acme3
The following directory tree will be removed permanently:
 ( WARNING all deployed web applications will be DELETED!! )
/PAS/wrkdir/acme3
/PAS/wrkdir/acme3/conf
/PAS/wrkdir/acme3/temp
/PAS/wrkdir/acme3/common
/PAS/wrkdir/acme3/common/lib
/PAS/wrkdir/acme3/logs
/PAS/wrkdir/acme3/webapps
/PAS/wrkdir/acme3/webapps/ROOT
/PAS/wrkdir/acme3/webapps/ROOT/static
/PAS/wrkdir/acme3/webapps/ROOT/static/error
/PAS/wrkdir/acme3/webapps/ROOT/static/auth
/PAS/wrkdir/acme3/webapps/ROOT/META-INF
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/adapters
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/adapters/rest/PingService
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/adapters/soap
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/classes
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/classes/com
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/classes/com/progress
/PAS/wrkdir/acme3/webapps/ROOT/WEB-INF/classes/com/progress/appserv
/PAS/wrkdir/acme3/work
/PAS/wrkdir/acme3/bin
Type 'yes' to continue
yes
Delete operation complete
server  removed at /PAS/wrkdir/acme3
```

See also

Create an instance (create) on page 35
Display server instances (instances) on page 43
The tcman command on page 14

Display and manage an instance's configuration (config)

Purpose

View, add, update, or delete the property values specified in `../conf/appserver.properties` and in `../conf/catalina.properties`.

When you run `tcman.sh config` with no parameters, it displays the core Tomcat server's configuration, and all the properties in both `.../conf/appserver.properties` and `.../conf/jvm.properties`. Note, however, that you can only view `jvm.properties`. You cannot modify its contents with the `config` action.

Syntax

```
tcman.sh config [general_options]
[prop_name | prop_name=value | +prop_name=value | ~prop_name]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help config` to see which general options are appropriate.

prop_name

Display the specified property and its value.

prop_name=value

Set the value of a property that exists in `.../conf/appserver.properties`.

+prop_name=value

Add a new property to `.../conf/appserver.properties` and set its value.

~prop_name

Remove the specified property from `.../conf/appserver.properties`.

Examples

Show the configuration and properties of `acme1`, an instance of the core server, `pashome`:

```
$: /psc/acme1/bin/tcman.sh config
Using CATALINA_BASE:   /psc/acme1
Using CATALINA_HOME:   /psc/pashome
Using CATALINA_TMPDIR: /psc/acme1/temp
Using JRE_HOME:        /tools/linuxx86_64/java64/jdk1.8.0_101
Using CLASSPATH:       /psc/pashome/bin/bootstrap.jar:/psc/pashome/bin/tomcat-juli.jar
Using CATALINA_PID:    /psc/acme1/temp/catalina.pid
Server version: Apache Tomcat/7.0.42
Server built:   Jul 2 2013 08:57:41
Server number:  7.0.42.0
OS Name:        Linux
OS Version:     2.6.18-164.el5
Architecture:   amd64
JVM Version:    1.7.0_02-b13
...
```

Display the value of a single property:

```
$: /psc/acme1/bin/tcman.sh config psc.as.http.port
psc.as.http.port=8501
```

Update the value of a property that exists in the `appserver.properties` file and then check the value:

```
$: /psc/acme1/bin/tcman.sh config psc.as.http.port=6543
$: tcman.sh config psc.as.http.port
psc.as.http.port=6543
```

Add a new property/value pair to the `appserver.properties` file and check the value:

```
$: /psc/acme1/bin/tcman.sh config +my.home.dir=/home/jarhead
$: tcman.sh config my.home.dir
my.home.dir=/home/jarhead
```

Update the server certificate in the `catalina.properties` file (see https://docs.oracle.com/cd/E19879-01/821-0185/ablqz/index.html for information about generating, exporting, and downloading a new server certificate):

```
$: /psc/acme1/bin/tcman.sh config psc.as.https.keyalias=myNewCert
```

Remove a property/value pair from the `appserver.properties` file and check if deletion was successful:

```
$: /psc/acme1/bin/tcman.sh config ~my.home.dir
$: tcman.sh config my.home.dir
Property does not exist - my.home.dir
```

Caution: There are no restrictions to property removal. The server will be unable to start if you remove a property required by `conf/server.xml`.

Notes

- All property names are case sensitive.
- You cannot enter multiple property names (*`prop_name`*) on the command line to view, update, or add properties to the `appserver.properties` file.
- You cannot use the config action to update existing values or add new values to the `jvm.properties` file

Display or modify the server features of an instance (feature)

Purpose

View, enable, or disable the server features contained in the `/conf/server.xml` file of an instance.

When you run `tcman.sh feature` with no parameters, it displays a list of the features (and their current status) that you can enable or disable. You can also display the status of a single server feature. After viewing the status of a feature, you can use `tcman.sh feature` to change its setting.

Syntax

```
tcman.sh feature [general_options] [feature_name[={on|off}]]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help feature` to see which general options are appropriate.

feature_name

Specify one of the features defined in an instance's `conf/server.xml` file. Running `tcman.sh feature` without *feature_name* displays a list of all the features.

`on`

Enables the named feature.

`off`

Disables the named feature.

Example

Display the list of server feature settings for `acme1`, enable AJP13 (Apache JServ Protocol. version 1.3), and verify that the feature is enabled:

```
$: /psc/acme1/bin/tcman.sh feature
SecurityListener=off
JMXLifecycle=off
PSCRegistry=on
HTTP=onHTTPS=on
AJP13=off
Cluster=off
UserDatabase=on
JAASRealm=off
LDAPRealm=off
PASInstrument=off
RemoteHostValve=on
RemoteAddrValve=onSingleSignOn=on
AccessLog=on
CrawlerSessionManager=on
StuckSessionValve=on

$: /psc/acme1/bin/tcman.sh feature AJP13=on


$: /psc/acme1/bin/tcman.sh feature AJP13
AJP13=on
```

Notes

- Server features for instances are set in `$CATALINA_BASE/conf/server.xml`. You can change feature status by manually editing this file. However, it is safer to use `tcman.sh feature` to avoid corrupting the file with erroneous entries.
- Run `tcman.sh feature` when the instance is offline.

See also

Display and manage an instance's configuration (config) on page 38
Stop an instance (stop) on page 51
The tcman command on page 14

Clean up or archive server log files (clean)

Purpose

Truncate, move, or delete the log files located in the `/logs` directory of the core server or instance. If the server is running, clean truncates log files to zero length. If the server is not running, clean deletes the log files from the file system.

You have the option to save log files to a subdirectory of `/logs`.

Syntax

```
tcman.sh clean [general_options] [-A]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help clean` to see which general options are appropriate.

`-A`

Archive log files to a subdirectory of `$CATALINA_BASE/logs`. The directory is automatically named with a month-day-year-second (*MM-DD-YYYY-HHmmss*) time-stamp format. If the server is not running, the files in `$CATALINA_BASE/logs` are deleted.

Example

Archive the log files of `acme1`, an instance of the core server `pashome`, and save to a file:

```
/psc/pashome/tcman.sh clean -I acme1 -A
```

See also

The tcman command on page 14

Display server instances (instances)

Purpose

Show the names and locations of the instances created from the PAS installed in `$CATALINA_HOME` by displaying the contents of the file where instances are registered for tracking.

By default, instances are registered for tracking `$CATALINA_HOME/conf/instances.{windows|.unix}`. The file name extension indicates the OS platform where the PAS server is installed.

Syntax

```
tcman.sh instances [general_options]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help instances` to see which general options are appropriate.

Output format

The following is the format of the output from a TCMAN `instances` action:

```
alias-name | full-file-path | type | state
```

alias-name

The user-defined name for the instance.

full-file-path

The location, in the OS file system, of the instance's root directory.

type

The designation of the server instance type (for example: `instance`, `service`, . . .).

state

An indication of the instance's validity. `OK` is returned for a valid server and `invalid` is returned for a corrupted or non-existant server.

Example

Display the instances of the core server installed in `/psc/pashome`:

```
/psc/pashome/bin/tcman.sh instances
acme1 | /psc/wrk/acme1 | instance | ok
acme2 | /psc/wrk/acme2 | instance | ok
```

Notes

- By default, instances are registered when you execute a `$CATALINA_HOME/bin/tcman{.sh|.bat} create` action, which automatically adds instance entries to an `instances` file. TCMAN removes instance entries from the file when you execute a `delete` action.

  You can manually add or remove instance entries from `instances` by using the `register` or `unregister` actions.

- By default, the name and location of the file where instances are registered is `$CATALINA_HOME/conf/instances.{windows|.unix}`.

  You can change the location of the instance registration file by adding and setting the `psc.as.instdir` property in the `appserver.properties` file. Use the TCMAN config action as in the following example:

  ```
  tcman.sh config '+psc.as.instdir=PATH'
  ```

  where *PATH* is a path name or an environment variable.

  You can also change the location and/or name of instance registration files by setting the environment variables, `PAS_AS_INSTANCE_DIR`, and `PAS_AS_INSTANCE_FILE`.

See also

Register an instance for tracking (register) on page 45
Stop tracking an instance (unregister) on page 54
The tcman command on page 14

List process ids (plist)

Purpose

List the process ids for all the processes that are running under an instance.

Syntax

```
tcman.sh plist [general_options] [-f]
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help plist` to see which general options are appropriate.

`-f`

Display verbose output. The output is indented and uses the plus (+) character to indicate parent-child relationships.

Examples

Display process id's for the running instance, `acme1` using the `-v` and `-f` options:

```
/psc/acme1/bin/tcman.sh plist -v
info: showing process ids for server 5942
5942 5963 5975 5988 6001 6015

/psc/acme1/bin/tcman.sh plist -f
5942
 +5963
 +5975
 +5988
 +6001
 +6015
```

Notes

The plist action is useful for administrative tasks such as:

- Checking if an instance is stopped or running. If it is running, it returns the process id of the instance. If it is stopped, 0 is returned.
- Using the output (which is easily parsable) in administrative scripts.
- If the instance's applications have started sub-processes, multiple process ids will be returned.
- If multiple process ids are listed, the instance is always listed first.

See also

Show Windows process information (showproc) on page 49

Register an instance for tracking (register)

Purpose

Register an instance for tracking purposes.

Note:

Instances are automatically registered for tracking when you execute a `create` action. You use the `register` action to restart tracking on instances after tracking was stopped.

A typical use for unregistering and then re-registering an instance is to make configuration changes when moving instances from one location (core server) to another. The `register` action enables tracking and also updates the value of `CATALINA_HOME` in all of the executable scripts in the instance's `/bin` directory to refer to the new core server.

Syntax

```
tcman.sh register alias_name instance_path
```

Parameters

`alias_name`

Specify a meaningful name for the instance. The alias name must be unique in the instances file.

`instance_path`

Specify the OS file system path to where the instance exists. This value will be expanded into a fully qualified OS directory path and will be verified to exist.

Example

Track `test1`, which is an alias for the instance `/psc/acme1`:

```
/psc/pashome/bin/tcman.sh register test1 /psc/acme1
```

Notes

When you register an instance for tracking or create a new instance with the `create` command, an entry is created in the core Progress Application Server's `$CATALINA_HOME/conf/instances.[unix|windows]` file.

The `instances.[unix|windows]` file is a simple text file, which can be manually edited (with care) in the event that it becomes out of date. The format for entries is:

```
instance_name = base_path
```

An `instances.unix` file uses Unix OS file path syntax (forward slashes), and an `instances.windows` file uses Windows OS file path syntax (backslashes) to specify *`base_path`*.

Also note that in an instances file:

- Any line starting with a pound-sign (`#`) is a comment line.
- Blank lines are skipped.
- Instance names cannot contain spaces or any of the following characters: `"[ . # | ] $ ? + = { / , }"`

See also

Stop tracking an instance (unregister) on page 54
The tcman command on page 14

Register and manage an instance as a Windows service (service)

Purpose

(Windows only) Registers or unregisters an instance as a Windows service. After an instance is registered, you can start, stop, or check the status of the service with this action. The `tcman service` action must be performed from the instance directory (`%CATALINA_BASE%/bin`) rather than the PAS for OpenEdge installation directory (`%CATALINA_HOME%/bin`).

Note: To register an instance as a Windows service, you must have JDK 1.8 installed on your computer. Installing JDK 1.8 is a prerequisite for installing OpenEdge. The instance registration process uses the `JAVA_HOME` environment variable (for example `JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101`).

Syntax

```
tcman.bat service [general_options] alias_name { register | unregister | start
| stop | status }
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.bat help service` to see which general options are appropriate.

alias_name

A required parameter that specifies the name of a PAS instance that was created using `tcman create`.

`register`

Create a new Windows service that runs the named PAS instance *alias_name*..

Set the `PR_DISPLAYNAME` and/or `PR_DESCRIPTION` variables to change the display name and description of the PAS instance service that appears in the Windows Service utility (**Services** tab of the Task Manager). The defaults for these variables are:

Variable	Default
`PR_DISPLAYNAME`	Progress Application Server *alias_name*
`PR_DESCRIPTION`	Progress Application Server (Tomcat 7) – http://www.progress.com

Set these variables before you register the instance. For example, if you wanted to change the defaults for `oepas1`:

```
set PR_DISPLAYNAME=PAS ROOT Server
set PR_DESCRIPTION=Progress Application Server
tcman service oepas1 register
```

`unregister`

Delete the Windows service that runs the named PAS instance *`alias_name`*

`start`

Start the Windows service that corresponds to the named PAS instance *`alias_name`*. The Windows service may also be started using the Windows service console or the SC command line utility.

`stop`

Stop the Windows service that corresponds to the named PAS instance *`alias_name`*. The Windows service may also be stopped using the Windows service console or the SC command line utility.

`status`

The registration status of the Windows service corresponding to the named PAS instance *`alias_name`*. The Windows service's status may be monitored using the Windows service console or SC command line utility

Example

Register the default instance oepas1 as a Windows service, then start, check status, stop, and unregister:

```
tcman service oepas1 register
  oepas1 service is registered

tcman service oepas1 start
  oepas1 started

tcman service oepas1 status
  Service oepas1 is running

tcman service oepas1 stop
  oepas1 is stopped

tcman service oepas1 unregister
  oepas1 is unregistered
```

Note

Be sure that the instance is not running before you attempt to register/unregister it.

Show Windows process information (showproc)

Purpose

(Windows only) Show information about the process specified by a process id.

Syntax

```
tcman.bat showproc [general_options] [process-id]
```

Parameters

general_options

Specify one or more of the options that can be used with any TCMAN action. Run `tcman.sh help showproc` to see which general options are appropriate.

process-id

The numerical identifier of a Windows process. You can obtain a list of process ids by running the TCMAN `plist` action.

Examples

Display process id's for the running instance, `acme1`, then specify process ids to show detailed information.

```
/psc/acme1/bin/tcman.bat plist -v
info: showing process ids for server with window title 13332
13332 14240

/psc/acme1/bin/tcman.bat showproc 13332
ProcesName                : java
SessionId                 : 2
StartTime                 : 10/04/2018 16:29:42
Threads                   : 26
TotalProcessorTime        : 00:00:19.9213277
UserProcessTime           :
CPU (seconds)             : 19.9213277
Description               : Java(TM) Platform SE binary
Path                      : C:\Program Files\Java\jdk1.8.0_101\bin\java.exe

/psc/acme1/bin/tcman.bat showproc 14240
ProcesName                : _mproapsv
SessionId                 : 2
StartTime                 : 10/04/2018 16:29:54
Threads                   : 7
TotalProcessorTime        : 00:00:00.3744024
UserProcessTime           :
CPU (seconds)             : 0.3744024
Description               : OpenEdge AppServer (Multi-thread)
Path                      : C:\Progress\OpenEdge\bin\_mproapsv.exe
```

See also

List process ids (plist) on page 44

Start an instance (start)

Purpose

Start an instance of a PAS, optionally in debug mode.

Syntax

```
tcman.sh start [general_options] [-D|-J]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help start` to see which general options are appropriate.

`-D`

Start the server in Tomcat debug mode. `-D` overrides the `-J` option.

`-J`

Start the server in debug mode using the JDPA (Java Platform Debugger Architecture) APIs for debugging. `-J` cannot be used if the `-D` option is specified.

Before you run a server with the `-J` option, you must define a port for the JDPA debugger by setting the `JDPA_ADDRESS` environment variable to a unique TCP network port number.

Example

Start the server in `/psc/acme1`, which is an instance of the core server in `/psc/pashome`:

```
/psc/acme1/bin/tcman.sh start
Using CATALINA_BASE:   /psc/acme1
Using CATALINA_HOME:   /psc/pashome
Using CATALINA_TMPDIR: /psc/acme1/temp
Using JRE_HOME:        /tools/linuxx86_64/java64/jdk1.8.0_101
Using CLASSPATH:       /psc/pashome/bin/bootstrap.jar:/psc/pashome/bin/tomcat-juli.jar
Using CATALINA_PID:    /psc/acme1/temp/catalina.pid
```

Notes

- When the TCMAN utility starts the server, it verifies the creation of the OS process and then records the server's process-id in a `.pid` file. The location of the `.pid` file is:

OS	PID File Path
UNIX	`$CATALINA_BASE/temp/catalina-`*`instance_name`*`.pid`
Windows	`$CATALINA_BASE\logs\catalina-`*`instance_name`*`.pid`

- You can obtain the process id of a server by running the TCMAN `env` action.

See also

Stop an instance (stop) on page 51
The tcman command on page 14

Stop an instance (stop)

Purpose

Stop a running instance, either gracefully or forcibly.

Note: TCMAN supports stopping a server instance that is not configured with a shutdown port.

On UNIX platforms stopping the running server instance is accomplished by sending a UNIX signal to the PAS process. Therefore, the administrator's process must have the UNIX permissions to signal the PAS process. On Windows platforms, the instance is identified using an OS process id that is used to stop server processes.

Syntax

```
tcman.sh stop [general_options] [-F[-w seconds]]
```

Parameters

`general_options`

Specify one or more of the general TCMAN options. Run `tcman.sh help stop` to see which general options are appropriate.

`-F`

Kill the sever process if it does not stop after a default wait time (5 seconds on UNIX, 10 seconds on Windows). Change the default wait interval by using the `–w` option.

`-w` *`seconds`*

Optionally specify the number of seconds to wait before killing a server process.

Example

Stop the server in `/psc/acme1`, which is an instance of the core server in `/psc/pashome`:

```
/psc/acme1/bin/tcman.sh stop
Using CATALINA_BASE:   /psc/acme1
Using CATALINA_HOME:   /psc/pashome
Using CATALINA_TMPDIR: /psc/acme1/temp
Using JRE_HOME:        /tools/linuxx86_64/java64/jdk1.8.0_101
Using CLASSPATH:       /psc/pashome/bin/bootstrap.jar:/psc/pashome/bin/tomcat-juli.jar
Using CATALINA_PID:    /psc/acme1/temp/catalina.pid
```

Notes

- TCMAN supports stopping a server instance that is not configured with a shutdown port.

  On UNIX platforms stopping the running server instance is accomplished by sending a UNIX signal to the PAS process. Therefore, the administrator's process must have the UNIX permissions to signal the PAS process. On Windows platforms, the instance is identified using an OS process id that is used to stop server processes.

  The following is an example a message you would see after a forced shut down with no shut down port:

  ```
  Sep 23, 2013 4:10:47 PM org.apache.catalina.startup.Catalina stopServer
  SEVERE: No shutdown port configured. Shut down server through OS signal.
  Server not shut down.
  Killing Tomcat with the PID: 14230
  ```

- Process ids are stored in the following locations:

OS	PID File Path
UNIX	`$CATALINA_BASE/temp/catalina-`*`instance_name`*`.pid`
Windows	`$CATALINA_BASE\logs\catalina-`*`instance_name`*`.pid`

- You can also obtain the process id of a server by running the TCMAN `env` action.

See also

Start an instance (start) on page 50
The tcman command on page 14

Display server, OS, and runtime version information (version)

Purpose

Show the Apache Tomcat runtime version and OS information for an instance.

To execute this action, the instance cannot be running

Syntax

```
tcman.sh version [general_options]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help version` to see which general options are appropriate.

Example

Display the server and runtime information for `acme1`, an instance of the core server installed in `/psc/pashome`:

```
$: /psc/pashome/bin/tcman.sh version -I acme1
Using CATALINA_BASE:   /psc/acme1
Using CATALINA_HOME:   /psc/pashome
Using CATALINA_TMPDIR: /psc/acme1/temp
Using JRE_HOME:        /tools/linuxx86_64/java64/jdk1.7.0_02/
Using CLASSPATH:
/psc/pashome/bin/bootstrap.jar:/users/doc/agarbacz/psc/pashome/bin/tomcat-juli.jar
Using CATALINA_PID:    /psc/acme1/temp/catalina.pid
Server version: Apache Tomcat/7.0.42
Server built:   Jul 2 2013 08:57:41
Server number:  7.0.42.0
OS Name:        Linux
OS Version:     2.6.18-164.el5
Architecture:   amd64
JVM Version:    1.7.0_02-b13
JVM Vendor:     Oracle Corporation
```

See also

Display OS and server information (info) on page 20
The tcman command on page 14

Test a server configuration (test)

Purpose

Displays information on the configuration and environment of an instance. It also displays information about error conditions.

The `test` action starts a server (instance), loads all the configuration files, and then displays information. The instance is stopped, exiting gracefully even if there is an error condition.

To execute this action, the instance cannot be running.

Syntax

```
tcman.sh test [general_options]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help test` to see which general options are appropriate.

Example

Run a test of the configuration of `acme1`, which is an instance of the core server installed at `/psc/pashome`:

```
$: /psc/pashome/bin/tcman.sh -I acme1 test
Using CATALINA_BASE:   /psc/acme1
Using CATALINA_HOME:   /psc/pashome
Using CATALINA_TMPDIR: /psc/acme1/temp
Using JRE_HOME:        /tools/linuxx86_64/java64/jdk1.8.0_101
Using CLASSPATH:       /psc/pashome/bin/bootstrap.jar:/psc/pashome/bin/tomcat-juli.jar
Using CATALINA_PID:    /psc/acme1/temp/catalina.pid
. . .
```

Notes

The `test` action is particularly useful for testing to verify that a server will start and run properly after you make changes to configuration and properties files.

See also

The tcman command on page 14

Stop tracking an instance (unregister)

Purpose

Stop tracking an instance by removing the instance's entry from the `$CATALINA_HOME/conf/instances.[unix|windows]` file.

Note:

You use the `register` action to restart tracking on instances after tracking was stopped with `unregister` .

A typical use for unregistering and then re-registering an instance, is to make configuration changes when moving instances from one location, or core server, to another. The `register` action not only enables tracking, it also updates the value of `CATALINA_HOME` in all of the executable scripts in the instance's `/bin` directory to refer to the new core server.

Syntax

```
tcman.sh unregister alias_name
```

Parameters

alias_name

Specify the alias name of the instance that you want to stop tracking. The alias name must exist in an `instances.[unix|windows]` file.

Example

Stop tracking `test1`, which is an instance of `/psc/pashome`:

```
/psc/pashome/bin/tcman.sh unregister test1
```

See also

Register an instance for tracking (register) on page 45
The tcman command on page 14

Create a Tomcat worker configuration file (workers)

Purpose

Create a preliminary `worker.properties` file that supports the configuration of supporting servers (workers) for a Progress Application Server (PAS) instance.

in the Apache Reference Guide, a worker is defined as an "instance that is waiting to execute servlets or any other content on behalf of some web server." In the context of the Progress Application Server, a worker is a server that is called by a PAS instance to perform a specific task. Typically,you would define worker instances to manage proxies, load balancing, clusters, or status monitoring. (For links to information on this functionality, see the Apache Tomcat Documentation Index.) There are probably other situations where you could improve the performance of a server instance by configuring worker instances to handle specific processing tasks.

In Apache Tomcat, workers are configured in a `worker.properties` file. The protocol implemented for communication between servers and workers is the Apache JServ Protocol (version 1.3, referred to as AJP13).

In TCMAN, the `workers` action adds the definitions of registered PAS instances to the content of the `$CATALINA_HOME/extras/workers.template` file and puts the result in `$CATALINA_HOME/temp/worker.properties`. The template file supplies a set of common directives that are referenced by all of the defined PAS instances. Individual instance definitions contain only the properties that are unique to the instance, such as the AJP13 network connection port. (See Table 3: worker.properties example on page 57.)

The `/temp/worker.properties` created by the `workers` action is a preliminary configuration file that you will probably need to modify to implement your deployment. See The Apache Tomcat Connector-Reference Guide for more information about configuring workers.

Syntax

```
$CATALINA_HOME/tcman.sh workers [general_options] [worker_list]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help worker` to see which general options are appropriate.

`worker_list`

A comma separated list of instance names and/or keywords. The keywords are:

status	Include an instance that has been implemented as a status server
lb	Include an instance that has been implemented as a load balance server
home	Include the CATALINA_HOME core server
all	Include all registered instances

If no *worker_list* is specified, the default worker list (all instances registered to CATALINA_HOME) will be added. If no instances have been created, then the default *worker_list* is CATALINA_HOME.

Examples

Assume there are:

- Two registered instances (`piw1` and `piw2`) that serve Web applications
- A Tomcat load balancer instance (`jklb`) that distributes the workload between `piw1` and `piw2`
- A status instance (`jkstatus`) that is used to monitor the runtime status of `piw1` and `piw2`

The following are examples of worker-lists showing various combinations of keywords and instances, and the resulting content in `$CATALINA_HOME/temp/worker.properties`:

Table 2: worker-list keywords

worker-list	Resulting content in worker.properties
blank	Default entries from `worker.template` plus entries for `piw1` and `piw2`
`piw1`	Default entries from `worker.template` plus an entry for `piw1`
`all`	Default entries from `worker.template` plus entries for `piw1` and `piw2`
`home`	Default entries from `worker.template` plus an entry for core server (CATALINA_HOME)
`home, all`	Default entries from `worker.template` plus an entry for core server, (CATALINA_HOME), and entries for for `piw1` and `piw2`

worker-list	Resulting content in worker.properties
`lb, status`	Default entries from `worker.template` plus entries for `jklb`, `jstatus`, `piw1`, and `piw2` **Note:** When no registered instance is specified, all registered instances are automatically included.
`lb, status, home, all`	Default entries from `worker.template` plus entries for `jklb`, `jstatus`, the core server (CATALINA_HOME),`piw1`, and `piw2`

The following is an example `workers.properties` file that includes entries for instances `piw1` and `piw2`:

Table 3: worker.properties example

```
# worker properties for instances rooted at U:\vobs\dlc\servers\pasw
# build date: 03/24/2014 11:09:23

# List of worker server instances
worker.list=piw1,piw2

#
# Global properties
#
# worker.maintain=60

#
# Common worker properties referenced by individual workers
#
worker.common.type=ajp13
worker.common.host=${psc.as.host.name}
worker.common.socket_timeout=10
worker.common.connect_timeout=10000
worker.common.socket_keepalive=true
worker.common.ping_mode=I
worker.common.ping_timeout=10000
worker.common.connect_timeout=0
worker.common.retry_interval=100
worker.common.recovery_options=7
# worker.common.connection_ping_interval=10
# worker.common.fail_on_status=0# worker.common.max_packet_size=8192
# worker.common.recover_time=60

# properties for alias piw1 with jvmRoute piw1
worker.piw1.port=9996
worker.piw1.reference=worker.common

# properties for alias piw2 with jvmRoute piw2
worker.piw2.port=9996
worker.piw2.reference=worker.common
```

Notes

- The `tcman workers` action must be run from the PAS installation's `$CATALINA_HOME/bin directory`.
- The `/extras/workers.template` file can be modified to adjust existing properties or to add additional static information. However, you cannot replace the common properties with a unique set of properties for each defined server.

- The `worker.common.connect_timeout=10000` is in milliseconds, and `worker.common.connection_ping_interval=10` is in seconds.

General actions

This section details the actions available for displaying help and server runtime environment information.

See also

The tcman command on page 14

Display help (help)

Purpose

Display summary or detailed help for all TCMAN actions, property names, and server features.

Syntax

```
tcman.sh help [action | property | feature]
```

Parameters

action

Show the syntax and options of the specified action. If no action is specified, show a list of all actions and the general options.

property

Show the settings for specified property.

feature

Show if the specified feature is enabled or disabled.

Example

Display the usage help for the `create` action:

```
$: tcman.sh help create
usage: tcman create [options] -p <http-port> [instance-opts] <new-base-path>
    instance-opts:
       [-s <shutdown-port>]
       [-P <https-port>]
       [-j <ajp13-port>]
       [-W <web-apps-dir>]
       [-N <inst-alias-name>]
       [-U <file-owner> -G <file-group>]

   general options:
      -u uid:pwd  pass uid and pwd for HTTP BASIC authentication
      -v          print verbose output
      -M url      override the CATALINA_BASE manager's URL with
                  <{http|https}://<host>:<port>/<mgr-app>
      -B          override CATALINA_BASE environment setting
      -n          debug run action but do not execute changes
```

Display runtime environment information (env)

Purpose

Show details about a server's state.

Syntax

```
tcman.sh env [general_options] [keyword]
```

Parameters

general_options

Specify one or more of the general TCMAN options. Run `tcman.sh help env` to see which general options are appropriate.

keyword

Specify one or more keywords that represent the name of the state that you want to view. If no keyword is specified, then all of the state information is displayed.

Keywords include:

Keyword	Description
`running`	Indicate if a server is running (`1`) or not running (`0`).
`mgrurl`	Display the URL of the manager application.

Keyword	Description
type	Display the server type.
alias	Display the server's alias.
parent	Display the pathname of the parent of an instance.
tracking	Indicate if tracking is on (1) or off (0).
http	Display the server's http port number.
https	Display the server's https port number.
shut	Display the server's shutdown port number. A value of -1 indicates that there is no shutdown port.
pid	Display the server's process id. A hyphen (-) indicates that the server is not running.

Example

Display all of the state information for the instance created in /psc/acme1:

```
/psc/acme1/bin/tscman.sh env
catalina home:     /psc/pashome
catalina base:     /psc/acme1
java home:         /tools/linuxx86_64/java64/jdk1.8.0_101
jre home:
manager http port: 8501
manager https port:8601
manager shut port: 8701
manager URL:       http://localhost:8501/manager
config type:       instance
config alias:      acme1
config parent:     /psc/pashome
server running:    0
instance tracking: 1
instance file:     /psc/pashome/conf/instances.unix
server process-id: -
```

3

REST API Reference for oemanager.war

The `oemanager.war` Java web application provides a REST API for remote management and monitoring of ABL applications deployed on a Progress Application Server (PAS) for OpenEdge instance. It is similar to the administration API supported by the Tomcat web server's JMX interface, but it uses JSON input/output payloads.

Requirements

To enable REST API access, either `oemanager.war` or `tcmanager.war` must be deployed, and Swagger UI must be enabled. For more information, see 'Use Swagger UI to explore management REST APIs' in *Manage Progress Application Server (PAS) for OpenEdge*.

Examples

The examples in this document show the use of `curl` and an ABL HTTP client. `curl` is a common open software utility used in command lines or scripts to transfer data. You can, however, employ the PAS for OpenEdge REST management APIs with many other tools and utilities. For more information about `curl`, see: http://curl.haxx.se/docs/. For more information about ABL HTTP clients, see *Develop an HTTP client application in ABL*.

For details, see the following topics:

- Session management
- Agent management
- Transport management
- Application management

Session management

The following topics describe the REST services that are available for managing PAS for OpenEdge sessions.

See also

Get session information on page 62
Get client information on page 64
Get current request information on page 65
Get configuration properties on page 67
Update configuration properties on page 69
Get runtime metrics on page 70
Reset runtime metrics on page 72
Cancel a request on page 73
Terminate a session on page 76

Get session information

Description

Query an ABL application to get a list of all its sessions including information on client and agent connections.

HTTP Operation

```
GET
```

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/sessions

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
404 Resource Not Found
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/sessions
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/sessions')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
"operation":"GET CLIENT SESSIONS",
"result":
    {
    "OEABLSession":
    [
        {
         "elapsedTimeMs":3343,
         "bound":false,
         "requestState":
         "RUNNING","requestID":"",
         "agentConnInfo":
             {
             "state":"RESERVED",
             "localAddr":"\/127.0.0.1:60464",
             "agentID":"AG-\/5vs3eApTMiPV5M+W47T3w",
             "connID":"AC-nRJgbuKtSNWcy6YWQknTbg",
             "connPoolID":"CP-30BInu3FRwOGQB0u0LwJ\/g",
             "agentAddr":"localhost\/127.0.0.1:62004"},
             "ablSessionID":"5",
             "sessionState":"RESERVED",
             "agentID":"AG-\/5vs3eApTMiPV5M+W47T3w",
             "clientConnInfo":
                 {
                 "elapsedTimeMs":5214,
                 "requestProcedure":"ngasPing.p",
                 "requestUrl":"http:\/\/localhost:16680\/apsv",
                 "clientName":"172.21.75.148",
                 "httpSessionId":"732277EE12485DDEC3A937DD30D7EF3267B3FE669C76.pas1",
                 "executerThreadId":"catalina-exec-8",
                 "reqStartTimeStr":"2014-09-17T08:41:48.138-0400",
                 "requestID":"ngasPing.p",
                 "adapterType":"HTTP",
                 "sessionID":"Iphz68lUQKexuHibyy6S+A"
                 },
             "adapterType":"HTTP",
             "sessionPoolID":"SP-RFplWI9\/Rta9j1pp+WcRyQ",
             "sessionID":"Iphz68lUQKexuHibyy6S+A",
             "sessionType":"SESSION_MANAGED",
             "lastAccessStr":"2014-09-17T08:41:50.006-0400"
        }
    ]
    },
    "outcome":"SUCCESS",
    "errmsg":"",
```

```
    "versionStr":"PASOE 11.5.0",
    "versionNo":1
}
```

Note: For more information about session states, see 'About session and request states' in *Manage Progress Application Server (PAS) for OpenEdge*.

Get client information

Description

Query an ABL application to get a information on client connections.

HTTP Operation

```
GET
```

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/clients[?timeoutms=*n*]

timeoutms

The period of time, specified in milliseconds (ranging from 1 to *n*), to report on client connections, including hung clients.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line examples

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/clients

curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/clients?timeoutms=100000
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
```

```
resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/clients')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
"operation":"GET CLIENT CONNECTIONS",
"result":
    {
    "ClientConnection":
        [
        {
        "elapsedTimeMs":427133,
        "requestProcedure":"ngasPing.p",
        "requestUrl":
        "http:\/\/localhost.com:16680\/apsv",
        "clientName":"172.21.75.148",
        "httpSessionId":"732277EE12485DDEC3A937DD30D7EF3267B3FE669C76.pas1",
        "executerThreadId":"catalina-exec-8",
        "reqStartTimeStr":"2014-09-17T08:41:48.138-0400",
        "requestID":"ngasPing.p",
        "adapterType":"HTTP",
        "sessionID":"Iphz68lUQKexuHibyy6S+A"
        }
        ]
    },
    "outcome":"SUCCESS",
    "errmsg":"",
    "versionStr":"PASOE 11.5.0",
    "versionNo":1
}
```


Get current request information

Description

Query an ABL application to get a list of all the requests currently in process.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/requests[?requestId=VALUE]

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line examples with and without request ID

```
curl -X GET -v http://localhost:16680/oemanager/applications/oepas1/requests -u
username:password

curl -X GET -v
http://localhost:16680/oemanager/applications/oepas1/requests?requestId=ROOT:s:00000002
 -u username:password
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/requests)
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
"operation": "GET ACTIVE REQUESTS ON SERVER",
"outcome": "SUCCESS",
"result": {
    "Request": [
        {
        "requestID": "ROOT:s:00000002",
            "tomcat": {
                "methodType": "POST",
                "startTime": "2019-07-15T02:58:21.816-0400",
                "url": "http://localhost:16680/soap"
            },
            "sessionManager": {
                "sessionId": "C0SS-8CkRbOhSv-QvyaYUA",
                "requestState": "RUNNING",
                "userId": "anonymousUser",
                "requestStartTimeStamp": "2019-07-15T02:58:21.997-0400",
                "requestElapsedTime": 197563
```

```
            },
            "agent": {
                "sessionId": “8”,
                "requestProcName": "server.p",
                "connectionId": "35",
                "startTime": "2019-07-10T07:43:30.517",
                "endTime": "",
                "requestNum": 1
            }
        },
    },
"errmsg": "",
"versionStr": "v12.1.0",
"versionNo": 2
}
```

Note: Identify `requestID` from the body of the response received when querying information about currently active requests.

Note: For more information about request states, see 'About session and request states' in *Manage Progress Application Server (PAS) for OpenEdge*.

Get configuration properties

Description

Query the `conf/openedge.properties` file to get the configuration properties of an application. The information is specified in the `AppServer.SessMgr.`*`App_name`* section of `openedge.properties`.

HTTP Operation

`GET`

URI

`//`*`host_name`*`:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/properties`

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
404 Resource Not Found
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/properties
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/properties')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
   "operation":"GET BROKER PROPERTIES",
   "outcome":"SUCCESS",
   "result":{
      "defaultAgentWaitAfterStop":"30000",
      "allowRuntimeUpdates":"0",
      "idleConnectionTimeout":"300000",
      "brkrDebuggerEnabled":"0",
      "defrdLoggingLevel":"0",
      "idleResourceTimeout":"0",
      "publishDir":"C:\/OpenEdge\/WRK\/oepas1\/openedge",
      "statusEnabled":"1",
      "connectionWaitTimeout":"3000",
      "requestWaitTimeout":"15000",
      "maxConnectionsPerAgent":"200",
      "collectMetrics":"3",
      "updateProperties":"false",
     "agentLogFile":"C:\\OpenEdge\\WRK\\oepas1\/logs\/oepas1.agent.{yyyy-mm-dd}.log",

      "maxABLSessionsPerAgent":"200",
      "socketTimeout":"3000",
      "agentStartLimit":"1",
      "defaultAgentWaitToFinish":"30000",
      "ipver":"IPv4",
      "defrdLogNumLines":"0",
      "agentExecFile":"C:\\Progress\\OpenEdge\/bin\/_mproapsv",
      "defrdLogEntryTypes":"",
      "idleAgentTimeout":"1800000",
      "agentLogEntryTypes":"ASPlumbing,DB.Connects",
      "minAgents":"1",
      "agentStartupParam":"-T \"C:\\OpenEdge\\WRK\\oepas1\/temp\" -db
C:\\OpenEdge\\WRK\\database\\sports2020.db",
      "maxAgents":"2",
      "agentListenerTimeout":"300000",
      "numInitialAgents":"1",
      "agentLoggingLevel":"2",
      "agentWatchdogTimeout":"3000",
      "idleSessionTimeout":"300000",
      "applications":"oepas1"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Update configuration properties

Description

Update an ABL application's configuration properties in the `AppServer.SessMgr` section of the `conf/openedge.properties` file.

HTTP Operation

`PUT`

URI

`//host_name:port/oemanager/applications/`*`ABL_app_name`*`/properties`

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/properties
-d {"agentLoggingLevel":"4"} -H "Content-Type: application/vnd.progress+json"
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('agentLoggingLevel',  '4').
resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
     :Put('http://localhost:16680/oemanager/applications/oepas1/properties', jsonData)

      :ContentType('application/vnd.progress+json')
     :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

```
{"agentLoggingLevel":"4"}
```

Response body examples

On success:

```
{
   "operation":"SET BROKER PROPERTIES",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

On failure:

```
{
   "operation":"SET BROKER PROPERTIES",
   "outcome":"FAILURE",
   "result":"",
   "errmsg":"Invalid property name: \"InvalidPropName\". (PRPMsg009)",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get runtime metrics

Description

Query the session manager for runtime metrics of an ABL application.

For more information, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

HTTP Operation

`GET`

URI

`//host_name:port/oemanager/applications/ABL_app_name/metrics`

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/metrics
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/metrics')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

The following example response is returned when `countMetrics=3` in the `openedge.properties` file:


```
{
   "operation":"GET SESSION-MGR METRICS",
   "outcome":"SUCCESS",
   "result":{
      "requests":10,
      "writeErrors":0,
      "reads":62,
      "concurrentConnectedClients":0,
      "maxAgentReadTime":0,
      "writes":20,
      "maxConcurrentClients":1,
      "readErrors":0,
      "avgReserveABLSessionWaitTime":0,
      "totReserveABLSessionWaitTime":3,
      "stdDevAgentReadTime":0,
      "maxReserveABLSessionWaitTime":3,
      "avgAgentReadTime":0,
      "numReserveABLSessionTimeouts":0,
      "minAgentReadTime":0,
      "numReserveABLSessionWaits":0,
      "type":"OE_BROKER",
      "startTime":"2019-12-03T14:53:47.385-05:00",
      "accessTime":"2019-12-03T15:45:12.262-05:00"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
```

```
   "versionNo":1
}
```

Reset runtime metrics

Description

Reset runtime metrics of the session manager.

For more information, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

HTTP Operation

`DELETE`

URI

`//host_name:port/oemanager/applications/`*`ABL_app_name`*`/metrics`

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/metrics
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Delete('http://localhost:16680/oemanager/applications/oepas1/metrics')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"RESET SESSION-MGR METRICS",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Cancel a request

Description

Cancel the request identified by a Request ID and a Session ID. You can obtain Request IDs and Session IDs by executing the `oemanager/applications/`*`ABL_app_name`*`/sessions`service.

HTTP Operation

`DELETE`

URI

`//host_name``:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/requests?requestID=`*`value`*`&sessionID=`*`value`*

Media type

`NA`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X DELETE -u username:password -v
"http://localhost:16680/oemanager/applications/oepas1
/requests?requestID=z9WHoRgGY4fnExrFWKWVlw&sessionID=G1TTNRPT3KiSOhkbyeSow"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Delete('http://localhost:16680/oemanager/applications/oepas1
/requests?requestID=z9WHoRgGY4fnExrFWKWVlw&sessionID=G1TTNRPT3KiSOhkbyeSow')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
```

```
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"CANCEL ACTIVE REQUEST ON SERVER",
   "outcome":"FAILURE",
   "result":"",
   "errmsg":null,
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Dump a session stack

Description

Dump a process stack for a specific session.

Note: As with the classic OpenEdge Appserver, you can generate a stack for PAS for OpenEdge by using a debugging tool. For the classic Appserver, you can also run the `proGetStack` utility to generate stack traces. For PAS for OpenEdge and its multi-session agent, this REST call is analogous to `proGetStack`.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentPID*/sessions/*sessionid*/stacks

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
Application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
"http://localhost:8810/oemanager/applications/oepas1/agents/P9PDwujXQqeFoGgnL7wZfw/sessions/7/stacks"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:8810/oemanager/applications/oepas1/agents/P9PDwujXQqeFoGgnL7wZfw/sessions/7/st

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body examples

Success

```
{
   "outcome":"SUCCESS",
   "errmsg":"",
   "versionStr":"v11.7.1 ( 2017-06-09)",
   "versionNo":1,
   "operation":"GET SESSION RUNTIME STACKS",
   "result":{
      "ABLStacks":[
         {
            "AgentSessionId":7,
            "StartupParams":"-pf
      \/usr1\/example\/117\/dlc\/startup.pf, -cpinternal ISO8859-1,-cpstream
ISO8859-1,-cpcoll
      Basic,-cpcase Basic, -d mdy,-numsep 44,-numdec 46,(end .pf),-logginglevel 2,
      -logfile \/usr1\/example\/117\/wrk\/oepas1\/logs\/oepas1.agent.log, -uburl
      AppServerDC:\/\/0.0.0.0:52511\/, -logname oepas1,-logentrytypes
      ASPlumbing,DB.Connects, -ubpropfile
      \/usr1\/example\/117\/wrk\/oepas1\/conf\/openedge.properties, -ASID 1,-ipver
IPv4,-sectok
      XXXXXXXXXXXXXXXXXXXXX, -T \/usr1\/example\/117\/wrk\/oepas1\/temp, -db
      \/usr1\/example\/117\/wrk\/sports2000",
            "Propath":…(more)
```

Failure

```
{
   "outcome":"FAILURE",
   "errmsg":"Communication layer message: General Error:READPACKET IOException :
java.io.EOFException (null). (7175)",
```

```
   "versionStr":"v11.7.1( 2017-06-09 )",
   "versionNo":1,
   "operation":"GET SESSION RUNTIME STACKS",
   "result":""
}
```

Terminate a session

Description

Terminate the session identified by the specified `sessionID`.

HTTP Operation

`DELETE`

URI

//host_name:*port*/oemanager/applications/*App_name*
/sessions?sessionID=*value*&terminateOpt={ 0 | 1 }

where a `terminateOpt` value of `0` causes a graceful termination and a value of `1` causes a forced termination.

Media type

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X DELETE -u username:password -v
"http://localhost:16680/oemanager/applications/oepas1/sessions?sessionID=123&terminateOpt=1"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/sessions?sessionID=123&terminateOpt=1')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"TERMINATE SESSIONS",
   "outcome":"SUCCESS",
   "result":false,
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Terminate idle ABL sessions

Description

Terminate all idle sessions associated with a specific agent.

Note: With the classic OpenEdge AppServer, you could trim idle sessions periodically by terminating agents, since each agent is a session. However, with PAS for OpenEdge and its multisession agent, trimming the agent could terminate both active and idle sessisons. Therefore, you should use this REST API call to trim the idle sessions for a specified agent.

HTTP Operation

DELETE

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*
/agentid/sessions

Media type

Application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl -X DELETE -u username:password -v
"http://localhost:8810/oemanager/applications/oepas1/RDGoyUUMTaaTQJbFPN7t1Q/sessions"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/RDGoyUUMTaaTQJbFPN7t1Q/sessions')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body examples

Success

```
{
   "operation":"TERMINATE SESSIONS",
   "outcome":"SUCCESS",
   "result":2,
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Failure

```
{
   "operation":"TERMINATE SESSIONS",
   "outcome":"FAILURE",
   "result":0,
  "errmsg":"AgentNotFound[SessionManager.terminateFreeSessions. Error! Agent not found.

Agent Id: \"WKoQ7x5MQMKWHJoyOGp8uQ\"]:Agent",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get dynamic session limit

Description

Query to get the current in-memory max available ABL Sessions (`dynmaxablsessions`) limit.

HTTP Operation

`GET`

URI

```
//host_name:port/oemanager/applications/ABL_app_name/agents/agent_ID/dynamicSessionLimit
```

Note: Agent PID (for example, 1144) or agent ID (for example, W9UI_6dJQ-qSsk-o25mWAw) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/39028/dynamicSessionLimit
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/39028/dynamicSessionLimit')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"GetDynamicABLSessionInfo",
   "outcome":"SUCCESS",
   "result":{
      "AgentSessionInfo":[
         {
            "ABLOutput":{
               "numABLSessions":2,
               "numAvailableSessions":2,
               "dynmaxablsessions":200
            },
            "ABLReturnVal":true,
```

```
            "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
            "pid":"10291"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.2.0 ( 2019-12-18 )",
   "versionNo":1
}
```

Update dynamic session limit

Description

Updates the in-memory maximum ABL Sessions per agent (`dynmaxablsessions`) to the specified `{sessionLimit}`.

HTTP Operation

`POST`

URI

`//host_name:port/oemanager/applications/`*`ABL_app_name`*`/agents/`*`agentPID`*`/dynamicSessionLimit/`*`{sessionLimit}`*

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

`{sessionLimit}`

The integer to set as the value of the in-memory `dynmaxablsessions`. This number cannot exceed the configured number of ABL sessions per agent (`maxABLSessionsPerAgent`) in `openedge.properties`.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X POST -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/1144/dynamicSessionLimit/100
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Post('http://localhost:16680/oemanager/applications/oepas1/agents/1144/dynamicSessionLimit/100')

      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"SetDynamicABLSessionLimit",
   "outcome":"SUCCESS",
   "result":{
      "AgentSessionInfo":[
         {
            "ABLOutput":{
               "numABLSessions":2,
               "numAvailableSessions":2,
               "dynmaxablsessions":100
            },
            "ABLReturnVal":true,
            "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
            "pid":"10291"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.2.0 ( 2019-12-18 )",
   "versionNo":1
}
```

Reset dynamic session limit

Description

Reset the in-memory maximum ABL Sessions per Agent (`dynmaxablsessions`) back to the value that `maxABLSessionsPerAgent` is set to in the `openedge.properties` file.

HTTP Operation

`DELETE`

URI

`//host_name``:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/agents/`*`agentPID`*`/dynamicSessionLimit`

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
403 Access Denied
500 Unexpected Server Error
```

Command-line example

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/1144/dynamicSessionLimit
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/agents/1144/dynamicSessionLimit')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"ResetDynamicABLSessionLimit",
   "outcome":"SUCCESS",
   "result":{
      "AgentSessionInfo":[
         {
            "ABLOutput":{
               "numABLSessions":2,
               "numAvailableSessions":2,
               "dynmaxablsessions":200
            },
            "ABLReturnVal":true,
            "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
            "pid":"10291"
         }
      ]
```

```
    },
    "errmsg":"",
    "versionStr":"v12.2.0 ( 2019-12-18 )",
    "versionNo":1
}
```

Agent management

The following topics describe the REST services that are available for managing PAS for OpenEdge agents.

See also

List agents on page 83
Get agent status information on page 89
Get session metrics on page 99
Get connection information on page 90
Get request information on page 91
Get agent configuration properties on page 93
Update agent configuration properties on page 95
Stop a multi-session agent on page 96

List agents

Description

Get a list of all the PAS for OpenEdge multi-session agents available to an ABL application.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/agents')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"GET AGENTS",
   "outcome":"SUCCESS",
   "result":{
      "agents":[
         {
            "agentId":"W9UI_6dJQ-qSsk-o25mWAw",
            "pid":"1144",
            "state":"AVAILABLE"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Check ABL object tracking status

Checks whether ABL object tracking is enabled or disabled. To change the status, see Enable or disable ABL object tracking on page 85.

HTTP Operation

`GET`

URI

`http(s)://`*`host_name`*`:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/agents/`*`agentPID`*`/ABLObjects/status`

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects/status
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects/status')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

Result is false when tracking is off. Result is true when tracking is on. In this example ABL object tracking is disabled.

```
{
   "operation":"TRACKING ABL OBJECTS",
   "outcome":"SUCCESS",
   "result":false,
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Enable or disable ABL object tracking

Description

Enable or disable the tracking of ABL objects. When enabled, ABL objects are tracked to identify potential memory leaks. To generate a report of these objects, see Get a report of tracked ABL objects on page 87. When object tracking is not required, disable. Disabled by default.

HTTP Operation

PUT

URI

```
http(s)://host_name:port/oemanager/applications/ABL_app_name/agents/agentPID/ABLObjects/status
```

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects/status
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('enable',  'true').
resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects/status,
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

```
{"enable":{"true" |  "false"} }
```

Response body example

Result returns true when value is changed. To get the current value, see Check ABL object tracking status on page 84.

```
{
   "operation":"TRACKING ABL OBJECTS",
   "outcome":"SUCCESS",
   "result":true,
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get a report of tracked ABL objects

Description

Get a report of ABL objects that may indicate potential memory leaks. Getting a report requires that object tracking is enabled. See Enable or disable ABL object tracking on page 85. After leaks have been identified and resolved, disable ABL object tracking.

HTTP Operation

`GET`

URI

`http(s)://`*`host_name`*`:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/agents/`*`agentPID`*`/ABLObjects`

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
```

```
:Get('http://localhost:16680/oemanager/applications/oepas1/agents/18668/ABLObjects')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

This excerpt shows the objects that are being tracked. `Source` identifies the related code.

```
{
    "result": {
        "ABLOutput": {
            "ABLObjects": [
                {
                    "AgentSessionId": 4,
                    "Objects": []
                },
                {
                    "AgentSessionId": 7,
                    "Objects": []
                }
            ]
        },
        "ABLReturnVal": true,
        "agentId": "y11SsN0cSC2skp9NV-N0Ww",
        "pid": "6064"
    },
    "errmsg": "",
    "versionStr": "v11.7.4 ( 2018-09-17 )",
    "versionNo": 1,
    "outcome": "SUCCESS",
    "operation": "GET ABL OBJECTS REPORT"
}
```

This excerpt shows results if object tracking is disabled.

```
{
    "result": {
        "ABLReturnVal": false,
        "agentId": "y11SsN0cSC2skp9NV-N0Ww",
        "pid": "6064"
    },
    "errmsg": "",
    "versionStr": "v11.7.4 ( 2018-09-17 )",
    "versionNo": 1,
    "outcome": "SUCCESS",
    "operation": "GET ABL OBJECTS REPORT"
}
```

Get agent status information

Description

Get a summary of sessions, connections, and requests for an agent of an ABL application.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentPID*/status

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/39028/status
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/39028/ABLObjects/status')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"GET AGENT STATUS",
   "outcome":"SUCCESS",
   "result":{
      "threads":1,
      "sessions":2,
      "connections":0,
      "requests":0
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get connection information

Description

Get information on the connections of an agent of an ABL application.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentPID*/connections

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/39028/connections
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/39028/ABLObjects/connections')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"GET AGENT CONNECTIONS",
   "outcome":"SUCCESS",
   "result":{
      "AgentConnection":[
         {
            "ConnectionId":1564,
            "ConnectionState":"CONNECTED",
            "SessionId":7
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get request information

Description

Get request information for an agent of an ABL application.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentPID*/requests

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/39028/requests
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
     :Get('http://localhost:16680/oemanager/applications/oepas1/agents/39028/requests')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
   "operation":"GET AGENT REQUESTS",
   "outcome":"SUCCESS",
   "result":{
      "AgentRequest":[
         {
            "RequestProcName":"Progress.Web.InternalWebHandler:HandleRequest",
            "SessionId":7,
            "ConnectionId":1564,
            "StartTime":"2019-12-04T11:54:55.859",
            "EndTime":"2019-12-04T11:54:56.236",
            "RequestNum":0,
            "BrokerSessionId":"0DXjcGm2SxC208AbN1NF9g",
            "ClientId":"ROOT:w:00000002"
         },
         {
            "RequestProcName":"OpenEdge.Rest.Admin.AppServerStatus:ServerStatus",
            "SessionId":7,
            "ConnectionId":1564,
            "StartTime":"2019-12-04T11:55:35.079",
            "EndTime":"2019-12-04T11:55:35.079",
            "RequestNum":3,
            "BrokerSessionId":"yetFYGUhTl2s8ioJXq5J2A",
```

```
            "ClientId":"ROOT:r:00000006"
         },
         {
            "RequestProcName":"Progress.Web.InternalWebHandler:HandleRequest",
            "SessionId":7,
            "ConnectionId":1564,
            "StartTime":"2019-12-04T11:55:40.443",
            "EndTime":"2019-12-04T11:55:40.603",
            "RequestNum":4,
            "BrokerSessionId":"0DXjcGm2SxC208AbN1NF9g",
            "ClientId":"ROOT:w:00000007"
         },
         {
            "RequestProcName":"ServerGetCustNameSample.p",
            "SessionId":7,
            "ConnectionId":1564,
            "StartTime":"2019-12-04T11:58:45.154",
            "EndTime":"2019-12-04T11:58:45.154",
            "RequestNum":5,
            "BrokerSessionId":"B93D78F6F33B774D0BD1364965E7CFD8CB281E308DDA.oepas1",
            "ClientId":"ROOT:a:00000010"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


Get agent configuration properties

Description

Get configuration information (properties) of agents for an ABL application.

HTTP Operation

GET

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/properties

Media type

application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/properties
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/agents/properties')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
   "operation":"GET AGENT PROPERTIES",
   "outcome":"SUCCESS",
   "result":{
      "sessionConnectProc":"",
      "agentMaxPort":"62202",
      "sessionStartupProcParam":"",
      "allowRuntimeUpdates":"0",
      "flushStatsData":"0",
      "lockAllNonThreadSafeExtLib":"",
      "sessionShutdownProc":"",
      "agentStartupProcParam":"",
      "uuid":"http://localhost:16680/pas1",
      "statusEnabled":"1",
      "collectStatsData":"0",
      "sessionStartupProc":"",
      "collectMetrics":"1",
      "sessionDisconnProc":"",
      "updateProperties":"false",
      "binaryUploadMaxSize":"0",
      "fileUploadDirectory":"",
      "sessionDeactivateProc":"",
      "workDir":"C:\\OpenEdge\\WRK\\oepas1/work",
      "agentShutdownProc":"",
      "usingThreadSafeExtLib":"",
      "sessionActivateProc":"",
      "agentStartupProc":"",
     "PROPATH":"C:\\OpenEdge\\WRK\\oepas1/webapps/pas1/WEB-INF/openedge,C:/OpenEdge/WRK/oepas1/webapps

      "sessionExecutionTimeLimit":"0",
      "agentMinPort":"62002",
      "infoVersion":"9010",
      "lockAllExtLib":"",
      "numInitialSessions":"2",
      "applications":"oepas1"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Update agent configuration properties

Description

Update the configuration properties in `conf/openedge.properties` for agents for an ABL application.

HTTP Operation

`PUT`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/properties
{"*prop_name*"=*value***[**, "*prop_name*"=*value***[**,**...]]}**

"*prop_name*"=*value***[**, *prop_name*"=*value***[**,**...]]**

Set one or more properties expressed as name/value pairs in a comma-separated list. See the `/conf/openedge.properties.README` file for a description of properties and their possible values.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/properties
-d {"collectMetrics":0} -H "Content-Type: application/vnd.progress+json"
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('collectMetrics',  '0').
resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Put('http://localhost:16680/oemanager/applications/oepas1/agents/properties',
jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
```

```
      :Request
   ).
```

Request body example

```
{"collectMetrics":0}
```

Response body example

On success:

```
{
   "operation":"SET AGENT PROPERTIES",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

On failure:

```
{
   "operation":"SET AGENT PROPERTIES",
   "outcome":"FAILURE",
   "result":"",
   "errmsg":"Invalid property name: \"propname\". (PRPMsg009)",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Stop a multi-session agent

Description

Stop a multi-session agent that belongs to an ABL application. Note that you identify the agent by the *agentID*, which you can obtain by executing `/oemanager/applications/`*ABL_app_name*`/agents`. You can specify, in milliseconds, how long the stop process must wait before stopping an agent, if the agent is currently serving a request, using the `waitToFinish` parameter. `waitToFinish` stops new requests to the agent and waits for any current requests to complete. If the current requests do not complete in the specified time, a STOP message is sent to the agent. You can also specify an additional wait time, in milliseconds, using the `waitAfterStop` parameter. `waitAfterStop` checks if the agent has indeed stopped. If the agent has not stopped, then at the end of the specified time, a KILL message is sent to the agent. In either case, if the agent finishes processing its current requests before the `waitToFinish` time expires, the agent will immediately shut down.

Note: Both `waitToFinish` and `waitAfterStop` are optional parameters. If you do not specify these, the agent will be stopped immediately, irrespective of whether the agent is currently idle or serving a request.

HTTP Operation

DELETE

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentID*?
waitToFinish=*time-in-milliseconds*&waitAfterStop=*time-in-milliseconds*

Media type

application/vnd.progress+json

Response codes

200 Success
403 Access Denied
500 Unexpected Server Error

Command-line example

```
curl -X DELETE -v
http://localhost:16680/oemanager/applications/oepas1/agents/AG-sCIVXeFSQYmcb7RcHo10Zw?waitToFinish=
60000 -u username:password -H "Content-Type: application/vnd.progress+json"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/agents/AG-sCIVXeFSQYmcb7RcHo10Zw?waitTo
60000')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

NA

Response body example

```
{
  "result": {
    "agentID": "AG-sCIVXeFSQYmcb7RcHo10Zw"
  },
  "versionStr": "v11.7.3 ( 2018-02-12 )",
  "versionNo": 1,
  "outcome":"SUCCESS",
  "errmsg":"",
```

```
  "operation": "STOP AGENT ",
  }
```

Add a multi-session agent

Description

Add a multi-session agent to an ABL application.

HTTP Operation

POST

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/addAgent

Media type

application/vnd.progress+json

Response codes

200 Success
403 Access Denied
500 Unexpected Server Error

Command-line example

```
curl -X POST -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/addAgent
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Post('http://localhost:16680/oemanager/applications/oepas1/addAgent')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"GET ABL OBJECTS REPORT",
   "outcome":"SUCCESS",
   "result":{
      "agentId":"Zi9XLdNfRwOWaDg3TGQHnQ",
      "pid":"11152",
      "state":"AVAILABLE"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get session metrics

Description

Get session metrics for an agent of an ABL application.

HTTP Operation

GET

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/agents/*agentPID*/sessions

Note: Agent PID (for example, 1144) or agent ID (for example, W9UI_6dJQ-qSsk-o25mWAw) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/39028/sessions
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
     :Get('http://localhost:16680/oemanager/applications/oepas1/agents/39028/sessions')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example


```
{
   "operation":"GET AGENT SESSIONS",
   "outcome":"SUCCESS",
   "result":{
      "AgentSession":[
         {
            "SessionId":4,
            "SessionState":"IDLE",
            "StartTime":"2019-12-04T11:48:22.453",
            "EndTime":null,
            "ThreadId":-1,
            "ConnectionId":null,
            "SessionExternalState":0,
            "SessionMemory":870085
         },
         {
            "SessionId":7,
            "SessionState":"IDLE",
            "StartTime":"2019-12-04T11:48:22.453",
            "EndTime":null,
            "ThreadId":-1,
            "ConnectionId":-1,
            "SessionExternalState":0,
            "SessionMemory":10310903
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


Refresh all ABL sessions of one or all agents

Description

Refresh Agents terminates all sessions within an ABL application, thereby refreshing the multi-session agent or agents so that changes to the application can be picked-up by the agent or agents.

Note that you identify the specific agent by the *agentID*, which you can obtain by executing /oemanager/applications/*ABL_app_name*/agents, or you can enter all in place of an agent ID to refresh all sessions in all agents in the ABL application.

HTTP Operation

DELETE

URI

//*host_name*:*port*/oemanager/applications/*App_name*/agents/**{***agentID***|**all**}**/sessions

Media type

application/vnd.progress+json

Response codes

200 Success
403 Access Denied
500 Unexpected Server Error

Command-line example

Refresh agents command to terminate sessions in all agents using the all endpoint:

```
curl -X DELETE -v
http://localhost:8810/oemanager/applications/oepas1/agents/all/sessions -u
username:password -H "accept: */*"
```

Refresh agent command to terminate sessions in a specified agent using an agent with the agent ID LdS8phtPR0GIrhRrrSL3Xg:

```
curl -X DELETE -v
http://localhost:8810/oemanager/applications/oepas1/agents/LdS8phtPR0GIrhRrrSL3Xg/sessions
 -u username:password -H "accept: */*"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
     :Delete('http://localhost:8810/oemanager/applications/oepas1/agents/all/sessions')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
   "operation":"REFRESH AGENTS",
   "outcome":"SUCCESS",
   "result":{
      "agents":[
         {
            "agentId":"q5NOCzTARCGpWy0eWv5rAw",
            "pid":"42636",
            "sessions":[
               {
                  "sessionId":4,
                  "terminateResult":"success"
               },
               {
                  "sessionId":7,
                  "terminateResult":"success"
               }
            ]
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Flush deferred log buffer to agent log file

Description

Writes the deferred log buffer of an agent or agents to the agent log file. For more information about deferred logging, see 'Use deferred logging in PAS for OpenEdge' in *Manage Progress Application Server (PAS) for OpenEdge*.

Note that you identify the specific agent by the *agentID*, which you can obtain by executing `/oemanager/applications/`*`ABL_app_name`*`/agents`, or you can specify `all` to flush the deferred log buffer to the agent log files of all agents in the application.

HTTP Operation

`GET`

URI

`//host_name``:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/agents/{`*`agentID`* `| all}/flushDeferredLog`

Note: Agent PID (for example, `1144`) or agent ID (for example, `W9UI_6dJQ-qSsk-o25mWAw`) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
403 Access Denied
500 Unexpected Server Error
```

Command-line example

Flush deferred log buffer to agent log file for an agent with the agent ID `LdS8phtPR0GIrhRrrSL3Xg`:

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/LdS8phtPR0GIrhRrrSL3Xg/flushDeferredLog
```

Flush deferred log buffer to agent log file for all agents:

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/all/flushDeferredLog
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/all/flushDeferredLog')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
  "operation": "FLUSH DEFERRED LOGS",
  "outcome": "SUCCESS",
  "result": {
    "flushDeferredLog": true
  },
  "errmsg": "",
  "versionStr": "v12.1.0",
  "versionNo": 1
}
```

Reset deferred log buffer

Description

Empties the deferred log buffer. For more information about deferred logging, see 'Use deferred logging in PAS for OpenEdge' in *Manage Progress Application Server (PAS) for OpenEdge*.

Note that you identify the specific agent by the *agentID*, which you can obtain by executing /oemanager/applications/*ABL_app_name*/agents, or you can specify *all* to flush the deferred log buffer to the agent log files of all agents in the application.

HTTP Operation

GET

URI

```
//host_name:port/oemanager/applications/ABL_app_name/agents/{agentID |
all}/resetDeferredLog
```

Note: Agent PID (for example, 1144) or agent ID (for example, W9UI_6dJQ-qSsk-o25mWAw) can be used to identify the multi-session agent. Both values can be obtained using the List agents on page 83 API.

Media type

```
application/vnd.progress+json
```

Response codes

```
200 Success
403 Access Denied
500 Unexpected Server Error
```

Command-line example

Flush deferred log buffer to agent log file for an agent with the agent ID LdS8phtPR0GIrhRrrSL3Xg:

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/LdS8phtPR0GIrhRrrSL3Xg/resetDeferredLog
```

Flush deferred log buffer to agent log file for all agents:

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/agents/all/resetDeferredLog
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/agents/all/resetDeferredLog')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Request body example

NA

Response body example

```
{
  "operation": "RESET DEFERRED LOGS",
  "outcome": "SUCCESS",
  "result": {
    "resetDeferredLog": "true"
  },
  "errmsg": "",
  "versionStr": "v12.1.0",
  "versionNo": 1
}
```

Transport management

The following topics describe the REST services that are available for managing PAS for OpenEdge SOAP, REST, APSV, and WEB transports.

See also

Get transport configuration properties on page 106
Update properties of a transport (SOAP/REST) on page 107
Enable or disable a transport (SOAP/REST) on page 112
Enable or disable a transport (APSV) on page 114
Get runtime metrics of a transport on page 115
Get runtime metrics of a transport (APSV)
Reset runtime metrics for a transport on page 119
Reset runtime metrics of a transport (APSV)
List deployed services (SOAP/REST) on page 121
Get information about a service (SOAP/REST) on page 124
Get runtime properties of a service (SOAP) on page 126
Deploy a new service (REST) on page 127
Deploy a new service (SOAP) on page 129

Update runtime properties of a service (SOAP) on page 131
Get runtime metrics for a service (SOAP) on page 133
Reset runtime metrics for a service (SOAP) on page 135
Enable or disable a service (SOAP) on page 136
Remove a service (SOAP/REST) on page 137

Get transport configuration properties

Description

Get configuration properties of APSV, REST, SOAP and WEB transports of a web application belonging to an ABL application.

HTTP Operation

`GET`

URI

hostname:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/{apsv | rest | soap | web}/properties

Content-Type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties
```

ABL HTTP client example

Note: As indicated in the URI syntax, `soap` could be interchanged with `apsv`, `rest`, or `web`.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body

NA

Response body example

The following response is for the SOAP transport. Similar responses are produced when calling the APSV, REST, or WEB transport properties in the URI.

```
{
   "operation":"GET SOAP TRANSPORT PROPERTIES",
   "outcome":"SUCCESS",
   "result":{
      "idleSessionTimeout":0,
      "waitIfBusy":0,
      "connectionLifetime":0,
      "staleO4GLObjectTimeout":0,
      "requestWaitTimeout":-1,
      "clientASKActivityTimeout":60,
      "initialSessions":1,
      "minIdleConnections":0,
      "clientASKResponseTimeout":60,
      "maxSessions":0,
      "minSessions":1,
      "appServerKeepalive":"denyClientASK,allowServerASK",
      "serviceFaultLevel":2,
      "openEdgeProps":{
         "statusEnabled":"1",
         "wsdlEnabled":"1",
         "collectMetrics":"1",
         "debugClients":"",
         "webApps":"ROOT,secondapp",
         "allowRuntimeUpdates":"0",
         "adminEnabled":"1",
         "adapterEnabled":"1",
         "wsaUrl":"http://localhost:16680/soap",
         "adminSoapAction":"urn:services-progress-com:wsa-admin:01"
      }
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Note: Runtime properties come from the *instance-name*/webapps/*webapplname*/WEB-INF/adapters/soap/runtime.props file. OpenEdge properties come from the *instance-name*/conf/openedge.properties file.

Update properties of a transport (SOAP/REST)

Description

Update the properties of a SOAP or REST transport of a web application belonging to an ABL application. For SOAP and REST transport properties, both runtime and OpenEdge properties can be modified.

Runtime properties are derived from from the *instance-name*/webapps/*webapplname*/WEB-INF/adapters/soap/runtime.props file. OpenEdge properties come from the *instance-name*/conf/openedge.properties file.

Runtime property fields are the same for both the REST and SOAP transport. SOAP offers a few additional OpenEdge properties. Obtain the properties for each service by running Get transport configuration properties on page 106 for the necessary transport.

HTTP Operation

PUT

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/{soap | rest}/properties

Content-Type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example for Runtime-only properties (SOAP or REST)

Substitute `soap` or `rest` in the URI for the desired transport. Both transports contain identical runtime properties (as can seen in the following command-line example) which can be updated.

SOAP:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d {"clientASKResponseTimeout":160,
"maxSessions":11,
"minSessions":1, "serviceFaultLevel":2}
```

REST:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d {"clientASKResponseTimeout":160,
"maxSessions":11,
"minSessions":1, "serviceFaultLevel":2}
```

ABL HTTP client example for Runtime-only properties (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to update properties for the `rest` transport. Obtain the property list for each service by running Get transport configuration properties on page 106 for the necessary transport.

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('clientASKResponseTimeout',  160).
```

```
jsonData:Add('maxSessions',  11).
jsonData:Add('minSessions',  1).
jsonData:Add('serviceFaultLevel',  2).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties',
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example for Runtime-only properties (SOAP or REST)

```
{
 "clientASKResponseTimeout": 160,
 "maxSessions": 11,
 "minSessions": 1,
 "serviceFaultLevel": 2
}
```

Command-line example for OpenEdge-only properties (SOAP or REST)

The SOAP transport offers additional OpenEdge properties which can be updated.

SOAP:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d {"openEdgeProps":{"collectMetrics":"2"}}
```

REST:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d {"openEdgeProps":{"collectMetrics":"2"}}
```

ABL HTTP client example for OpenEdge-only properties (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to update properties for the `rest` transport. Obtain the property list for each service by running Get transport configuration properties on page 106 for the necessary transport.

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.
```

```
define variable oepropsJsonData as JsonObject.

oepropsJsonData = new Progress.Json.ObjectModel.JsonObject().
oepropsJsonData:Add('collectMetrics', "2").

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('openEdgeProps', oepropsJsonData).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties',
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example for OpenEdge-only properties (SOAP or REST)

```
{
 "openEdgeProps": {
  "collectMetrics":"2"
 }
}
```

Command-line example for Combined (OpenEdge and Runtime) properties (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to update properties for the `rest` transport. Obtain the property list for each service by running Get transport configuration properties on page 106 for the necessary transport.

SOAP:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d
{"clientASKResponseTimeout":60,"maxSessions":11,
"minSessions":1,"serviceFaultLevel": 2,"openEdgeProps": {"collectMetrics": "2"}}
```

REST:

```
curl -X PUT -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/properties
 -u username:password
-H "Content-Type: application/vnd.progress+json" -d
{"clientASKResponseTimeout":60,"maxSessions":11,
"minSessions":1,"serviceFaultLevel": 2,"openEdgeProps": {"collectMetrics": "2"}}
```

ABL HTTP client example for Combined (OpenEdge and Runtime) properties (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to update properties for the `rest` transport. Obtain the property list for each service by running Get transport configuration properties on page 106 for the necessary transport.

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

oepropsJsonData = new Progress.Json.ObjectModel.JsonObject().
oepropsJsonData:Add('collectMetrics', "2").

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('clientASKResponseTimeout',  60).
jsonData:Add('maxSessions',  11).
jsonData:Add('minSessions',  1).
jsonData:Add('serviceFaultLevel',  2).
jsonData:Add('openEdgeProps', oepropsJsonData).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/properties',
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username, 'password'))
      :Request
   ).
```

Request body example for Combined (OpenEdge and Runtime) properties (SOAP or REST)

```
{
 "clientASKResponseTimeout": 60,
 "maxSessions": 11,
 "minSessions": 1,
 "serviceFaultLevel": 2,
 "openEdgeProps": {
  "collectMetrics": "2"
 }
}
```

Response body example (SOAP)

```
{
   "operation":"GET SOAP TRANSPORT PROPERTIES",
   "outcome":"SUCCESS",
   "result":{
      "idleSessionTimeout":0,
      "waitIfBusy":0,
      "connectionLifetime":0,
      "staleO4GLObjectTimeout":0,
      "requestWaitTimeout":-1,
      "clientASKActivityTimeout":60,
      "initialSessions":2,
```

```
      "minIdleConnections":0,
      "clientASKResponseTimeout":60,
      "maxSessions":11,
      "minSessions":1,
      "appServerKeepalive":"denyClientASK,allowServerASK",
      "serviceFaultLevel":2,
      "openEdgeProps":{
         "statusEnabled":"1",
         "wsdlEnabled":"1",
         "collectMetrics":"2",
         "debugClients":"",
         "webApps":"ROOT",
         "allowRuntimeUpdates":"0",
         "adminEnabled":"1",
         "adapterEnabled":"1",
         "wsaUrl":"http://localhost:8810/soap",
         "adminSoapAction":"urn:services-progress-com:wsa-admin:01"
      }
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


Response body example (REST)


```
{
   "operation":"GET REST TRANSPORT PROPERTIES",
   "outcome":"SUCCESS",
   "result":{
      "maxSessions":0,
      "initialSessions":1,
      "serviceFaultLevel":2,
      "waitIfBusy":1,
      "minSessions":1,
      "clientASKActivityTimeout":60,
      "clientASKResponseTimeout":60,
      "requestWaitTimeout":-1,
      "minIdleConnections":0,
      "appServerKeepalive":"denyClientASK,allowServerASK",
      "staleO4GLObjectTimeout":0,
      "idleSessionTimeout":0,
      "connectionLifetime":0,
      "openEdgeProps":{
         "statusEnabled":"1",
         "collectMetrics":"1",
         "webApps":"ROOT",
         "allowRuntimeUpdates":"0",
         "adapterEnabled":"1"
      }
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


Enable or disable a transport (SOAP/REST)

Description

Enable or disable the REST or SOAP transport of a web application belonging to an ABL application.

HTTP Operation

PUT

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*Web_app_name*/transports/{soap | rest}/state

Content-Type

```
application/vnd.progress+json
```

Request body

```
{
  "STATE" : "{ENABLED | DISABLED}"
}
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example (SOAP)

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/state

-d {"state":"ENABLED"}  -H "Content-Type: application/vnd.progress+json"
```

Response body example (SOAP)

```
{
   "operation":"SET SOAP TRANSPORT PROPERTIES",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (REST)

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/state

-d {"state":"DISABLED"} -H "Content-Type: application/vnd.progress+json"
```

Response body example (REST)

Check `oemanager/applications/oepas1/webapps/ROOT/transports/rest/properties` to ensure `adapterEnabled` is set to `0` or `1`, depending on whether the state was disabled or enabled.

ABL HTTP client example (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to update state for the `rest` transport.

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('state', 'ENABLED').

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/state',
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Enable or disable a transport (APSV)

Description

Enable or disable the APSV transport of a web application belonging to an ABL application.

HTTP Operation

`PUT`

URI

`host_name``:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/webapps/`*`Web_app_name`*`/transports/apsv/state`

Content-Type

`application/vnd.progress+json`

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -d {"state":"DISABLED"}  -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/apsv/state
```

Request body example

```
{"state":"DISABLED"}
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('state', 'DISABLED').

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/apsv/state',
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Response body example

```
{
   "operation":"SET APSV TRANSPORT STATE",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Get runtime metrics of a transport

Description

Get runtime metrics for the APSV, REST, SOAP, or WEB transport of an ABL web application deployed in an ABL application.

For more information, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

HTTP Operation

GET

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/{apsv | soap | rest | web}/metrics

Content-Type

application/vnd.progress+json

Request body

NA

Response codes

200 Success
500 Unexpected Server Error

Command-line example (APSV)

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/apsv/metrics
```

Response body example (APSV)


```
{
   "operation":"GET APSV TRANSPORT STATISTICS",
   "outcome":"SUCCESS",
   "result":{
      "statusRequests":0,
      "connectRequests":0,
      "sessionErrors":0,
      "connectErrors":0,
      "disconnectRequests":0,
      "sessionRequests":0,
      "disconnectErrors":0,
      "forbiddenErrors":0,
      "type":"OE_APSV_TRANSPORT",
      "startTime":"2020-01-10T11:29:43.160-05:00",
      "accessTime":"2020-01-10T11:30:16.692-05:00"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (SOAP)

```
curl -x GET -u username:password  -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/metrics
```

Response body example for count and time metrics (SOAP)

```
{
   "operation":"GET OE_SOAP_TRANSPORT METRICS",
   "outcome":"SUCCESS",
   "result":{
      "minTotalTime":24,
      "statusRequests":0,
      "stdDevTotalTime":4,
      "avgTotalTime":30,
      "serviceDisabled":2,
      "soapProcessorErrors":2,
      "httpRequestErrors":0,
      "httpRequests":0,
      "maxTotalTime":36,
      "methodNotAllowederrors":0,
      "soapRequests":10,
      "wsdlRequests":0,
      "successfulSoapRequests":8,
      "activeRequests":0,
      "urlNotFoundErrors":0,
      "type":"OE_SOAP_TRANSPORT",
      "startTime":"2020-01-14T15:36:15.372-05:00",
      "accessTime":"2020-01-14T15:36:39.128-05:00"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (REST)

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/metrics
```

Response body example (REST)

```
{
   "operation":"GET OE_REST_TRANSPORT METRICS",
   "outcome":"SUCCESS",
   "result":{
      "avgConnectTime":0,
      "statusRequests":0,
      "successfulRunRequests":0,
      "requests":0,
      "serviceUnavailableRequests":0,
      "failedRequests":0,
      "minDisconnectTime":0,
      "stdDevDisconnectTime":0,
      "connectRequests":0,
```

```
      "stdDevConnectTime":0,
      "expressionErrors":0,
      "successfulConnectRequests":0,
      "stdDevSessionTime":0,
      "successfulRequests":0,
      "maxConnectTime":0,
      "minSessionTime":0,
      "runRequests":0,
      "avgSessionTime":0,
      "maxDisconnectTime":0,
      "serviceNotFoundErrors":0,
      "minConnectTime":0,
      "avgDisconnectTime":0,
      "maxSessionTime":0,
      "type":"OE_REST_TRANSPORT",
      "startTime":"2020-01-14T15:36:15.371-05:00",
      "accessTime":"2020-01-14T15:38:07.557-05:00"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


Command-line example (WEB)

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/web/metrics
```

Response body example (WEB)


```
{
   "operation":"GET WEB TRANSPORT STATISTICS",
   "outcome":"SUCCESS",
   "result":{
      "patchErrors":0,
      "postErrors":0,
      "failedServletRequests":0,
      "maxABLConnectTime":0,
      "stdABLDevConnectTime":0,
      "getRequests":0,
      "optionsErrors":0,
      "connectRequests":0,
      "putRequests":0,
      "getErrors":0,
      "deleteRequests":0,
      "minABLProcessingTime":0,
      "maxABLProcessingTime":0,
      "stdDevABLProcessingTime":0,
      "minABLConnectTime":0,
      "patchRequests":0,
      "ablConnectErrors":0,
      "postRequests":0,
      "successfulServletRequests":0,
      "avgABLProcessingTime":0,
      "avgABLConnectTime":0,
      "optionsRequests":0,
      "traceRequests":0,
      "putErrors":0,
      "deleteErrors":0,
      "headRequests":0,
      "traceErrors":0,
```

```
      "servletRequests":0,
      "headErrors":0,
      "ablRuntimeErrors":0,
      "type":"OE_OEWEB_TRANSPORT",
      "startTime":"2020-01-14T15:36:15.373-05:00",
      "accessTime":"2020-01-14T15:38:47.395-05:00"
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

ABL HTTP client example

Note: Substitute `rest`, `apsv`, or `web` for `soap` in the URI below if you want to get metrics for the `rest`, `apsv`, or `web` transport.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/metrics')

      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Reset runtime metrics for a transport

Description

Reset aggregate runtime metrics for a APSV, SOAP, REST, or WEB transport of an ABL web application deployed in an ABL application.

For more information, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge* .

HTTP Operation

`DELETE`

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*Web_app_name*/transports/{soap | rest}/metrics

Content-Type

`application/vnd.progress+json`

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example (APSV)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/apsv/metrics
```

Response body example (APSV)

```
{
   "operation":"RESET APSV TRANSPORT STATISTICS",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (SOAP)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/metrics
```

Response body example (SOAP)

```
{
   "operation":"RESET OE_SOAP_TRANSPORT METRICS",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (REST)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/metrics
```

Response body example (REST)

```
{
   "operation":"RESET OE_REST_TRANSPORT METRICS",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Command-line example (WEB)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/web/metrics
```

Response body example (WEB)

```
{
   "operation":"RESET WEB TRANSPORT STATISTICS",
   "outcome":"SUCCESS",
   "result":"",
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

ABL HTTP client example (all transports)

Note: Substitute `apsv`, `rest`, or `web` for `soap` in the URI below if you want to get metrics for the `apsv`, `rest`, or `web` transport.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/metrics')

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

List deployed services (SOAP/REST)

Description

Get a list all the deployed SOAP or REST services in a web application deployed in an ABL application.

HTTP Operation

GET

URI

host_name:*port*/oemanager/applications/*App_name*/webapps/*Web_app_name*/transports/{soap | rest}/oeservices

Content-Type

application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Request body

NA

Command-line example (SOAP)

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices
```

Response body example(SOAP)


```
{
  "result": {
    "OESoapServiceDesc": [
      {
 "accessUrl":"http://localhost:8810/soap/wsdl?targetURI=urn:qad-pft",
 "archiveName":"PFT.wsm",
 "archiveLocation": "/webapps/ROOT/WEB-INF/adapters/soap/ROOT/SimpleTest.wsdl",
 "oetype":"SOAP_DESCRIPTOR",
 "name":"PFT",
 "state":"ENABLED",
 "type":"OPENEDGE",
 "version":"12.0.0",
 "description": "PAS OpenEdge SOAP Service Artifact",
 "securedUri": ""
 }
 {
 "accessUrl":"http://localhost:8810/soap/wsdl?targetURI=urn:nsRL1",
 "archiveName": "simpleRL.wsm",
 "archiveLocation": "/webapps/ROOT/WEB-INF/adapters/soap",
 "oetype":"SOAP_DESCRIPTOR",
 "name":"simpleRL",
 "state":"ENABLED",
 "type": "OPENEDGE",
 "version":"12.0.0",
 "description": "PAS OpenEdge SOAP Service Artifact",
 "securedUri":""
 }
    ]
```

```
  },
  "operation": "GET SOAP TRANSPORT DESCRIPTORS",
  "versionStr": null,
  "versionNo": 1,
  "errmsg": ""
  "outcome": "SUCCESS"
}
```

Command-line example (REST)

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/pas1/webapps/ROOT/transports/rest/oeservices
```

Response body example(REST)

```
{
   "operation":"GET REST TRANSPORT DESCRIPTORS",
   "outcome":"SUCCESS",
   "result":{
      "OERestServiceDesc":[
         {
            "name":"_oepingService",
            "version":"v12.1.0 ( 2019-08-07 )",
            "uri":"http:\/\/localhost:8810\/rest\/_oepingService",
            "securedUri":"https:\/\/localhost:8811\/rest\/_oepingService",
            "description":"PAS OpenEdge REST Service Descriptor",
            "type":"OPENEDGE",

"archiveLocation":"WEB-INF\/adapters\/rest\/_oepingService\/_oepingService.paar",
            "accessUrl":"\/rest\/_oepingService",
            "archiveName":"_oepingService.paar",
            "oetype":"REST_DESCRIPTOR"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

ABL HTTP client example

Note: Substitute `rest` for `soap` in the URI below if you want to list the services for the `rest` transport.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices')

      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
```

```
      :Request
   ).
```

Get information about a service (SOAP/REST)

Description

Get details about a named SOAP or REST service in a web application deployed in an ABL application.

HTTP Operation

GET

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/{soap|rest}/oeservices/{*SoapServiceName*|*RestSeviceName*}

Content-Type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Request body

NA

Command-line example (SOAP)

```
curl -X GET -u username:password -v
http://localhost:8810/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/SimpleTes
```

Response body example (SOAP)

```
{
  "result": {
    "OESoapServiceDesc": [
      {
        "accessUrl":"http://localhost:8810/soap/wsdl?targetURI=urn:nsRL1",
        "archiveName": "simpleRL.wsm",
        "archiveLocation": "/WEB-INF/adapters/soap/",
        "oetype": "SOAP_DESCRIPTOR",
        "name": "simpleRL",
        "state": "ENABLED",
        "type": "OPENEDGE"
        "version": "12.0.0",
        "description": "PAS OpenEdge SOAP Service Artifact",
```

```
        "securedUri": ""
      }
    ]
  },
  "operation": "GET SOAP TRANSPORT DESCRIPTOR",
  "versionStr": null,
  "versionNo": 1,
  "errmsg": "",
  "outcome": "SUCCESS"
}
```

Command-line example (REST)

```
curl -X GET -u username:password -v
http://localhost:8810/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices/_oepingSe
```

Response body example (REST)

```
{
   "operation":"GET REST TRANSPORT DESCRIPTOR",
   "outcome":"SUCCESS",
   "result":{
      "OERestServiceDesc":[
         {
            "name":"_oepingService",
            "version":"v12.1.0 ( 2019-08-07 )",
            "uri":"http:\/\/localhost:8810\/rest\/_oepingService",
            "securedUri":"https:\/\/localhost:8811\/rest\/_oepingService",
            "description":"PAS OpenEdge REST Service Descriptor",
            "type":"OPENEDGE",

"archiveLocation":"WEB-INF\/adapters\/rest\/_oepingService\/_oepingService.paar",
            "accessUrl":"\/rest\/_oepingService",
            "archiveName":"_oepingService.paar",
            "oetype":"REST_DESCRIPTOR"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

ABL HTTP client example

Note: Substitute `soap` for `rest`, as well as a soap service descriptor, in the URI below if you want to get information about a service for the `soap` transport.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices/_o
```

```
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Get runtime properties of a service (SOAP)

Description

Get runtime configuration properties of a named SOAP service in a web application deployed in an ABL application.

HTTP Operation

GET

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/properties

Content-Type

application/vnd.progress+json

Response body

NA

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/SimpleTe
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/Si

      :ContentType('application/vnd.progress+json')
```

```
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

```
{
   "result":{
      "idleSessionTimeout":0,
      "waitIfBusy":0,
      "connectionLifetime":0,
      "requestWaitTimeout":-1,
      "staleO4GLObjectTimeout":0,
      "clientASKActivityTimeout":60,
      "initialSessions":1,
      "clientASKResponseTimeout":60,
      "minIdleConnections":0,
      "maxSessions":0,
      "minSessions":1,
      "appServerKeepalive":"denyClientASK,allowServerASK",
      "serviceFaultLevel":2
   },
   "outcome":"SUCCESS",
   "versionNo":1,
   "errmsg":"NA",
   "versionStr":"PASOE 11.5.0",
   "operation":"GET SOAP TRANSPORT PROPERTIES"
}
```

Note: Runtime properties for a deployed SOAP service are stored in *instance-name*/webapps/*web_app_name*/WEB-INF/adapters/soap/*service_name*/*service_name*.props

Deploy a new service (REST)

Description

Deploy a new REST service in a web application deployed in an ABL application.

HTTP Operation

POST

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*Web_app_name*/transports/rest/oeservices/

Content-Type

application/vnd.progress.paar+zip

Content disposition

```
attachment; filename={ AppNewService.paar | AppNewService.zip }
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X POST -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices
--data-binary @RESTService.paar
-H "Accept: application/vnd.progress+json"
-H "Content-Type:application/vnd.progress.paar+zip"
-H "Content-Disposition: attachment; filename=RESTService.paar"
```

ABL HTTP client example

```
DEFINE VARIABLE oPaar AS CLASS Memptr NO-UNDO.
DEFINE VARIABLE mMyMemPtr AS MEMPTR NO-UNDO.
DEFINE VARIABLE resp AS OpenEdge.Net.HTTP.IHttpResponse.

FILE-INFO:FILE-NAME = "C:\OpenEdge\WRK\RESTService.paar".
SET-SIZE(mMyMemPtr) = FILE-INFO:FILE-SIZE.

INPUT FROM VALUE(FILE-INFO:FILE-NAME) BINARY NO-MAP NO-CONVERT.
IMPORT mMyMemPtr.
INPUT CLOSE.

oPaar = new Memptr(mMyMemPtr).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Post('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices')

      :WithData(oPaar)
      :AcceptContentType('application/vnd.progress+json')
      :AddHeader('Content-Disposition', 'attachment; filename="RESTService.paar"')
      :ContentType('application/vnd.progress.paar+zip')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).

SET-SIZE(mMyMemPtr) = 0.
```

Request body example

```
application/vnd.progress.paar+zip
```

Response body example

```
{
   "operation":"DEPLOY REST TRANSPORT DESCRIPTOR",
   "outcome":"SUCCESS",
   "result":{
```

```
      "OERestServiceDesc":[
         {
            "name":"RESTService",
            "version":"v12.1.0 ( 2019-08-07 )",
            "uri":"",
            "securedUri":"",
            "description":"PAS OpenEdge REST Service Descriptor",
            "type":"OPENEDGE",
          "archiveLocation":"WEB-INF\/adapters\/rest\/RESTService\/RESTService.paar",

            "accessUrl":"\/rest\/RESTService",
            "archiveName":"RESTService.paar",
            "oetype":"REST_DESCRIPTOR"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Deploy a new service (SOAP)

Description

Deploy a new SOAP service in a web application deployed in an ABL application.

HTTP Operation

POST

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps
/Web_app_name/transports/soap/oeservices/

Content-Type

application/vnd.progress.wsm+xml

Content disposition

attachment; filename=*AppNewService*.wsm

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl  -X POST -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices
--data-binary @SimpleTest2.wsm
```

```
-H "Accept: application/vnd.progress+json"
-H "Content-Type: application/vnd.progress.wsm+xml"
-H "Content-Disposition: attachment; filename=SimpleTest2.wsm"
```

ABL HTTP client example

```
DEFINE VARIABLE oWsm AS CLASS Memptr NO-UNDO.
DEFINE VARIABLE mMyMemPtr AS MEMPTR NO-UNDO.
DEFINE VARIABLE resp AS OpenEdge.Net.HTTP.IHttpResponse.

FILE-INFO:FILE-NAME = "C:\OpenEdge\WRK\SimpleTest2.wsm".
SET-SIZE(mMyMemPtr) = FILE-INFO:FILE-SIZE.

INPUT FROM VALUE(FILE-INFO:FILE-NAME) BINARY NO-MAP NO-CONVERT.
IMPORT mMyMemPtr.
INPUT CLOSE.

oWsm = new Memptr(mMyMemPtr).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Post('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices')

      :WithData(oWsm)
      :AcceptContentType('application/vnd.progress+json')
      :AddHeader('Content-Disposition', 'attachment; filename="SimpleTest2.wsm"')
      :ContentType('application/vnd.progress.wsm+xml')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).

SET-SIZE(mMyMemPtr) = 0.
```

Request body example

NA

Response body example

```
{
   "operation":"DEPLOY SOAP TRANSPORT DESCRIPTOR",
   "outcome":"SUCCESS",
   "result":{
      "OESoapServiceDesc":[
         {
            "accessUrl":"http://localhost:16680/soap/wsdl?targetURI=urn:nsRL1",
            "archiveName":"simpleRL.wsm",
            "archiveLocation":"/webapps/ROOT/WEB-INF/adapters/soap",
            "oetype":"SOAP_DESCRIPTOR",
            "name":"simpleRL",
            "state":"ENABLED",
            "type":"OPENEDGE",
            "version":"12.0.0",
            "description":"PAS OpenEdge SOAP Service Artifact",
            "securedUri":""
         }
      ]
   },
   "errmsg":"",
```

```
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

Update runtime properties of a service (SOAP)

Description

Update the runtime properties of a SOAP service in the web application of a deployed ABL application.

HTTP Operation

PUT

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/properties

Content-Type

```
application/vnd.progress+json
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/service/TestService
-H "Content-Type: application/vnd.progress+json" -d {"maxSessions":1}
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('maxSessions',  1).

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/service/TestS
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Request body example

```
{"maxSessions":1}
```

Response body example

```
{"result":
          {"idleSessionTimeout":0,
           "waitIfBusy":1,
           "connectionLifetime":0,
           "requestWaitTimeout":-1,
           "staleO4GLObjectTimeout":0,
           "clientASKActivityTimeout":60,
           "initialSessions":1,
           "clientASKResponseTimeout":60,
           "minIdleConnections":0,
           "maxSessions":1,
           "minSessions":1,
           "appServerKeepalive":"denyClientASK,allowServerASK",
           "serviceFaultLevel":2
          },
 "outcome":"SUCCESS",
 "versionNo":1,
 "errmsg":"NA",
 "versionStr":
 "PASOE 11.5.0",
 "operation":"SET SOAP TRANSPORT PROPERTIES"
}
```

Reset runtime properties of a service (SOAP)

Description

Reset the runtime properties of a SOAP service in the web application of a deployed ABL application back to the default values.

HTTP Operation

DELETE

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/properties

Content-Type

application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Command-line example

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/server/oeservices/ROOT/transports/soap/service/TestService2/properti
-H "Content-Type: application/vnd.progress+json"
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/server/oeservices/ROOT/transports/soap/service/TestService2

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Response body example

```
{
        "result": "",
        "operation": "RESET SOAP TRANSPORT PROPERTIES",
        "versionStr": null,
        "versionNo": 1,
        "errmsg": "",
        "outcome": "SUCCESS"
}
```

Get runtime metrics for a service (SOAP)

Description

Update the runtime metrics of a SOAP service in the web application deployed in an ABL application.

HTTP Operation

GET

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/metrics

Content-Type

application/vnd.progress+json

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/SimpleTe
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/Si

      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Response body example

```
{"result":
         {"requests":0,
          "appserverConnections":0,
          "nameserverError":0,
          "activeRequests":0,
          "providerError":0,
          "subAppObjects":0,
          "objPoolFull":0,
          "objPoolExpired":0,
          "connectError":0,
          "faults":0,
          "ABLClientError":0,
          "procObjects":0,
          "startTime":"2014-07-17T08:40:07.315-04:00",
          "appObjects":0,
          "objNotFound":0,
          "ABLServerError":0,
          "accessTime":"2014-07-18T02:27:07.212-04:00",
          "ABLAppError":0,
          "clientError":0,
          "type":"OE_SOAP_TRANSPORT_APP"
          },
 "outcome":"SUCCESS",
 "versionNo":1,
 "errmsg":"NA",
 "versionStr":null,
 "operation":"GET OE_SOAP_TRANSPORT_APP METRICS"
}
```

Reset runtime metrics for a service (SOAP)

Description

Reset runtime metrics for a SOAP service in a web application deployed in an ABL application.

HTTP Operation

DELETE

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/metrics

Content-Type

```
application/vnd.progress+json
```

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/SimpleTe
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Response body example

```
{
 "result":"NA",
```

```
 "outcome":"SUCCESS",
 "versionNo":1,
 "errmsg":"NA",
 "versionStr":null,
 "operation":"RESET OE_SOAP_TRANSPORT_APP METRICS"
}
```

Enable or disable a service (SOAP)

Description

Enable or disable a SOAP service in a web application deployed in an ABL application.

HTTP Operation

PUT

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/soap/oeservices/*SoapServiceName*/state

Content-Type

```
application/vnd.progress+json
```

Request body

```
{ "STATE" : "{ENABLED | DISABLED}" }
```

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X PUT -u username:password -v
http://localhost:16680/oemanager/applications/pas1/webapps/ROOT/transports/soap/oeservices/SimpleTest
-H "Content-Type: application/vnd.progress+json" -d {"state":"DISABLED"}
```

ABL HTTP client example

```
using Progress.Json.ObjectModel.*.
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.
define variable jsonData as JsonObject.

jsonData = new Progress.Json.ObjectModel.JsonObject().
jsonData:Add('state', 'DISABLED').

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
```

```
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Put('http://localhost:16680/oemanager/applications/pas1/webapps/ROOT/transports/soap/oeservices/Simp
 jsonData)
      :ContentType('application/vnd.progress+json')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Response body example

```
{
 "result": "",
 "operation": "SET SOAP TRANSPORT PROPERTIES",
 "versionStr": null,
 "versionNo": 1,
 "errmsg": "",
 "outcome": "SUCCESS"
}
```

Note: The service runtime property “serviceAvailable” is located in *service_name*.props file. For example, *instance-name*\webapps\ROOT\WEB-INF\adapters\soap\PFT\PFT.props.

Remove a service (SOAP/REST)

Description

Remove a specified SOAP or REST service from a web application deployed in an ABL application.

HTTP Operation

DELETE

URI

host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*web_app_name*/transports/{soap | rest}/oeservices{*SoapServiceName* | *RestSevviceName*}

Content-Type

application/vnd.progress+json

Response codes

200 Success
500 Unexpected Server Error

Request body

NA

Command-line example (SOAP)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/soap/oeservices/SimpleTe
```

Response body example (SOAP)

```
{
  "result":
  {
  "OESoapServiceDesc": [
      {
      "accessUrl": "http://localhost:8810/soap/wsdl?targetURI=urn:nsRL1",
      "archiveName": "simpleRL.wsm",
      "archiveLocation":"/webapps/ROOT/WEB-INF/adapters/soap",
      "oetype":"SOAP_DESCRIPTOR",
      "name":"simpleRL",
      "state":"DISABLED",
      "type":"OPENEDGE",
      "version":"12.0.0",
      "description": "PAS OpenEdge SOAP Service Artifact",
      "securedUri":""
      }
  ]
},
"operation": "UNDEPLOY SOAP TRANSPORT DESCRIPTOR",
"versionStr": null,
"versionNo": 1,
"errmsg": "",
"outcome": "SUCCESS"
}
```

Command-line example (REST)

```
curl -X DELETE -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices/_oepingS
```

Response body example (REST)

```
{"result":
  {
    "OERestServiceDesc":[
      {
        "status":"INITIALIZED",
        "contextPath":"\/\/wrk\/pas1\/webapps\/ROOT",
        "descriptors":[
          {

"archiveLocation":"WEB-INF\/adapters\/rest\/_oepingService\/_oepingService.paar",
            "archiveName":"_oepingService.paar",
            "accessUrl":"\/rest\/_oepingService",
            "oetype":"REST_DESCRIPTOR",
            "version":"11.5.0",
```

```
            "description":"PAS OpenEdge REST Service Descriptor",
            "name":"_oepingService",
            "type":"OPENEDGE",
            "uri":"http:\/\/localhost.com:16680\/rest\/_oepingService"
          }
        ],
        "oetype":"REST",
        "version":"v11.4.0 (12-Dec-2013)",
        "description":"PAS OpenEdge REST Transport.",
        "name":"REST",
        "state":"DISABLED",
        "type":"OPENEDGE",
        "uri":"http:\/\/localhost.com:16680\/rest"
      }
    ]
  },
  "operation":"UNDEPLOY REST TRANSPORT DESCRIPTOR",
  "versionStr":"PASOE 11.5.0",
  "versionNo":1,
  "outcome":"SUCCESS",
  "errmsg":""
}
```

ABL HTTP client example (SOAP or REST)

Note: Substitute `rest` for `soap` in the URI below if you want to get metrics for the `rest` transport.

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder

:Delete('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT/transports/rest/oeservices

      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))
      :Request
   ).
```

Application management

The following topics describe the REST services that are available for managing ABL applications and ABL web applications based on `oeabl.war` that are deployed on PAS for OpenEdge.

See also

List ABL application information on page 140
Get information about a specific ABL application on page 142
List web applications deployed to an ABL application on page 144
Get information about a specific web application on page 147

List ABL application information

Description

Get information about ABL applications and deployed ABL web applications from the OEManager.

HTTP Operation

GET

URI

//host_name:*port*/oemanager/applications

Media type

application/vnd.progress+json

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v http://localhost:16680/oemanager/applications
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Response body example

```
{
   "operation":"GET OEABL SERVICES",
   "outcome":"SUCCESS",
   "result":{
```

```
      "Application":[
         {
            "name":"oepas1",
            "version":"v12.1.0 ( 2019-08-07 )",
            "description":"PAS Application",
            "type":"OPENEDGE",
            "webapps":[
               {
                  "name":"ROOT",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/",
                  "securedUri":"https://localhost:16681/",
                  "description":"PAS OpenEdge WebApp",
                  "type":"OPENEDGE",
                  "state":"STARTED",
                  "transports":[
                     {
                        "name":"APSV",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/apsv",
                        "securedUri":"https://localhost:16681/apsv",
                        "description":"PAS OpenEdge APSV Transport.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "status":"INITIALIZED",
                        "oetype":"APSV"
                     },
                     {
                        "name":"SOAP",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/soap",
                        "securedUri":"https://localhost:16681/soap",
                        "description":"PAS OpenEdge SOAP Service.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "status":"INITIALIZED",
                        "descriptors":[

                        ],
                        "oetype":"SOAP"
                     },
                     {
                        "name":"REST",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/rest",
                        "securedUri":"https://localhost:16681/rest",
                        "description":"PAS OpenEdge REST Transport.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "descriptors":[
                           {
                              "name":"_oepingService",
                              "version":"v12.1.0 ( 2019-08-07 )",
                              "uri":"http://localhost:16680/rest/_oepingService",
                            "securedUri":"https://localhost:16680/rest/_oepingService",

                              "description":"PAS OpenEdge REST Service Descriptor",
                              "type":"OPENEDGE",

"archiveLocation":"WEB-INF/adapters/rest/_oepingService/_oepingService.paar",
                              "accessUrl":"/rest/_oepingService",
                              "archiveName":"_oepingService.paar",
                              "oetype":"REST_DESCRIPTOR"
                           }
                        ],
                        "status":"INITIALIZED",
                        "contextPath":"C:/OpenEdge/WRK/oepas1/webapps\\ROOT",
                        "oetype":"REST"
                     },
```

```
                    {
                       "name":"WEB",
                       "version":"v12.1.0 ( 2019-08-07 )",
                       "uri":"http://localhost:16680/web",
                       "securedUri":"https://localhost:16681/web",
                       "description":"PAS OpenEdge WebSpeed Transport.",
                       "type":"OPENEDGE",
                       "state":"ENABLED",
                       "status":"INITIALIZED",
                       "oetype":"WEB"
                    }
                 ],
                 "applicationName":"oepas1",
                 "oetype":"WEBAPP"
              }
           "oetype":"APPLICATION"
        }
     ]
  },
  "errmsg":"",
  "versionStr":"v12.1.0 ( 2019-08-07 )",
  "versionNo":1
}
```


Get information about a specific ABL application

Description

Get information about a specified ABL application from the OEManager.

HTTP Operation

`GET`

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*

Media type

`application/vnd.progress+json`

Request body example

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v http://localhost:16680/oemanager/applications/oepas1
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Response body example


```
{
   "operation":"GET OEABL SERVICE",
   "outcome":"SUCCESS",
   "result":{
      "Application":[
         {
            "name":"oepas1",
            "version":"v12.1.0 ( 2019-08-07 )",
            "description":"PAS Application",
            "type":"OPENEDGE",
            "webapps":[
               {
                  "name":"ROOT",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/",
                  "securedUri":"https://localhost:16681/",
                  "description":"PAS OpenEdge WebApp",
                  "type":"OPENEDGE",
                  "state":"STARTED",
                  "transports":[
                     {
                        "name":"APSV",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/apsv",
                        "securedUri":"https://localhost:16681/apsv",
                        "description":"PAS OpenEdge APSV Transport.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "status":"INITIALIZED",
                        "oetype":"APSV"
                     },
                     {
                        "name":"SOAP",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/soap",
                        "securedUri":"https://localhost:16681/soap",
                        "description":"PAS OpenEdge SOAP Service.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "status":"INITIALIZED",
                        "descriptors":[

                        ],
                        "oetype":"SOAP"
                     },
                     {
                        "name":"REST",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/rest",
```

```
                        "securedUri":"https://localhost:16681/rest",
                        "description":"PAS OpenEdge REST Transport.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "descriptors":[
                           {
                              "name":"_oepingService",
                              "version":"v12.1.0 ( 2019-08-07 )",
                              "uri":"http://localhost:16680/rest/_oepingService",
                            "securedUri":"https://localhost:16680/rest/_oepingService",

                              "description":"PAS OpenEdge REST Service Descriptor",
                              "type":"OPENEDGE",

"archiveLocation":"WEB-INF/adapters/rest/_oepingService/_oepingService.paar",
                              "accessUrl":"/rest/_oepingService",
                              "archiveName":"_oepingService.paar",
                              "oetype":"REST_DESCRIPTOR"
                           }
                        ],
                        "status":"INITIALIZED",
                        "contextPath":"C:/OpenEdge/WRK/oepas1/webapps\\ROOT",
                        "oetype":"REST"
                     },
                     {
                        "name":"WEB",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/web",
                        "securedUri":"https://localhost:16681/web",
                        "description":"PAS OpenEdge WebSpeed Transport.",
                        "type":"OPENEDGE",
                        "state":"ENABLED",
                        "status":"INITIALIZED",
                        "oetype":"WEB"
                     }
                  ],
                  "applicationName":"oepas1",
                  "oetype":"WEBAPP"
               }
            "oetype":"APPLICATION"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```


List web applications deployed to an ABL application

Description

Get information from the OEManager about all the web applications deployed to an ABL application.

HTTP Operation

`GET`

URI

`//`*`host_name`*`:`*`port`*`/oemanager/applications/`*`ABL_app_name`*`/webapps`

Media type

```
application/vnd.progress+json
```

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/webapps')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Response body example


```
{
   "operation":"GET OEABL SERVICES",
   "outcome":"SUCCESS",
   "result":{
      "WebApp":[
         {
            "name":"ROOT",
            "version":"v12.1.0 ( 2019-08-07 )",
            "uri": "http://localhost:16680",
            "securedUri": "https://localhost:16681",
            "description": "PAS OpenEdge WebApp",
            "type": "OPENEDGE",
            "state": "STARTED"
            "transports":[
               {
                  "name":"APSV",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/apsv",
                  "securedUri":"https://localhost:16681/apsv",
                  "description":"PAS OpenEdge APSV Transport.",
                  "type":"OPENEDGE",
                  "state":"ENABLED",
                  "status":"INITIALIZED",
                  "oetype":"APSV"
```

```
                },
                {
                   "name":"SOAP",
                   "version":"v12.1.0 ( 2019-08-07 )",
                   "uri":"http://localhost:16680/soap",
                   "securedUri":"https://localhost:16681/soap",
                   "description":"PAS OpenEdge SOAP Service.",
                   "type":"OPENEDGE",
                   "state":"ENABLED",
                   "status":"INITIALIZED",
                   "descriptors":[

                   ],
                   "oetype":"SOAP"
                },
                {
                   "name":"REST",
                   "version":"v12.1.0 ( 2019-08-07 )",
                   "uri":"http://localhost:16680/rest",
                   "securedUri":"https://localhost:16681/rest",
                   "description":"PAS OpenEdge REST Transport.",
                   "type":"OPENEDGE",
                   "state":"ENABLED",
                   "descriptors":[
                      {
                         "name":"_oepingService",
                         "version":"v12.1.0 ( 2019-08-07 )",
                         "uri":"http://localhost:16680/rest/_oepingService",
                         "securedUri":"https://localhost:16680/rest/_oepingService",
                         "description":"PAS OpenEdge REST Service Descriptor",
                         "type":"OPENEDGE",

"archiveLocation":"WEB-INF/adapters/rest/_oepingService/_oepingService.paar",
                         "accessUrl":"/rest/_oepingService",
                         "archiveName":"_oepingService.paar",
                         "oetype":"REST_DESCRIPTOR"
                      }
                   ],
                   "status":"INITIALIZED",
                   "contextPath":"C:/OpenEdge/WRK/oepas1/webapps\\ROOT",
                   "oetype":"REST"
                },
                {
                   "name":"WEB",
                   "version":"v12.1.0 ( 2019-08-07 )",
                   "uri":"http://localhost:16680/web",
                   "securedUri":"https://localhost:16681/web",
                   "description":"PAS OpenEdge WebSpeed Transport.",
                   "type":"OPENEDGE",
                   "state":"ENABLED",
                   "status":"INITIALIZED",
                   "oetype":"WEB"
                }
             ],
             "applicationName":"oepas1",
             "oetype":"WEBAPP"
             }
          }
       ]
  },
  "errmsg":"",
  "versionStr":"v12.1.0 ( 2019-08-07 )",
  "versionNo":1
}
```

Get information about a specific web application

Description

Get information from the OEManager about a specific web application deployed to an ABL application.

HTTP Operation

GET

URI

//host_name:*port*/oemanager/applications/*ABL_app_name*/webapps/*Web_app_name*

Media type

application/vnd.progress+json

Request body

NA

Response codes

```
200 Success
500 Unexpected Server Error
```

Command-line example

```
curl -X GET -u username:password -v
http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT
```

ABL HTTP client example

```
define variable resp as OpenEdge.Net.HTTP.IHttpResponse.

resp = OpenEdge.Net.HTTP.ClientBuilder:Build():Client
   :Execute(OpenEdge.Net.HTTP.RequestBuilder
      :Get('http://localhost:16680/oemanager/applications/oepas1/webapps/ROOT')
      :UsingCredentials(new OpenEdge.Net.HTTP.Credentials('', 'username', 'password'))

      :Request
   ).
```

Response body example

```
{
   "operation":"GET OEABL SERVICE",
   "outcome":"SUCCESS",
```

```
   "result":{
      "WebApp":[
         {
            "name":"ROOT",

            "version":"v12.1.0 ( 2019-08-07 )",
            "uri":"http://localhost:16680",
            "securedUri":"https://localhost:16681",
            "description":"PAS OpenEdge WebApp",
            "type":"OPENEDGE",
            "state":"STARTED",
            "transports":[
               {
                  "name":"APSV",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/apsv",
                  "securedUri":"https://localhost:16681/apsv",
                  "description":"PAS OpenEdge APSV Transport.",
                  "type":"OPENEDGE",
                  "state":"ENABLED",
                  "status":"INITIALIZED",
                  "oetype":"APSV"
               },
               {
                  "name":"SOAP",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/soap",
                  "securedUri":"https://localhost:16681/soap",
                  "description":"PAS OpenEdge SOAP Service.",
                  "type":"OPENEDGE",
                  "state":"ENABLED",
                  "status":"INITIALIZED",
                  "descriptors":[

                  ],
                  "oetype":"SOAP"
               },
               {
                  "name":"REST",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/rest",
                  "securedUri":"https://localhost:16681/rest",
                  "description":"PAS OpenEdge REST Transport.",
                  "type":"OPENEDGE",
                  "state":"ENABLED",
                  "descriptors":[
                     {
                        "name":"_oepingService",
                        "version":"v12.1.0 ( 2019-08-07 )",
                        "uri":"http://localhost:16680/rest/_oepingService",
                        "securedUri":"https://localhost:16681/rest/_oepingService",
                        "description":"PAS OpenEdge REST Service Descriptor",
                        "type":"OPENEDGE",

"archiveLocation":"WEB-INF/adapters/rest/_oepingService/_oepingService.paar",
                        "accessUrl":"/rest/_oepingService",
                        "archiveName":"_oepingService.paar",
                        "oetype":"REST_DESCRIPTOR"
                     }
                  ],
                  "status":"INITIALIZED",
                  "contextPath":"C:/OpenEdge/WRK/oepas1/webapps\\ROOT",
                  "oetype":"REST"
               },
               {
                  "name":"WEB",
                  "version":"v12.1.0 ( 2019-08-07 )",
                  "uri":"http://localhost:16680/web",
                  "securedUri":"https://localhost:16681/web",
```

```
                  "description":"PAS OpenEdge WebSpeed Transport.",
                  "type":"OPENEDGE",
                  "state":"ENABLED",
                  "status":"INITIALIZED",
                  "oetype":"WEB"
               }
            ],
            "applicationName":"oepas1",
            "oetype":"WEBAPP"
         }
      ]
   },
   "errmsg":"",
   "versionStr":"v12.1.0 ( 2019-08-07 )",
   "versionNo":1
}
```

4

OEJMX Query Reference

System administrators can monitor and manage PAS for OpenEdge instances through scripts using the OpenEdge JMXcommand line utility, OEJMX , to connect to a local PAS for OpenEdge instance. Based on JMX (Java Management Extensions), OEJMX connects to an instance using Java objects called Managed Beans, MBeans or simply beans. One or more beans instrument a resource using attributes and operations. This direct access to the underlying beans provides administrators with the freedom to write scripts to monitor and manage applications in ways that may not be available within OpenEdge Management or JConsole.

For details about OEJMX usage, see the 'Use OEJMX to manage and monitor and instance' in *Manage Progress Application Server (PAS) for OpenEdge*. This topic details some common OEJMX queries as reference information.

For details, see the following topics:

- Get agents query
- Get active session information query
- Get a stack trace for a session query
- Get metrics for an ABL application query
- Reset metrics for an ABL application query
- Get hung client connections for an ABL application query
- Get agent metrics query
- Get agent session metrics query
- Get active threads from Tomcat thread pool query
- Get transport metrics for an ABL web application query

- Reset transport metrics for an ABL web application query
- Refresh agent query
- Refresh all agents query
- Flush deferred log buffer to agent log file query
- Reset deferred log buffer query
- Get dynamic ABL session limit query
- Update dynamic ABL session limit query
- Reset dynamic ABL session limit query
- Enable or disable ABL object tracking query
- Check ABL object tracking status query
- Get a report of tracked ABL objects query

Get agents query

Get agents from an ABL application

When to use:

To identify agents in the ABL application.

Query:

```
# Tokens: APP_NAME
# Get agents in an ABL application
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getAgents","oepas1"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`.

Result:

The output provides details about the agents including agent ID, PID, and state information. The agent ID can be used in other OEJMX queries such as to refresh an agent in an ABL application.

```
{
  "getAgents": {
    "agents": [
      {
        "agentId": "LMufO6dzRiSguShJOsJ53Q",
        "pid": "13992",
        "state": "AVAILABLE"
      }
    ]
  }
}
```

Get active session information query

Get active session information

When to use:

To identify active sessions in order to gather further information about those sessions.

Query:

```
# Tokens: APP_NAME
# Get active sessions in an ABL application
{"O":"PASOE:type=OEManager,name=SessionManager","M":["getMetrics","oepas1"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`.

Result:

The output provides details about client connections including avgQueueDepth, maxConcurrentClients, maxQueueDepth, reads, requests and timesQueued. The maxQueueDepth can be used to determine if there are any bottlenecks causing the requests to be queued which could indicated areas to improve performance by reducing the queuing of requests.

```
{
  "getMetrics": {
    "accessTime": "2018-06-25T15:43:12.766-04:00",
    "avgQueueDepth": "0",
    "concurrentConnectedClients": "0",
    "maxConcurrentClients": "50",
    "maxQueueDepth": "0",
    "readErrors": "0",
    "reads": "30133",
    "requests": "15059",
    "reserveConnectionTimeouts": "0",
    "startTime": "2018-06-25T15:20:33.981-04:00",
    "timesQueued": "0",
    "type": "OE_BROKER",
    "writeErrors": "0",
    "writes": "30132"
  }
}
```

Get a stack trace for a session query

Get a stack trace for a session

When to use:

By supplying a specific *AGENT_ID* and *SESSION_ID*, the query can return a stack trace for the session.

Query:

```
# Tokens: AGENT_ID, SESSION_ID
# Get a stack trace for a session in an ABL application
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getSessionStacks","16952","7"]}
```

Note: In this example, the token *AGENT_ID* is replaced with 16952 and *SESSION_ID* is replaced with 7.

Result:

The output displays a list of the stack trace listing files that were in use by the session. Used in conjunction with the GetSessionMetrics to identify active sessions like **SessionID=7**, developers can target specific files for the **ACTIVE** session to investigate.

```
{
  "getSessionStacks": {
    "ABLStacks": [
      {
        "Status": "Executing",
        "Databases": "",
        "BrokerSessionId": "A8A564D83461D3718EFA71A19C055C0D793DA292F6DF.oepas",
        "Callstack": [
          {
            "Line": 50,
            "Routine": "GetLogistics.p",
            "Source": "GetLogistics.p"
          },
          {
            "Line": 3,
            "Routine": "BalanceReport.p",
            "Source": "BalanceReport.p"
          }
        ],
        "StartupParams": "-pf /usr/Progress/dlc/startup.pf,(end .pf),-logginglevel
2,-logfile /usr/Progress/wrk/oepas/logs/oepas.agent.log,-uburl
AppServerDC://0.0.0.0:43420/,-logname oepas,-logentrytypes
ASPlumbing,DB.Connects,-ubpropfile /usr/Progress/wrk/oepas/conf/openedge.properties,-ASID
 1,-ipver IPv4,-sectok XXXXXXXXXXXXXXXXXXXXX,-T /usr/Progress/wrk/oepas/temp",
        "Propath": "propath entries",
        "AgentSessionId": 7,
        "OO4GLObjs": [
          {
            "ObjectId": "001026",
            "Class": "OpenEdge/Testing/GetInfo.cls"
          },
          {
            "ObjectId": "STATIC",
            "Class": "OpenEdge/Testing/APIClass.cls"
          },
          {
            "ObjectId": "STATIC",
            "Class": "OpenEdge/Testing/GetInfo.cls"
          },
          {
            "ObjectId": "001000",
            "Class": "OpenEdge/Rest/Admin/AppServerStatus.r"
          }
        ],
        "PersProcs": ""
      }
    ]
  }
}
```

Get metrics for an ABL application query

Get metrics for a specific ABL application

When to use:

Session Manager metrics provide information about the number of requests that were executed/failed/queued while running on a specific ABL Application.

For more information about PAS for OpenEdge Session Manager metrics, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

Query:

```
# Tokens: APP_NAME
# Get session manager metrics from an ABL application
{"O":"PASOE:type=OEManager,name=SessionManager","M":["getMetrics","oepas1"]
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`.

Result:

The output demonstrates the number of **requests** and in this example the `numReserveABLSessionTimeouts` is 0, which indicates that no clients were ever timed out. There were also no read or write errors. This indicates that the server is running well.

```
{
   "getMetrics":{
      "maxConcurrentClients":"50",
      "concurrentConnectedClients":"0",
      "numReserveABLSessionTimeouts":"0",
      "writeErrors":"0",
      "reads":"30133",
      "numReserveABLSessionWaits":"0",
      "startTime":"2020-01-14T11:50:24.634-05:00",
      "requests":"15059",
      "type":"OE_BROKER",
      "writes":"30132",
      "readErrors":"0",
      "accessTime":"2020-01-14T11:51:16.703-05:00"
   }
}
```

Reset metrics for an ABL application query

Reset metrics for a specific ABL application

When to use:

Reset the Session Manager metrics to collect new information about the number of requests that were executed/failed/queued while running on a specific ABL Application.

For more information about PAS for OpenEdge Session Manager metrics, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

Query:

```
# Tokens: APP_NAME
# Get session manager metrics from an ABL application
{"O":"PASOE:type=OEManager,name=SessionManager","M":["resetMetrics","oepas1"]
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`.

Result:

The response is returned as null.

```
{"resetMetrics":null}
```

If you run a Get metrics for an ABL application query on page 155 on the session manager, the metrics collection start time is from after the reset call was made (as seen in the `lastResetTime` property).

Get hung client connections for an ABL application query

Get hung client connections for a specific ABL application

When to use:

Session Manager metrics provides information about the number of requests that are still running in a given ABL Application to identify clients connections that are hung. In this query you can specify the number of miliseconds you want to elapse before sending information back about hung client connections.

Query:

```
# Tokens: APP_NAME
# Get session manager metrics about hung client connections from an ABL application
{"O":"PASOE:type=OEManager,name=SessionManager","M":["getHungClientConnections","oepas1",6000]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and `6000` is the minimum elapsed time in miliseconds to be reported.

Result:

The `elapsedTimeMs` can be used to help identify long running requests which could indicated a hung client connection.

```
{
  "getHungClientConnections": [
    {
      "httpSessionId": "",
      "reqStartTimeStr": "2018-06-25T15:32:48.139-0400",
      "clientName": "172.16.21.106",
      "executerThreadId": "thd-5",
      "requestID": "3fj1RO35XYk/FDkZSN5UUw",
      "requestProcedure": "retrieveAccnts.p",
      "requestUrl": "http://localhost:8810/apsv",
      "sessionID": "C4F5EA614E9B1F8D94E28D7A87185DA923C4EE22F204.oepas1",
      "adapterType": "APSV",
      "elapsedTimeMs": 35298
    },
    {
      "httpSessionId": "",
      "reqStartTimeStr": "2018-06-25T15:32:47.667-0400",
      "clientName": "localhost",
      "executerThreadId": "thd-1",
      "requestID": "rw4IDagih4Q/FDkZAMm8TQ",
      "requestProcedure": "mycustomer.p",
      "requestUrl": "http://localhost:8810/apsv",
      "sessionID": "9E0DDCB1CF61036B5D52587FA41CDEAAE9C8D8713B59.oepas1",
      "adapterType": "APSV",
      "elapsedTimeMs": 35771
    },
    {
      "httpSessionId": "",
      "reqStartTimeStr": "2018-06-25T15:33:11.131-0400",
      "clientName": "localhost",
      "executerThreadId": "thd-1",
      "requestID": "gBR4KKyWS6E/FDoZuBeYZQ",
      "requestProcedure": "getlist.p",
      "requestUrl": "http://localhost:8810/apsv",
      "sessionID": "13512413B5FE9737F0180B7439EF7921EA0CDBA52753.oepas1",
      "adapterType": "APSV",
      "elapsedTimeMs": 12307
    }
  ]
}
```

Get agent metrics query

Get information about an agent

When to use:

To get agent metrics including state of the active threads, connections, memory consumed, and session information.

Query:

```
# Tokens: AGENT_ID
# Get agent metrics
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getAgentMetrics","16952"]}
```

Note: In this example, the token *AGENT_ID* is replaced with 16952.

Result:

Returns agent metrics:


```
{
  "getAgentMetrics": {
    "AgentStatHist": [
      {
        "ActiveThreads": 1,
        "OpenConnections": 1,
        "ExitedSessions": 0,
        "ExitedThreads": 0,
        "CStackMemory": 2097152,
        "ClosedConnections": 0,
        "OverheadMemory": -7015299910500,
        "ActiveSessions": 2
      }
    ]
  }
}
```


Get agent session metrics query

Get agent session metrics

When to use:

Agent Session Metrics provides state of the session, memory consumed, session-Id, time it was started for each agent session.

Query:

```
# Tokens: AGENT_ID
# Get agent session metrics
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getSessionMetrics","16952"]}
```

Note: In this example, the token *AGENT_ID* is replaced with 16952.

Result:

The output demonstrates information for all sessions (**4 & 7**). Information includes SessionState, ConnectionId, SessionMemory, EndTime, ThreadId, SessionExternalState, StartTime and SesssionId.

```
{
  "getSessionMetrics": {
    "AgentSession": [
      {
        "SessionState": "IDLE",
        "ConnectionId": "org.codehaus.jettison.json.JSONObject$Null",
        "SessionMemory": 759943,
        "EndTime": "org.codehaus.jettison.json.JSONObject$Null",
        "ThreadId": -1,
        "SessionExternalState": 0,
        "StartTime": "2018-07-11T13:33:45.491",
        "SessionId": 4
      },
      {
        "SessionState": "IDLE",
        "ConnectionId": "org.codehaus.jettison.json.JSONObject$Null",
        "SessionMemory": 6458674,
        "EndTime": "org.codehaus.jettison.json.JSONObject$Null",
        "ThreadId": -1,
        "SessionExternalState": 0,
        "StartTime": "2018-07-11T13:33:45.491",
        "SessionId": 7
      }
    ]
  }
}
```

Get active threads from Tomcat thread pool query

Get active threads from Tomcat thread pool

When to use:

Gets information about the active threads for a given Tomcat thread pool

Query:

```
# Tokens: APP_NAME, AGENT_ID, AGENT_PID, SESSION_ID
# Get active threads from Tomcat thread Pool
{"O":"Catalina:type=Executor,name=tomcatThreadPool","A":"activeCount"}
```

Note: For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns **10** as activeCount.

```
{"activeCount":10}
```

Get transport metrics for an ABL web application query

Get transport metrics for an ABL web application

Transport metrics are available for the APSV, SOAP, REST, and WEB transports in an ABL web application. Each transport metric query is described in this topic.

For more information about PAS for OpenEdge transport metrics, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

Get APSV Transport metrics for a given ABL web application

When to use:

Gets the metrics for the requests that were executed on the APSV transport. It returns the number of requests that were connected, disconnected, failing, and more.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Get APSV Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=ApsvTransportManager","M":["getMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns the APSV transport metrics for a given OEABL web application.

```
{
    "getMetrics": {
        "disconnectRequests": "15",
        "connectErrors": "0",
        "disconnectErrors": "0",
        "sessionErrors": "0",
        "startTime": "2018-07-11T08:17:35.329-04:00",
        "sessionRequests": "39",
        "forbiddenErrors": "0",
        "type": "OE_APSV_TRANSPORT",
        "connectRequests": "15",
        "accessTime": "2018-07-11T13:11:48.554-04:00",
        "statusRequests": "0"
    }
}
```

Get REST Transport metrics for a given ABL web application

When to use:

Gets the metrics for the requests that were executed on the REST transport. It returns the number of requests that were connected, disconnected, failing, and more.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Get REST Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=RestTransportManager","M":["getMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns the REST transport metrics for a given OEABL web application.


```
{
   "getMetrics":{
      "stdDevConnectTime":"0",
      "minSessionTime":"0",
      "serviceNotFoundErrors":"0",
      "minConnectTime":"0",
      "requests":"0",
      "type":"OE_REST_TRANSPORT",
      "expressionErrors":"0",
      "maxSessionTime":"0",
      "maxConnectTime":"0",
      "runRequests":"0",
      "successfulRunRequests":"0",
      "startTime":"2020-01-14T12:23:00.686-05:00",
      "failedRequests":"0",
      "stdDevSessionTime":"0",
      "minDisconnectTime":"0",
      "successfulConnectRequests":"0",
      "maxDisconnectTime":"0",
      "serviceUnavailableRequests":"0",
      "avgSessionTime":"0",
      "avgDisconnectTime":"0",
      "successfulRequests":"0",
      "stdDevDisconnectTime":"0",
      "avgConnectTime":"0",
      "connectRequests":"0",
      "accessTime":"2020-01-14T12:44:23.666-05:00",
      "statusRequests":"0"
   }
}
```


Get SOAP Transport metrics for a given ABL web application

When to use:

Gets the metrics for the requests that were executed on the SOAP transport. It returns the number of requests that were connected, disconnected, failing, and more.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME, SOAP_SERVICE_NAME
# Get SOAP Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=SoapTransportManager","M":["getMetrics","oepas1","ROOT","soapSvc"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, *WEB_APP_NAME* is replaced with `ROOT`, and *SOAP_SERVICE_NAME* is replaced with `soapSvc`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns the SOAP transport metrics for a given SOAP service.

```
{
   "getMetrics":{
      "objPoolFullErrors":"0",
      "appObjects":"0",
      "clientErrors":"0",
      "requests":"0",
      "appserverConnections":"0",
      "subAppObjects":"0",
      "ABLServerErrors":"0",
      "faults":"0",
      "activeRequests":"0",
      "objPoolExpiredErrors":"0",
      "otherErrors":"0",
      "ABLClientErrors":"0",
      "procObjects":"0",
      "connectErrors":"0",
      "nameserverErrors":"0",
      "startup":"Tue Jan 14 12:45:49 EST 2020",
      "ABLAppErrors":"0",
      "providerErrors":"0",
      "lastReset":"Tue Jan 14 12:45:49 EST 2020",
      "objNotFoundErrors":"0"
   }
}
```

Get WEB Transport metrics for a given ABL web application

When to use:

Gets the metrics for the requests that were executed on the WEB transport. It returns the number of requests that were connected, disconnected, failing, and more.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Get WEB Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=WebTransportManager","M":["getMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns the WEB transport metrics for a given OEABL web application.


```
{
   "getMetrics":{
      "maxABLConnectTime":"0",
      "deleteErrors":"0",
      "optionsRequests":"0",
      "servletRequests":"0",
      "stdABLDevConnectTime":"0",
      "type":"OE_OEWEB_TRANSPORT",
      "getErrors":"0",
      "patchRequests":"0",
      "postRequests":"0",
      "ablConnectErrors":"0",
      "stdDevABLProcessingTime":"0",
      "getRequests":"0",
      "minABLProcessingTime":"0",
      "headRequests":"0",
      "startTime":"2020-01-14T12:23:00.688-05:00",
      "traceRequests":"0",
      "failedServletRequests":"0",
      "successfulServletRequests":"0",
      "patchErrors":"0",
      "maxABLProcessingTime":"0",
      "postErrors":"0",
      "minABLConnectTime":"0",
      "ablRuntimeErrors":"0",
      "optionsErrors":"0",
      "traceErrors":"0",
      "putRequests":"0",
      "headErrors":"0",
      "avgABLProcessingTime":"0",
      "connectRequests":"0",
      "putErrors":"0",
      "deleteRequests":"0",
      "accessTime":"2020-01-14T12:47:51.057-05:00",
      "avgABLConnectTime":"0"
   }
}
```


Reset transport metrics for an ABL web application query

Reset transport metrics for an ABL web application

Reset APSV, SOAP, REST, and WEB transport metrics in an ABL web application to collect new information about the number of requests that were executed/failed/queued while running on a specific ABL Application.

For more information about PAS for OpenEdge transport metrics, see 'Collect runtime metrics' in *Manage Progress Application Server (PAS) for OpenEdge*.

Reset APSV Transport metrics for a given ABL web application

When to use:

Reset the metrics for the requests that were executed on the APSV transport.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Reset APSV Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=ApsvTransportManager","M":["resetMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

The response is returned as null.

```
{"resetMetrics":null}
```

If you run a Get transport metrics for an ABL web application query on page 160 on the APSV transport, the metrics collection start time is from after the reset call was made (as seen in the `lastResetTime` property).

Reset REST Transport metrics for a given ABL web application

When to use:

Resets the metrics for the requests that were executed on the REST transport.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Reset REST Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=RestTransportManager","M":["resetMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

The response is returned as null.

```
{"resetMetrics":null}
```

If you run a Get transport metrics for an ABL web application query on page 160 on the REST transport, the metrics collection start time is from after the reset call was made (as seen in the `lastResetTime` property).

Reset SOAP Transport metrics for a given ABL web application

When to use:

Resets the metrics for the requests that were executed on the SOAP transport.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME, SOAP_SERVICE_NAME
# Reset SOAP Transport metrics for the given ABLApp, WebApp, and SoapSVC
{"O":"PASOE:type=OEManager,name=SoapTransportManager","M":["resetMetrics","oepas1","ROOT","soapSvc"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, *WEB_APP_NAME* is replaced with `ROOT`, and *SOAP_SERVICE_NAME* is replaced with `soapSvc`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

The response is returned as null.

```
{"resetMetrics":null}
```

If you run a Get transport metrics for an ABL web application query on page 160 on the SOAP transport, the metrics collection start time is from after the reset call was made (as seen in the `lastResetTime` property).

Reset WEB Transport metrics for a given ABL web application

When to use:

Resets the metrics for the requests that were executed on the WEB transport.

Query:

```
# Tokens: APP_NAME, WEB_APP_NAME
# Reset WEB Transport metrics for the given ABLApp and WebApp
{"O":"PASOE:type=OEManager,name=WebTransportManager","M":["resetMetrics","oepas1","ROOT"]}
```

Note: In this example, the token *APP_NAME* is replaced with `oepas1`, and *WEB_APP_NAME* is replaced with `ROOT`. For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

The response is returned as null.

```
{"resetMetrics":null}
```

If you run a Get transport metrics for an ABL web application query on page 160 on the WEB transport, the metrics collection start time is from after the reset call was made (as seen in the `lastResetTime` property).

Refresh agent query

Refresh a specific agent in an ABL application

When to use:

The refresh agent query terminates all sessions of a specified multi-session agent within a ABL application, thereby refreshing the agent so that changes to the application can be picked-up by the agent. Supply the agent ID to refresh a specific agent.

To obtain the agent IDs of all agents running in your ABL application, you should first run a Get agents query on page 152.

For more information, see 'Refresh agents in an ABL application' in *Manage Progress Application Server (PAS) for OpenEdge*.

Query:

```
# Tokens: AGENT_ID
# Refresh a specific agent in an ABL application
{"O":"PASOE:type=OEManager,name=AgentManager","M":["refreshAgent","AGENT_ID"]}
```

Note: For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns messaging about the actions performed on the specified agent:

```
{
  "refreshAgent": {
    "agent": [
      {
        "agentId": "LdS8phtPR0GIrhRrrSL3Xg",
        "pid": "7672",
        "sessions": [
          {
            "sessionId": 4,
            "terminateResult": "success"
          },
          {
            "sessionId": 7,
            "terminateResult": "success"
          }
        ]
      }
    ]
  }
}
```

Refresh all agents query

Refresh all agents in an ABL application

When to use:

Refresh Agents terminates all sessions within an ABL application, thereby refreshing the multi-session agent or agents so that changes to the application can be picked-up by the agent or agents. Supply the ABL application name to refresh all agents in the ABL application. For more information, see 'Refresh agents in an ABL application' in *Manage Progress Application Server (PAS) for OpenEdge*.

Query:

```
# Tokens: APP_NAME
# Refresh Agents in an ABL application
{"O":"PASOE:type=OEManager,name=AgentManager","M":["refreshAgents","APP_NAME"]}
```

Note: For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns messaging about the actions performed on the agents:

```
{
  "refreshAgents": {
    "agents": [
      {
        "agentId": "LdS8phtPR0GIrhRrrSL3Xg",
        "pid": "7672",
        "sessions": [
          {
            "sessionId": 4,
            "terminateResult": "success"
          },
          {
            "sessionId": 7,
            "terminateResult": "success"
          }
        ]
      },
      {
        "agentId": "dz7DZReNRFCU5TIAxneVYA",
        "pid": "65681",
        "sessions": [
          {
            "sessionId": 4,
            "terminateResult": "success"
          },
          {
            "sessionId": 7,
            "terminateResult": "success"
          }
        ]
      }
    ]
  }
}
```

Flush deferred log buffer to agent log file query

Flush deferred log buffer to agent log file

When to use:

Writes the deferred log buffer of an agent or agents to the agent log file. For more information about deferred logging, see 'Use deferred logging in PAS for OpenEdge' in *Manage Progress Application Server (PAS) for OpenEdge*.

To obtain the agent IDs of all agents running in your ABL application, you should first run a Get agents query on page 152.

Query:

```
# Tokens: APP_NAME, AGENT_ID
# Refresh Agents in an ABL application
{"O":"PASOE:type=OEManager,name=AgentManager","M":["flushDeferredLog","APP_NAME","AGENT_ID"]}
```

Note: For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns messaging about the actions performed on the deferred log buffer:

```
{
"flushDeferredLog":{
  "flushDeferredLog":true
  }
}
```

Reset deferred log buffer query

Reset deferred log buffer

When to use:

Empties the deferred log buffer. For more information about deferred logging, see 'Use deferred logging in PAS for OpenEdge' in *Manage Progress Application Server (PAS) for OpenEdge*.

To obtain the agent IDs of all agents running in your ABL application, you should first run a Get agents query on page 152.

Query:

```
# Tokens: APP_NAME, AGENT_ID
# Reset deferred log buffer
{"O":"PASOE:type=OEManager,name=AgentManager","M":["resetDeferredLog","APP_NAME","AGENT_ID"]}
```

Note: For more information about using tokens, see 'OEJMX utility' in *Manage Progress Application Server (PAS) for OpenEdge*.

Result:

Returns messaging about the actions performed on the deferred log buffer:

```
{
"resetDeferredLog":{
  "resetDeferredLog":true
  }
}
```

Get dynamic ABL session limit query

Get dynamic ABL session limit

When to use:

To get the current in-memory `maxABLSessionsPerAgent` limit, which is represented by the property `dynmaxablsessions`.

Query:

```
# Tokens: ABL_APP_NAME, AGENT_PID
# Get dynamic ABL session limit
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getDynamicABLSessionLimit","oepas1","16952"]}
```

Note: In this example, the token *`AGENT_PID`* is replaced with `oepas1`, and the *`AGENT_PID`* is replaced with `16952`.

Result:

Returns dynamic ABL session limit:

```
{
   "getDynamicABLSessionLimit":[
      {
         "ABLOutput":{
            "numAvailableSessions":2,
            "dynmaxablsessions":120,
            "numABLSessions":2
         },
         "ABLReturnVal":true,
         "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
         "pid":"16952"
      }
   ]
}
```

Update dynamic ABL session limit query

Update dynamic ABL session limit

When to use:

To update the current in-memory `maxABLSessionsPerAgent` limit, which is represented by the property `dynmaxablsessions`.

Query:

```
# Tokens: ABL_APP_NAME, AGENT_PID, SESSON_LIMIT
# Get dynamic ABL session limit
{"O":"PASOE:type=OEManager,name=AgentManager","M":["setDynamicABLSessionLimit","oepas1","16952","151"]}
```

Note: In this example, the token *`AGENT_PID`* is replaced with `oepas1`, the *`AGENT_PID`* is replaced with `16952`, and the *`SESSION_LIMIT`* is replaced with `151`.

Result:

Returns the updated dynamic ABL session limit:

```
{
   "setDynamicABLSessionLimit":[
      {
         "ABLOutput":{
            "numAvailableSessions":2,
            "dynmaxablsessions":151,
            "numABLSessions":2
         },
         "ABLReturnVal":true,
         "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
         "pid":"16952"
      }
   ]
}
```

Reset dynamic ABL session limit query

Reset dynamic ABL session limit

When to use:

To reset the current in-memory `maxABLSessionsPerAgent` limit, which is represented by the property `dynmaxablsessions`, back to the value specified for `maxABLSessionsPerAgent` in `openedge.properties`.

Query:

```
# Tokens: ABL_APP_NAME, AGENT_PID
# Get dynamic ABL session limit
{"O":"PASOE:type=OEManager,name=AgentManager","M":["resetDynamicABLSessionLimit","oepas1","16952"]}
```

Note: In this example, the token *`AGENT_PID`* is replaced with `oepas1`, and the *`AGENT_PID`* is replaced with `16952`.

Result:

Returns the reset dynamic ABL session limit:

```
{
   "resetDynamicABLSessionLimit":[
      {
         "ABLOutput":{
            "numAvailableSessions":2,
            "dynmaxablsessions":200,
            "numABLSessions":2
         },
         "ABLReturnVal":true,
         "agentId":"iDpJxd-FTmyZ7CnX0IJpMw",
         "pid":"16952"
      }
   ]
}
```

Enable or disable ABL object tracking query

Enable ABL object tracking

When to use:

To enable or disable the tracking of ABL objects. ABL objects are tracked to identify potential memory leaks. To generate a report of these objects, see Get a report of tracked ABL objects query. When object tracking is not required, disable. Disabled by default.

Query:

```
# Tokens: AGENT_ID, SWITCH
# Enable or disable ABL object tracking
{"O":"PASOE:type=OEManager,name=AgentManager","M":["trackABLObjects","AGENT_ID",SWITCH]}
```

Note: In this example, the token *AGENT_ID* must be replaced with the application's agent ID or PID. Replace the token *SWITCH* with `true` to enable ABL object tracking, or `false` to disable it.

Result:

Returns a message that the update was received:

```
{"trackABLObjects":true}
```

Note: The `{"trackABLObjects":true}` message does not indicate the status of the ABL object tracking. To check if the tracking ABL object status is `true` or `false`, see Check ABL object tracking status query on page 172.

Check ABL object tracking status query

Check ABL object tracking status

When to use:

To check the status of ABL object tracking. To change the status, see Enable or disable ABL object tracking query on page 171.

Query:

```
# Tokens: AGENT_ID
# Check the status of ABL object tracking
{"O":"PASOE:type=OEManager,name=AgentManager","M":["trackingABLObjects","AGENT_ID"]}
```

Note: In this example, the token *AGENT_ID* must be replaced with the application's agent ID or PID.

Result:

Returns the status of ABL object tracking.

If ABL object tracking is enabled:

```
{"trackingABLObjects":true}
```

If ABL object tracking is disabled:

```
{"trackingABLObjects":false}
```

Get a report of tracked ABL objects query

Get a report of tracked ABL objects

When to use:

To get a report of ABL objects that may indicate potential memory leaks. Getting a report requires that object tracking is enabled. To change the status of ABL object tracking, see Enable or disable ABL object tracking query on page 171. After leaks have been identified and resolved, disable ABL object tracking.

Query:

```
# Tokens: AGENT_ID
# Get tracked ABL objects report
{"O":"PASOE:type=OEManager,name=AgentManager","M":["getABLObjectsReport","AGENT_ID"]}
```

Note: In this example, the token *AGENT_ID* must be replaced with the application's agent ID or PID.

Result:

Returns the report of the tracked ABL objects, for example:

```
{
   "getABLObjectsReport":{
      "ABLOutput":{
         "ABLObjects":[
            {
               "MemUsed":757793,
               "Objects":[
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33029,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33026,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33023,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33020,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33017,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33014,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33011,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33008,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
                     "Line":13
                  },
                  {
                     "ObjType":"MEMPTR",
                     "HandleId":33005,
                     "Size":50000,
                     "Source":"srvMemLeak.p",
```

```
      "Line":13
   },
   {
      "ObjType":"MEMPTR",
      "HandleId":33002,
      "Size":50000,
      "Source":"srvMemLeak.p",
      "Line":13
   },
   {
      "ObjType":"PROCEDURE",
      "Name":"srvMemLeak.p",
      "HandleId":33000,
      "Source":"SERVER.p",
      "Line":2
   },
   {
      "ObjType":"QUERY",
      "HandleId":33001,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33003,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33004,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33006,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33007,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33009,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33010,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33012,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
```

```
      "ObjType":"QUERY",
      "HandleId":33013,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33015,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33016,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33018,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33019,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33021,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33022,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33024,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33025,
      "Source":"srvMemLeak.p",
      "Line":12
   },
   {
      "ObjType":"Progress.Lang.Object",
      "Name":"Progress.Json.ObjectModel.JsonObject",
      "HandleId":33027,
      "Source":"srvMemLeak.p",
      "Line":14
   },
   {
      "ObjType":"QUERY",
      "HandleId":33028,
      "Source":"srvMemLeak.p",
      "Line":12
```

```
                    },
                    {
                       "ObjType":"Progress.Lang.Object",
                       "Name":"Progress.Json.ObjectModel.JsonObject",
                       "HandleId":33030,
                       "Source":"srvMemLeak.p",
                       "Line":14
                    }
                 ],
                 "AgentSessionId":4
              },
              {
                 "MemUsed":757793,
                 "Objects":"",
                 "AgentSessionId":7
              }
           ]
        },
        "ABLReturnVal":true,
        "agentId":"VK3SfrVPTW27mHpgylb-FQ",
        "pid":"32792"
     }
}
```