WebSphere Process Server
Business Process Choreographer

Process Cleanup Service Sample V2
Enhanced business process instance deletion

Susan Herrmann
IBM Development Lab Böblingen, Germany

Abstract

WebSphere Process Server provides an execution environment for business processes. Business processes can be defined so that they are persisted in WebSphere Process Servers Business Process Choreographer database. They can remain in the database after their completion.

This document describes an application that allows you to delete completed business process instances from the database. This cleanup can either be performed immediately or it can be scheduled to occur regularly.

This sample application is provided for download and includes the source code, so you can examine and adapt it.

Contents

Change History

Document Version	Date	Comments
V1.0	May 23, 2009	First edition

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
CICS®
Cloudscape®
DB2 Universal Database™
DB2®
developerWorks®
FFST™
HACMP™
IBM®
IMS™
Informix®
Lotus®
MQSeries®
Rational®
Redbooks®
Redbooks (logo)®
SupportPac™
Tivoli®
WebSphere®
z/OS®

The following terms are trademarks of other companies:
Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.
SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.
EJB, Enterprise JavaBeans, J2EE, J2SE, Java, JavaBeans, Javadoc, JavaScript, JavaServer, JDBC, JDK, JNI, JSP, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Access, ActiveX, Expression, Microsoft, SQL Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

1. Introduction

Business Process Choreographer (BPC) allows you to run business processes and human tasks within WebSphere Process Server.

Long-running business processes are stored in the Business Process Choreographer database. This database is intended to keep in-flight data, like uncompleted long-running process instances and uncompleted human tasks. These are instances which are currently executing, as well as instances that need manual resolution, such as ones that contain activities that are in the stopped state.

Completed instances should not be kept indefinitely in the Business Process Choreographer database because it can cause administration issues, such as exhausted disk space and reduced performance. Also, because the instance data in the database only reflects the current status of an instance, any visualization tool that uses this data, for example the BPC Explorer, can only display a snapshot of the instance data at a certain time. This means that the Business Process Choreographer runtime database is not suitable for auditing or reporting purposes.

There are several ways that you can delete completed instances from the Business Process Choreographer database. They are summarized in the following V6.2 information center page:

Cleanup procedures for Business Process Choreographer:
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.bpc.620.doc/doc/bpc/c2_cleanup.html

With WebSphere Process Server V6.2 a scheduled deletion service for business process instances and human task instances has been introduced. In order to provide similar functionality for earlier releases, the following Process Cleanup Service sample was provided. You can use this sample to schedule the deletion of completed business process instances from the Business Process Choreographer database:
Process Cleanup Service for Business Process Choreographer:
http://www-01.ibm.com/support/docview.wss?uid=swg27007816

This document discusses an enhanced version of this Process Cleanup Service sample. Like the original sample, this Process Cleanup Service provides a way to delete completed business process instances, related inline human tasks, and dependent child human tasks. However, this application cannot delete standalone human task. The enhancements over the original Process Cleanup Service are listed in section 3 – Enhancements.

2. Scope

This document assumes familiarity with general information about the original Process Cleanup Service sample. It is therefore recommended to make sure that you are familiar with the documentation available for the original service before reading this document.

Process Cleanup Service for Business Process Choreographer:
http://www-01.ibm.com/support/docview.wss?uid=swg27007816

The current Process Cleanup Service, like the original one, is a sample application and is therefore provided with as-is source code. This material contains programming source code for your consideration. The Process Cleanup Service sample has not been thoroughly tested under all conditions. IBM can therefore not guarantee or imply reliability, serviceability, or function of this program.

3. Enhancements

The following enhancements are addressed by this version of the Process Cleanup Service sample application.

3.1. Execution modes

Two modes of execution are now supported:

1) Daemon mode
   Runs as background process and performs business process deletion according to a predefined configuration. This is the default operating mode for cleaning up instances after the audit and tracing period.

2) Interactive mode
   Can be run by an administrator when required to perform unplanned business process instance maintenance to solve problems such as business processes failing because of external back-end unavailability or performance degradation.

The daemon mode has been enhanced. It can now select business process instances by end state and by template name for deletion.
The interactive mode has been newly introduced in this version of the Process Cleanup Service sample.

3.2. Deletion capabilities

In both execution modes, you can delete business process instances according to the following criteria:

- Delete all finished instances for a given business process template.
- Delete completed instances for a given business process template.
- Delete finished instances for a given business process template that are older than a given timestamp.
- Delete completed instances for a given business process template that are older than a given timestamp.

Furthermore, the tool enables the administrator to control its execution when running in interactive mode. After a deletion command has been started in interactive mode, the administrator has the option to stop it and start it again. This stop operation does not affect the daemon mode execution.

3.3. Display capabilities

In interactive mode, the tool can count the number of business process instances according to the following criteria. You can use these commands to identify how many instances are in the system and how many qualify for the deletion operations before you actually perform the deletion.

1) Counting business process instances for any template:
   - Display the number of all completed ('finished', 'terminated' and 'failed') business process instances.
   - Display how many business process instances are in the state 'finished', 'terminated' or 'failed' (limited to one state per query).
   - Display how many completed business process instances are older than a given timestamp.

2) Counting business process instances for a given template:
   - Display the number of all completed business process instances.
   - Display how many business process instances are in the state 'finished', 'terminated' or 'failed'.
   - Display how many completed business process instances are older than a given timestamp.

3.4. Tracking progress and logging

The progress of every single deletion operation is printed out. Also, the administrative actions fulfilled during interactive mode are logged.

1) Tracking progress after deletion started:
   - Number of instances deleted.

2) Logging:
   - Capability to log the administration activities.

4. Installation and configuration

To install and configure the Process Cleanup Service sample application, perform the following steps:

1) Customize the `processCleanupService.properties` file and store a copy of it in the properties directory of each WebSphere Process Server installation, where your deployment targets hosting the Business Flow Manager container are running.

2) Install the `ProcessCleanupServiceV2.ear` file and start the ProcessCleanupService application.

3) Store the `manageProcessCleanupCustomService.py` script in the directory that contains your administrative scripts.

4.1. Configuration

In your customized processCleanupService.properties file, in the *`install_home`*`/properties` directory of each WebSphere Process Server installation hosting servers or clusters that have a Business Process Choreographer configuration, you can define the following properties.

The following properties apply for both the daemon and interactive mode processing:

- **slice**
  This property specifies how many completed business process instances are deleted in one transaction. Larger values give better performance but need more transaction log space for the database. The default value is 20. Normally you should use values between 10 and 100.

- **maxRetryCount**
  This property specifies the maximum number of retries that will be done in case of errors during the deletion of business process instances during one Process Cleanup Service sample run. The value '0' means that there is no limit. If this property is not specified, the default value of ‘100’ is used.

- **enableSeparateLog**
  This property specifies whether the Process Cleanup Service sample additionally logs its actions to a separate cleanup service log file in the server’s logs directory.
  The default is false. Specifying 'true' enables this additional logging. In this case you will find a log file called `processCleanupSample#.log` in the servers log directory.
  The Process Cleanup Service sample uses the `java.logging` facilities. You can make changes to this logging implementation in the provided source code.

The following properties only apply for daemon mode processing:

- **frequency**
  This property specifies how often the cleanup service is run in daemon mode. It uses the 'crontab' format. See the information center documentation for the WebSphere scheduler for details about the format:

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.javadoc.doc/public_html/api/com/ibm/websphere/scheduler/UserCalendar.html

  Examples for a valid format are:

  - Every hour
    frequency=0 0 * * * ?
  - Every saturday at 2 am
    frequency=00 00 02 ? * SAT
  - Every last day in the month at 3 am
    frequency=00 00 03 L * ?

- **deleteOlderThan**
  This property specifies how long the completed business processes must stay in the database after they reach their end state before they can be deleted.

  format is:
  <minutes> <hours> <days> <months> <years>

  Examples for a valid format are:

  - older than five minutes
    deleteOlderThan=5 0 0 0 0
  - older than two months
    deleteOlderThan=0 0 0 2 0

- **processTemplates**
  This property specifies for which templates instances will be deleted. Multiple business process template names can be specified separated by comma. If nothing is specified, then instances of any template will be deleted.

- **processStates**
  This property specifies in which end state an instance must be for it to be deleted by the Process Cleanup Service sample. To specify multiple end states use a comma to

separate them. Possible values are `terminated`, `failed`, and `finished`.

Examples for a valid format are:

- delete for any end state
  processStates=terminated,failed,finished

- delete only terminated business process instances
  processStates=terminated

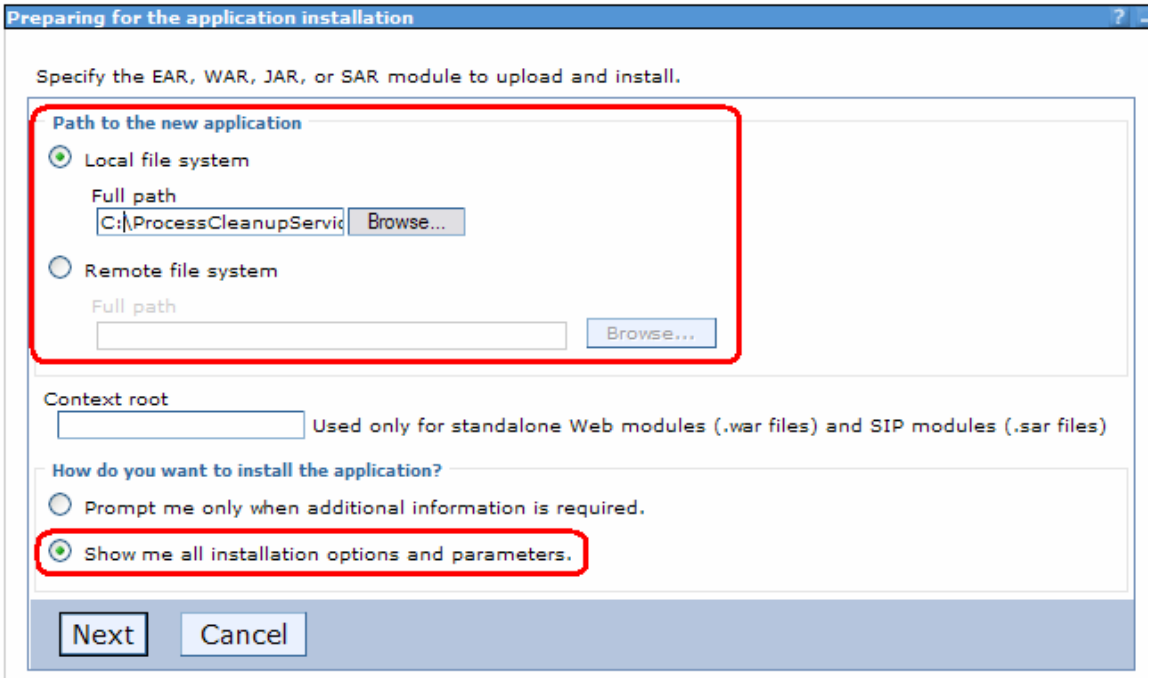
- **maxNumberInstances**
  This property specifies the maximum number of instances that will be deleted in one run in daemon mode. The value '0' means that there is no limit.

4.2. Installation

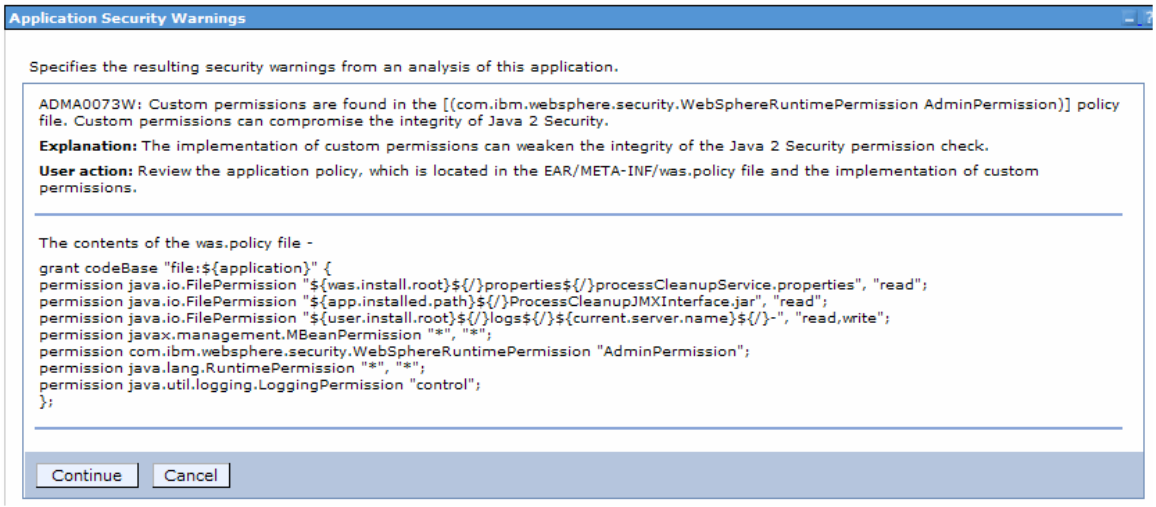
There are multiple ways available to install an application in WebSphere Process Server. In the following, installation using the Administrative Console is shown. Specifics to the Process Cleanup Service sample application configuration are highlighted. General knowledge about installing an application is assumed.

1) In the Enterprise Applications panel of the Administrative Console click Install.

2) Select `the ProcessCleanupServiceV2.ear` and click "Show me all installation options and parameters":



3) Click Next until you reach the Application Security Warnings panel:

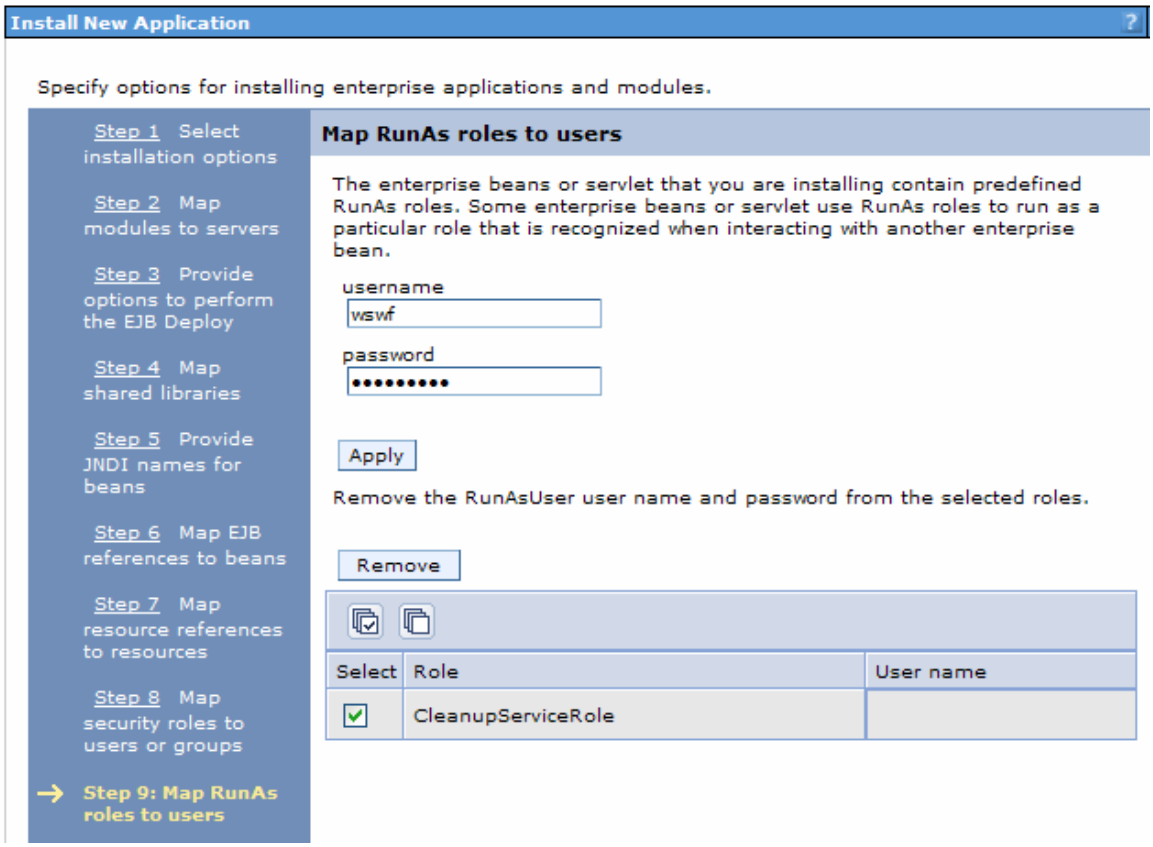


The permissions specified here are required to run the Process Cleanup Service sample when Java2 Security is enabled.

4) Click Continue and perform the standard application installation steps.

5) At Step 2, make sure that the application is mapped to a deployment target where Business Process Choreographer is configured.

6) At Step 8 and 9, add a user who has administrative rights for the Business Flow Manager API (Business Process Administrator):



7) Finish installing the Process Cleanup Service sample application. Start the application and check the log files for any exceptions.

4.3. Additional steps for interactive mode configuration

For interactive execution mode there is a Jython script named `manageProcessCleanupCustomService.py` that is provided with the Process Cleanup Service sample.

Store the script in your normal directory for administrative scripts.

5. Running the Process Cleanup Service sample

The following chapters explain how the Process Cleanup Service sample operates in the two different execution modes.

5.1. Running in daemon mode

You configured the frequency used for the Process Cleanup Service sample in the `processCleanupService.properties` files. When the defined time is reached, the Process Cleanup Service sample will start its processing in daemon mode.

5.1.1. Deleting business process instances

If the Process Cleanup Service sample started in daemon mode, you will see the following output in the `SystemOut.log`,:

```
cleanup       I   Process Cleanup Service started in daemon mode.
```

After processing each slice, statistics are printed out about the current run:

```
cleanup       I     Instances deleted: 0 in 0 seconds.
```

When it is finished, an appropriate message is output:

```
cleanup       I   Process Cleanup Service did not find more instances eligible
                    for deletion and will stop processing.
```

5.2. Running in interactive mode

For interactive mode, use the script `manageProcessCleanupCustomService.py` for processing. You must run this script using the wsadmin tool in connected mode.

Here is an example command to run the script:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server
server1 -delete all
```

If no parameters are specified, a help and options dialog is displayed:

```
Usage: wsadmin -f manageProcessCleanupCustomService.py
       [ ([-node <nodeName>] -server <serverName>) | (-cluster <clusterName>) ]

       ( ( -stop) |

         ( -count  all|finished|terminated|failed
                   [ -templateName <templateName> ]
                   [ -completedBefore <utc timestamp> ] )  |

         ( -delete all|finished|terminated|failed
                   [ -templateName <templateName> ]
                   [ -completedBefore <utc timestamp> ]
                   [-sync))

Manage process cleanup custom service. This script must be run in connected
mode, that is, the application server or at least one cluster member and the
deployment manager must be running.

The '-node', '-server', and '-cluster' parameters denote the Business Process
```

```
Choreographer configuration to work with.

The default values for the optional parameters are:
  node:               <local node> (N1)
  templateName:       any process template
  completedBefore:    current time
```

5.2.1. Counting business process instances

The `count` command is to be used to display how many business process instances are in a certain end state and/or have a particular template name and/or were completed before a given time.

See the usage summary of the `manageProcessCleanupCustomService.py` script for more details.

The following example command counts how many business process instances of the template 'MyProcessTemplate' are in the 'failed' state:

```
wsadmin.bat –f C:\scripts\manageProcessCleanupCustomService.py –server
server1 –count failed –templateName MyProcessTemplate
```

The result is written to the `SystemOut.log` file:

```
cleanup       I   Number of instances found: 0
```

In addition, the result returned to the wsadmin client and displayed there, for example:

```
B:\w\p\_Srv01\bin>wsadmin.bat –f C:\scripts\manageProcessCleanupCustomService.py
–server server1 –count failed –templateName MyProcessTemplate
WASX7209I: Connected to process "server1" on node N1 using SOAP connector;  The
type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[-serve
r, server1, –count, failed, –templateName, MyProcessTemplate]"

Counting process instances in end state failed with templateName
<MyProcessTemplate>
Number of instances found: 0.
```

5.2.2. Deleting business process instances

Use the `delete` command to delete business process instances that are in a certain end state and/or have a particular template name and/or were completed before a given time.

See the usage summary of the `manageProcessCleanupCustomService.py` script for more details.

The following example command deletes all completed business process instances of the template 'MyProcessTemplate' which completed before December 20, 2008:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server
server1 -delete all -templateName MyProcessTemplatete -completedBefore
2008-12-20
```

The number of business process instances deleted is written to the `SystemOut.log` file:

```
cleanup       I   Process Cleanup Service started in interactive mode.
cleanup       I     Instances deleted: 0 in 0 seconds.
```

Note, that the number of deleted business process instances is not returned to the wsadmin client. This is because the deletion operation is normally a long-running process. There could be long wait times for the client which could lead to connection timeouts on the client side. Therefore the deletion operation is normally started asynchronously. Progress information is written to the `SystemOut.log` file of the server executing the deletion operation. In a cluster environment, the server executing the deletion operation can be determined by the output of the wsadmin command:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server server1 -
delete all -templateName MyProcessTemplatete -completedBefore 2008-12-20
WASX7209I: Connected to process "server1" on node N1 using SOAP connector;  The
type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[-server, server1,
-delete, all, -templateName, MyProcessTemplatete, -completedBefore, 2008-12-20]"

Deleting process instances in end state all with templateName
<MyProcessTemplatete>  with completedBefore: < 2008-12-20>
Deletion started sucessfully on
WebSphere:cell=TestNode01Cell,version=6.1.0.21,spec=1.0,name=processCleanupMBean
Id,mbeanIdentifier=processCleanupMBeanId,type=ProcessCleanupCustomService,node=N
1,process=server1
Check the servers log files for progress.
```

You can specify the optional parameter `sync` to start the deletion operation in synchronous mode. In this case, the number of deleted business process instances is also returned to the wsadmin client. Use this option with care and only for small a small number of instances:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server server1 -
delete all -templateName MyProcessTemplatete -completedBefore 2008-12-20 -sync
WASX7209I: Connected to process "server1" on node N1 using SOAP connector;  The
type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[-serve
r, server1, -delete, all, -templateName, MyProcessTemplatete, -completedBefore,
2008-12-20, -sync]"

Deleting process instances in end state all with templateName <
MyProcessTemplatete>  with completedBefore: < 2008-12-20>
Deleting synchronously.
Number of instances deleted: 0.
```

5.2.3. Stop the currently running deletion process

You can use the `stop` command to stop the execution of a currently running deletion operation that was started in interactive mode.

The following example command stops any currently running deletion operation in interactive mode when the currently running transaction slice has finished:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server
server1 -stop
```

The following message is written to the `SystemOut.log` file:

```
cleanup       I   Received stop request for interactive Process Cleanup Service
processing.
```

The following output is displayed in the wsadmin client:

```
wsadmin.bat -f C:\scripts\manageProcessCleanupCustomService.py -server server1 -
stop
WASX7209I: Connected to process "server1" on node N1 using SOAP connector;  The
type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[-serve
r, server1, -stop]"

Stop command issued. Cleanup service processing will be stopped in interactive
mode.
Executed command sucessfully on mbean
WebSphere:cell=TestNode01Cell,version=6.1.0.21,spec=1.0,name=processCleanupMBean
Id,mbeanIdentifier=processCleanupMBeanId,type=ProcessCleanupCustomService,node=N
1,process=server1pMBeanId,type=ProcessCleanupCustomService,node=N1,process=serve
r1
```

5.3. Tracking progress and logging

The enhanced Process Cleanup Service sample application uses Java logging facilities in to write progress information to the `SystemOut.log` file.

It uses a logger called `com.ibm.bpe.sample.cleanup`.

Information, warning and error messages are logged to the `SystemOut.log` file. You can enable detailed tracing using the WebSphere trace settings by specifying `com.ibm.bpe.sample.cleanup=finest`.

Information, warning and error messages can also be logged to a separate log file called `processCleanupSample#.log` in the servers log directory. To enable this separate logging, you must modify the `processCleanupService.properties` file, as specified above.

Both enabling the tracing and enabling the separate logging has an impact on the performance of the system. It should therefore only be enabled with care.

You can customize the logging by adapting the provided sample code. More information on that can be found in the section describing the implementation of the Process Cleanup Service sample.

6. Implementation overview

For the 'daemon' execution mode, the original Process Cleanup Service sample already met the majority of the requirements. For the 'interactive' execution mode, a new interface has been introduced and the available Process Cleanup Service sample has been enhanced. In the following, the two execution modes and all options are explained.

The Process Cleanup Service sample uses the following components that are available in WebSphere Application Server and WebSphere Process Server. For more information about these components, refer to the WebSphere Application Server information center.

- **Startup beans**
  They allow business logic to run when an application starts or stops.

- **Scheduler service**
  The scheduler service enables application tasks to run at specific times or intervals. The Process Cleanup Service sample uses the scheduler to control when it is run. The scheduler notifies the task handler session bean, which triggers the deletion processing.

- **Asynchronous beans**
  An asynchronous bean is a Java object or enterprise bean that can be run asynchronously by a Java 2 Platform Enterprise Edition (J2EE) application, using the J2EE context of the asynchronous bean creator. An asynchronous bean work object is used by the Process Cleanup Service sample to start a long running deletion process.

- **MBeans**
  MBeans are JavaBeans used as access points to applications or servers for the management interface of WebSphere Application Server. You can use the wsadmin tool to call the MBean that is provided by the Process Cleanup Service sample, which provides the interactive mode interface.

6.1. Exploring the Process Cleanup Service sample

Because the Process Cleanup Service sample is a sample application, it provided with as-is source code. This material contains programming source code for your consideration.

The Process Cleanup Service sample has not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of this program.

It is based on the original Process Cleanup Service sample application which can be found at the following site:
http://www-01.ibm.com/support/docview.wss?uid=swg27007816

The description of the original sample includes a description of how that original Process Cleanup Service sample was implemented. This section only describes the changes and enhancements made in this version of the Process Cleanup Service sample. Therefore, to get a full picture it is recommended that you read the original documentation.

The following figure shows the components of the Process Cleanup Service sample. The components highlighted in orange are either new or have changed extensively compared to the original Process Cleanup Service sample.
Only the orange highlighted components are discussed here. For information about the the other components, refer to the documentation for the original Process Cleanup Service sample.

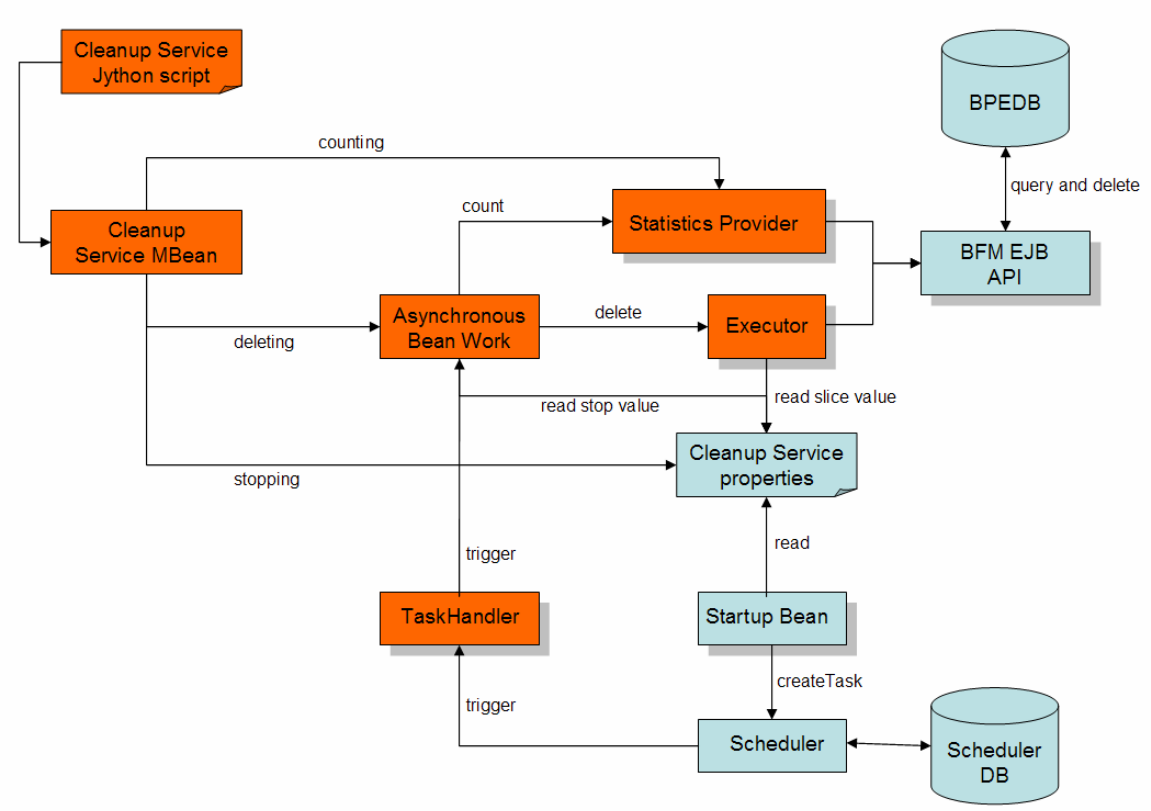


Figure 1 Process Cleanup Service sample and its components. The components highlighted in orange are new or have changed significantly compared to the original Process Cleanup Service sample.

6.1.1. The ProcessCleanupMBean managed bean

The Process Cleanup Service sample uses an MBean to accept calls from the WebSphere administration interface. The implementation of this MBean is available in class `ProcessCleanupMBean`.

The MBean is activated and deactivated the ProcessCleanupService application is started and stopped. The source code for the activation/ deactivation can be found in the `ProcessCleanupServiceStartupBean` implementation.

Tto change the privileges for accessing this MBean, You can introduce a security descriptor as described in the WebSphere information center:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjmx_admin_finegr_mbsec.html

For general information about MBeans see the following information center documentation:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjmx_extend.html

6.1.2. The manageProcessCleanupCustomService.py script

This Jython script is used in interactive execution mode to call the MBean methods provided by the `ProcessCleanupMBean`.

This script must be executed using the wsadmin tool in connected mode. In a network deployment (ND) environment, the Deployment Manager and at least one server hosting the Business Process Choreographer configuration must be running.

6.1.3. The ProcessCleanupStatisticsProviderBean session bean

This session bean is used to query the Business Flow Manager EJB API for completed business process instances that meet certain criteria.

6.1.4. The ProcessCleanupExecutorBean session bean

This session bean uses the Business Flow Manager API to delete a given number of completed business processes instances that meet certain criteria. The number of instances to delete is determined by the “slice” value, which is specified in the `ProcessCleanupService.properties` file.

6.1.5. The ProcessCleanupWork asynchronous bean

This asynchronous bean work object has high level control of the deletion operation and is responsible for calculating the statistics after each deletion step.

It is started asynchronously in a separate thread. The advantage of using this over the original Process Cleanup Service sample implementation is that neither the Task Handler bean, nor the MBean must wait for the outcome of the cleanup operation. Those wait times led to timeouts in the original sample, which do not occur anymore with this implementation.

By calling the Executor bean multiple times until the cleanup operation has finished, the asynchronous bean work also eliminates the need for user transactions. In this sample, all deletion operations are performed under the control of container managed global transactions.

For more information about asynchronous beans refer to the following information center documentation:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/asyncbns/concepts/casb_asbover.html

6.1.6. The ProcessCleanupTaskHandlerBean session bean

In the original Process Cleanup Service sample, the Task Handler bean was responsible for executing the deletion operation. In this version, it is used by the scheduler to notify the Process Cleanup Service sample when a run in daemon mode is required.

It calls the `ProcessCleanupWork` implementation asynchronously to trigger the deletion operation in daemon mode.

6.1.7. The ProcessCleanupLogger class

This `ProcessCleanupLogger` is a utility class used for logging information, warning, and error messages as well as tracing the Process Cleanup Service sample, if required.

It uses the `java.logging` facilities and allows logging messages to be written to the `SystemOut.log` file and also into custom files.

6.1.8. The BPEScheduler usage

Like the original Process Cleanup Service sample, this version uses the scheduler that is configured for Business Process Choreographer (`BPEScheduler`) to schedule the daemon mode cleanup times. In addition, it uses the work manager that is configured for Business Process Choreographer (`wm/BPEScheduler`) for starting the asynchronous bean work.

To minimize the impact on the Business Process Choreographer runtime, it is recommended that you configure a custom scheduler and custom work manager and adapt the sample code accordingly to use them.

For more information on configuring schedulers and work managers, refer to the following information center pages:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/scheduler/tasks/tsch_configurescheduler.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/asyncbns/tasks/tasb_workmanager.html

6.2. Daemon execution mode

This is the default operating mode to clean up instances after the audit and tracing period. In daemon mode, the deletion operation runs in the background at a predefined time and according to a predefined configuration. It deletes completed business process instances according to their end state, template name and/or a date before which the instances have been completed.

The following figure shows an overview of the architecture for the Process Cleanup Service sample and its components that are used in daemon mode.

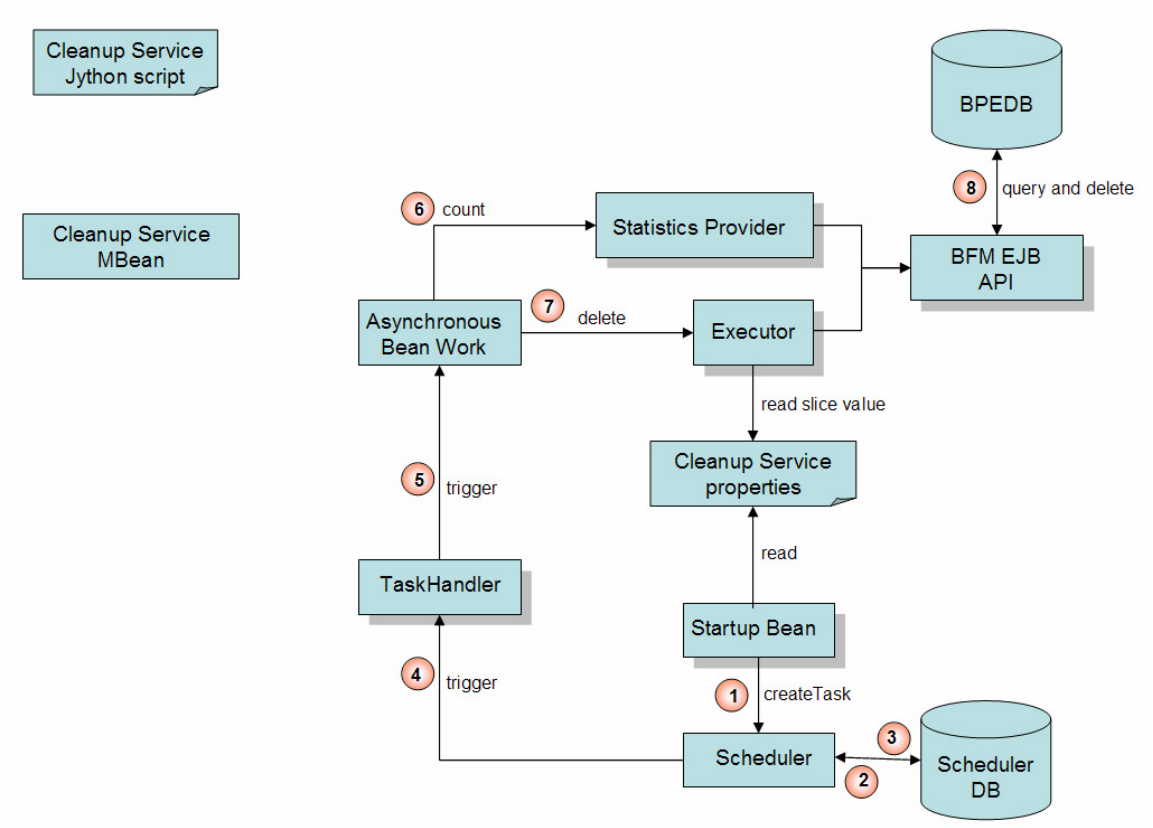


Figure 2 Process Cleanup Service sample components - Deletion in daemon mode

The deletion operation during daemon mode works like this:

1) When starting the Process Cleanup Service sample application, the Startup bean reads the configuration data for the daemon mode processing from the `processCleanupService.properties` file and registers an appropriate scheduled task with the Scheduler service.

2) The Scheduler service stores the task in its database.

4) When a task is due, the Scheduler service triggers the Task Handler bean.

5) The Task Handler Bean calls the asynchronous bean work object.

7) The Work object uses the Executor session bean to delete one slice of business process instances. It repeats this step until the deletion operation has completed.

8) The Executor session bean uses the Business Flow Manager EJB API to query and delete the specified business process instances.

6.3. Interactive execution mode

The interactive execution mode has been introduced so that administrators can perform unplanned business process instance maintenance. This mode should be used to solve the effects of problems such as business processes failing due to external back-end unavailability or performance degradation.

The Process Cleanup Service sample provides an MBean interface, which provides the functionality to count and delete business process instances. It also provides the option to stop a currently running deletion operation that was started in interactive mode.

Using the wsadmin tool, an administrator can count and delete completed business process instances by specific criteria. Also for simplified usage, a Jython script is provided with this Process Cleanup Service sample, which provides commands to call the MBean operations.

The following figure shows an overview of the architecture for the Process Cleanup Service sample and its components that are used in interactive mode.

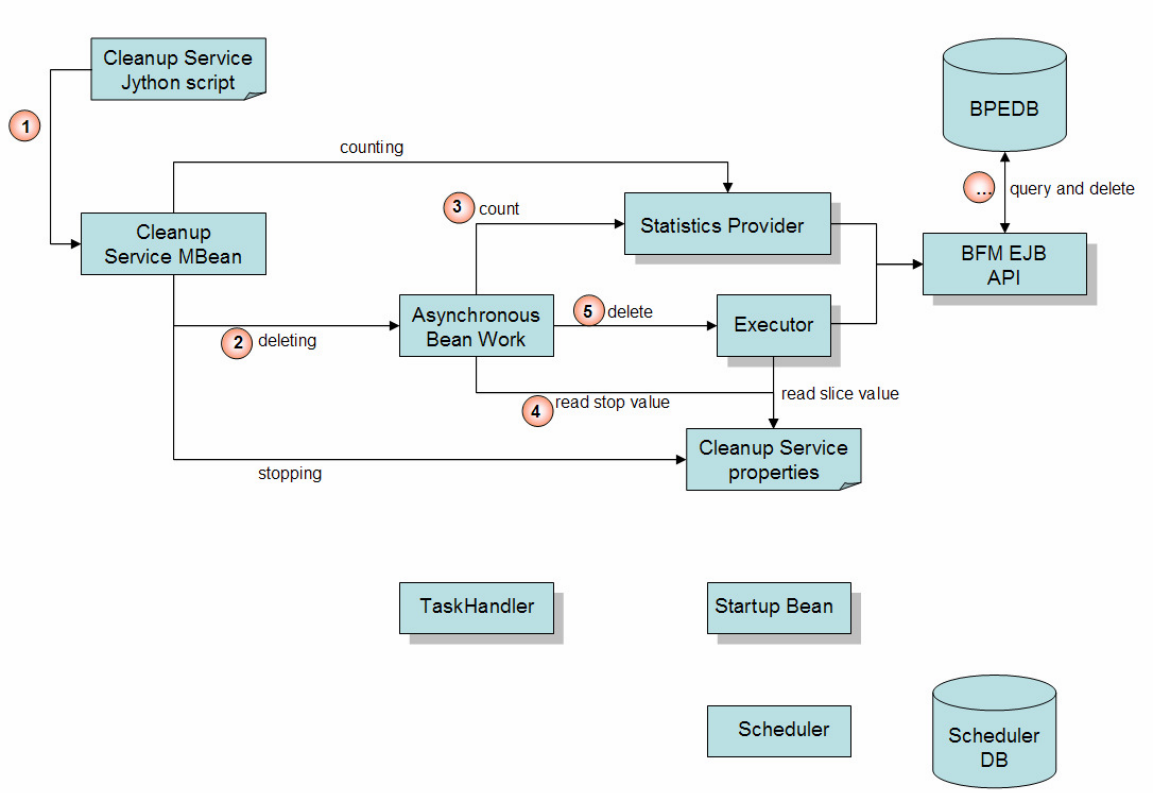


Figure 3 Process Cleanup Service sample components - Deletion in interactive mode

The deletion operation during interactive mode works like this:

1) A WebSphere Process Server administrator uses the `manageProcessCleanupCustomService.py` script to invoke the deletion operation for business process instances of a certain template, end state and/or completion time. Wsadmin connects to the WebSphere management framework in connected mode and the corresponding deletion operation on the Process Cleanup Service sample MBean is invoked.

2) The MBean starts the asynchronous bean work object.

4) The work object retrieves the value for "stop" from the configuration. In interactive mode, an administrator can issue a "stop" command to cancel the interactive mode deletion operation. If "stop" is issued, then no further deletion will take place after the current deletion transaction (slice) has finished.

5) The work object uses the Executor session bean to delete one slice of business process instances. It repeats steps 4 and 5 until the deletion operation has completed.

6) The Executor session bean uses the Business Flow Manager EJB API to query and delete the specified business process instances.

7. Best practices

When using this sample, the following best practices should be applied:.

1) In interactive mode, only execute one deletion operation at a time.
2) In interactive mode, only use the `-sync` parameter on the delete operation if a small number of instances are to be deleted.
3) A “slice” value of between 10 and 30 has been found to be a good value.

8. Recommended reading

The original Process Cleanup Service sample for Business Process Choreographer:
http://www-01.ibm.com/support/docview.wss?uid=swg27007816

Cleanup procedures for Business Process Choreographer
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.bpc.620.doc/doc/bpc/c2_cleanup.html