App Agent Node Properties Reference

This reference page contains information about app agent node properties. The properties are listed in alphabetical order.

- adaptive-callgraph-granularity
- ado-new-resolvers
- api-thread-activity-timeout-in-seconds
- api-transaction-timeout-in-seconds
- aspdotnet-mvc-naming-controlleraction
- async-tracking
- async-transaction-demarcator
- bci-log-config
- boot-amx
- callgraph-granularity-in-ms
- capture-404-urls
- capture-error-urls
- capture-raw-sql
- capture-set-status
- capture-spring-bean-names
- collect-user-data-sync
- collection-capture-period-in-minutes
- disable-custom-exit-points-for
- dev-mode-suspend-cpm
- disable-exit-call-correlation-for
- disable-exit-call-metrics-for
- disable-percentile-metrics
- dont-show-packages
- downstream-tx-detection-enabled
- enable-default-http-error-code-reporter
- enable-info-point-data-in-snapshots
- enable-instance-monitoring
- enable-interceptors-for-security
- enable-json-bci-rules
- enable-object-size-monitoring
- enable-soap-header-correlation
- enable-startup-snapshot-policy
- enable-transaction-correlation
- enable-xml-bci-rules
- end-to-end-message-latency-threshold-millis
- find-entry-points
- force-hotspot-if-diag-session
- heap-storage-monitor-devmode-disable-trigger-pct
- hotspot-collect-cpu
- jdbc-callable-statements

- jdbc-connections
- jdbc-dbcam-integration-enabled
- jdbc-prepared-statements
- jdbc-statements
- jmx-appserver-mbean-finder-delay-in-seconds
- jmx-rediscover-mbean-servers
- log-request-payload
- max-business-transactions
- max-call-elements-per-snapshot
- max-concurrent-snapshots
- max-jdbc-calls-per-callgraph
- max-jdbc-calls-per-snapshot
- max-service-end-points-per-node
- max-service-end-points-per-thread
- max-urls-per-error-code
- min-duration-for-jdbc-call-in-ms
- min-load-per-minute-diagnostic-session-trigger
- minimum-age-for-evaluation-in-minutes
- minimum-number-of-elements-in-collection-to-deep-size
- minimum-size-for-evaluation-in-mb
- on-demand-snapshots
- percentile-method-option
- percentiles-to-report
- rest-num-segments
- rest-transaction-naming
- rest-uri-segment-scheme
- rmqsegments
- show-packages
- slow-request-deviation
- slow-request-monitor-interval
- slow-request-threshold
- spring-integration-receive-marker-classes
- thread-correlation-classes
- thread-correlation-classes-exclude
- thread-cpu-capture-overhead-threshold-in-ms

adaptive-callgraph-granularity

Description: This property enables adaptive snapshots. The call graph granularity for adaptive snapshots is based on the average response time for the business transaction during the last one minute and is thus adaptive. The following distribution is used:

- Granularity of 10 ms for average response time of <= 10 seconds
- 50 ms for 10 to 60 seconds
- 100 ms for 60 to 600 seconds
- 200 ms for > 600 seconds

Type: Boolean

Default value: false

Platform: Java, .NET

ado-new-resolvers

Description: Enable database detection and naming for ODP.NET backends labeled "Unknown0". See Resolve Unknown0 Database Backend Name.

Type: Boolean

Default value: false

Platform: .NET

api-thread-activity-timeout-in-seconds

Description: This property provides a time-out value that comes into play when you have added global transactions to your application using APIs from the AppDynamics SDK. In the event that the added transaction spawns additional threads that do not return or complete, this property provides a safety valve time-out value. The value is in seconds. The removeCurrentThread method is invoked after the specified time out period.

Type: Integer

Default value: 300 seconds

Range: Minimum =1; Maximum=3600

Platform: Java

api-transaction-timeout-in-seconds

Description: This property provides a time-out value that comes into play when you have added global transactions to your application using APIs from the AppDynamics SDK. In the event that the added transaction does not return or complete, this property provides a safety valve time-out value. The time-out value is in seconds. The endTransaction method is invoked after the specified time-out period.

Type: Integer

Default value: 300 seconds

Range: Minimum=1; Maximum=3600

Platform: Java

aspdotnet-mvc-naming-controlleraction

Description If set to true, this property configures the agent to identify ASP.NET MVC business transactions as Controller/Action. See Identify MVC Transactions by Controller and Action.

Type: Boolean

Default value: false

Platform: .NET

async-tracking

Description: Enable detection of asynchronous exit points. *New in 3.8.5*, also enable thread correlation for ThreadPoolQueueUserWorkItem. See Monitor Async Transactions for .NET.

Type: Boolean

Default value: false

Platform: .NET

async-transaction-demarcator

Description: This class name and method name combination marks the end of an asynchronous distributed transaction.
Use the format ClassName/MethodName. For example, foo/bar where foo is the class name and bar is the method name.

Type: String

Default value: none

Platform: Java

bci-log-config

Description: Use this property to configure the bytecode transformer log (BCT log). This log shows what AppD instruments and what classes are loaded in the JVM. The initial file (the *0.log*) is saved to preserve the context of the server start up and is not rolled over. The subsequent files rotate. The format of the file name is ByteCodeTransformer.<*timestamp*>.<*N*>.log. The time stamp is represented as YYYY_MM_DD_HH_mm_ss. N increments starting from zero.

For example:

```
ByteCodeTransformer.2012_09_12_20_17_57.0.log
```

Because the file size is checked every 15 seconds, the files may be a bit larger than the specified threshold value before they are rolled over.

The first log file generated is named as follows:ByteCodeTransformer.<timestamp>.0.log
The value for this property is of the format: "20,5,4".

Type: String

Default value: 20,5,4

Example:

For value = 20,5,4
20 = size, in MB, of the first log file, the *.0* version
5 = size in MB for each subsequent rolling file
4 = the number of ByteCodeTransformer log files to keep

Platform: Java

boot-amx

Description: If set to true, enables support for Glassfish AMX MBeans. See GlassFish Startup Settings for more information about Glassfish server support.

Type: Boolean

Default value: true

Platform: Java

callgraph-granularity-in-ms

Description: Specifies the granularity for call graphs for this node. The global configuration is ignored if this property is used. This value is ignored if the adaptive-callgraph-granularity property is set to true. A value of zero (default) means the global configuration (from **Configure --> Instrumentation --> Call Graph Settings)** is used. Does not need a restart.

Type: Integer

Default value: 0 (zero)

Range: Minimum=0; Maximum=5000

Platform: Java, .NET

capture-404-urls

Description: This property disables or enables the capture of the URLs causing 404 errors. The URLs are reported as ERROR events every 15 minutes and are viewable through the Event Viewer. The JVM needs a restart if retransformation is not supported for the JVM version.

404 errors usually mean that no application code is executed, so there is nothing to snapshot. You can get insight into the 404 error by setting this property to true. It reports all the URLs which caused 404 error.

The capture-404-urls node property is **deprecated** in AppDynamics v. 3.6 and replaced with capture-error-urls.

Type: Boolean

Default value: false

Platform: Java

capture-error-urls

Description: This property enables or disables the capture of the following HTTP errors:

- 401 - Unauthorized
- 500 - Internal Server Error
- 404 - Page Not Found
- All other error codes are put in a generic HTTP error code bucket.

For these 4 categories, the agent collects URLs, limited to 25 per category per minute, and sends an event out every 5 minutes.

You can see these URLs when you drill down on an error code by clicking Troubleshoot -> Errors -> Exceptions tab -> HTTP Error Codes.

Type: Boolean

Default value: true

Platform: Java

capture-raw-sql

Description: Capture raw SQL calls for this node. For example:

```
select * from user where ssn = '123-123-1234'
```

If you change this node property in an environment that is using the IBM JVM, you need to restart the JVM. This is because the feature requires retransformation of certain JDBC classes, which is not possible with our IBM agent.

Type: Boolean

Default value: false

Platform: Java, .NET

capture-set-status

Description: Directs the agent to capture errors where the webservice is using setStatus() to send back an error. By default only sendError() is instrumented by the agent.

Type: Boolean

Default value: false

Platform: Java

capture-spring-bean-names

Description: *New for 3.8.1* When a class is mapped to multiple Spring bean names, by default only the name of the first Spring bean found displays. This can be misleading. For example, when you see a call graph for web service A that has Spring beans from web service B. Setting this property to false shows only the class name when these conflicts occur and does not show the Spring bean name.

Type: Boolean

Default value: true

Platform: Java

collect-user-data-sync

Description: Collect user data from diagnostic POJO data collectors synchronously. Does not require a restart.

Type: Boolean

Default value: true

Platform: Java

collection-capture-period-in-minutes

Description: Total interval in minutes since server restart for which collections are captured for leak evaluation. The property takes effect only after the node restart.

Type: Integer

Default value: 30

Range: Minimum=5; Maximum=N/A

Platform: Java

disable-custom-exit-points-for

Description: There is a limited set of "custom" exit points configured by default. You can disable these preconfigured custom exit points by specifying the type here using this property. Allowed values are: SAP, Mail, LDAP, MongoDB. For multiple values, use a comma separated list.

Type: String

Default value: none

Platform: Java

dev-mode-suspend-cpm

Description: *New in 3.8* The maximum number of transactions monitored per minute during Development mode before the the system switches out of development mode and into normal operation mode.

Type: Integer

Default value: 500

Range: Minimum=0; Maximum=N/A

Platform: Java

disable-exit-call-correlation-for

Description: Disable exit call correlation for a specific type of call. For example, HTTP, JMS, RMI. By default all exit call correlations are enabled.

Type: String

Default value: none

Platform: Java, .NET

disable-exit-call-metrics-for

Description: Disables exit call monitoring for a specific type of exit call; for example, HTTP, JMS, WEB_SERVICE. If this property is set, the average data (calls/min, avg response time) for the specific exit call type is not collected. However, for a snapshot all details are collected. Set this property if the application makes a large number of exit calls per transaction and the avg metrics are not important.

For lists of backends that are automatically discovered see Backend Monitoring.

Type: String

Default value: By default all exit call metrics are enabled.

Platform: Java, .NET

disable-percentile-metrics

Description: Percentile metrics for business transaction response time can be displayed in the Metric Browser or Custom Dashboard. Percentiles are generally a better indicator than averages because they are not sensitive to outliers while averages can get distorted. Enable this property if you want to view the percentile metrics. This property is used in conjunction with the percentiles-to-report node property where you can change the percentile point to report. You can change this property in the app-agent-config.xml file or from within the UI.

Percentile metrics can be interpreted as follows: the 95th percentile point indicates that 95% of all response times were less than the metric value. It provides a sense of how the response times are distributed. For example if the metric Observed (Average) or percentile value is 300ms, it implies that 95% of the business transaction response times are less than 300ms, and therefore implies that 95% of the business transaction response times are within acceptable ranges. It also implies that 5% of the requests have been taking more time and could be a cause for concern.

Under certain circumstances, you may want to change how the percentile is calculated. For more information, see percentile-method-option node property.

Changes to this property do not require that you restart the agent.

Type: Boolean, true or false

Default value: By default this property is set to 'true' and all percentile metrics are disabled. When set to 'false', the 95th percentile metrics are captured and reported.

Platform: Java

dont-show-packages

Description: Do not show these packages / class names in addition to the ones configured in the global call graph configuration, for the call graphs captured on this node. Does not need a restart.

Type: String

Default value: none

Platform: Java, .NET

downstream-tx-detection-enabled

Description: If the agent cannot reach the controller for a prolonged period, it turns off most services and notifies the continuing tiers that upstream transaction was detected and is not being monitored. Set this property to true to enable the continuing tiers to detect their own transactions in the event of network failure on the upstream tiers.

Type: Boolean

Default value: false

Platform: Java, .NET

enable-default-http-error-code-reporter

Description: This property disables or enables automatic HTTP error code reporting for error codes between 400 - 505.

Type: Boolean

Default value: true

Platform: Java, .NET

enable-info-point-data-in-snapshots

Description: This property disables or enables the capture of information point calls in snapshots. When this property is set to true, information point calls appear in the User Data section of the snapshot.

Type: Boolean

Default value: false

Platform: Java, .NET

enable-instance-monitoring

Description: This property enables or disables Instance tracking on this node. Does not need a JVM restart.

Type: Boolean

Default value: false

Platform: Java

enable-interceptors-for-security

Description: This property enables or disables security interceptors on this node. Set this property to true in environments where the Java 2 Security Manager is enabled. If the Java 2 Security Manager is enabled, and this property is not set to true, then the agent will encounter SecurityExceptions, and will not be able to collect the data that it should. Does not need a JVM restart.

Type: Boolean

Default value: false

Platform: Java

enable-json-bci-rules

Description: Set this property to true to enable JSON bytecode instrumentation rules. AppDynamics instruments the *get* and *getString* methods within the package/class *org.json.JSONObject* when you set this value to true. Needs a JVM restart.

Type: Boolean

Default value: *New in 3.8* True. This only affects new applications; applications created with a 3.7.x controller will still have this property set by default to false.

Platform: Java

enable-object-size-monitoring

Description: This property is related to Automatic Leak Detection (ALD) and enables or disables Object Size monitoring on this node. Changing this property does not need a JVM restart. ALD is supported for JVM version 1.6 and up.

Type: Boolean

Default value: false

Platform: Java

enable-soap-header-correlation

Description: *New in 3.8* This property controls correlation with web service transactions. When enabled, a node which receives a web service transaction may correlate that transaction with any downstream transactions. The ability to correlate depends on the particular web services framework. When disabled, the agent will not perform correlation through any web service tiers.

Type: Boolean

Default value: false

Platform: Java, .NET

enable-spring-integration-entry-points

Description: This property disables or enables the default detection of Spring Integration entry points. Set to 'false' to disable.

Default detection of Spring Integration entry points is based on MessageHandler. In cases where a lot of application flow happens before the first MessageHandler is executed,

- set this property to 'false',
- configure suitable POJO entry points,
- and specify the property spring-integration-receive-marker-classes.

See also Spring Integration Support.

Type: Boolean

Default value: true

Platform: Java

enable-startup-snapshot-policy

Description: This property disables or enables the policy for start-up transaction snapshot. This means snapshots are collected for all BTs for all invocations for the first 15 minutes of an application server start.

Type: Boolean

Default value: false

Platform: Java, .NET

enable-transaction-correlation

Description: This property disables or enables transaction correlation. It does not require a restart.

Type: Boolean

Default value: true

Platform: Java, .NET

enable-xml-bci-rules

Description: This property enables Java XML Binding and DOM Parser bytecode instrumentation rules. Set to true to enable. The change takes effect after a JVM restart.

Type: Boolean

Default value: *New in 3.8* true. This only affects new applications; applications created with a 3.7.x controller will still have this property set by default to false.

Platform: Java

end-to-end-message-latency-threshold-millis

Description: Enables end-to-end message latency monitoring for distributed asynchronous systems by setting up a threshold. Any message taking more time than the threshold is viewable through the Event Viewer.

Type: Integer

Default value: 0

Range: Minimum=0; Maximum=36000

Platform: Java

find-entry-points

Description: Set this property to true to log all potential entry points that are hitting instrumented exit points or loggers to the Business Transactions log file.

Use this property when you suspect that some traffic is not being detected as business transactions.

Note that this property should be enabled for debugging only. It is strongly recommended to disable this property in production setup by changing the value from true to false.

Type: Boolean

Default value: false

Platform: Java, .NET

force-hotspot-if-diag-session

Description: Set this property to true to collect hotspot snapshots for all transactions for which there is a manual diagnostic session defined. See also hotspot-collect-cpu.

Type: Boolean

Default value: false

Platform: Java

heap-storage-monitor-devmode-disable-trigger-pct

Description: The maximum Java heap utilization percentage for development mode. If the heap utilization exceeds this value, Development mode is automatically disabled.
Type: Integer
Default value: 90
Range: Minimum=0, Maximum=100
Platform: Java

hotspot-collect-cpu

Description: Set this property to true to collect CPU time, instead of real time, within hotspot snapshots, if the snapshots are collected as a result of the force-hotspot-if-diag-session=true property.

Type: Boolean

Default value: false

Platform: Java

jdbc-callable-statements

Description: Use this property to indicate the implementation classes of the *java.sql. CallableStatement* interface that should be instrumented. For example, to instrument calls to Times Ten (an unsupported database), you could set this property to the following:

```
com.timesten.jdbc.JdbcOdbcCallableStatement
```

Type: String

Default value: none

Platform: Java

jdbc-connections

Description: Use this property to indicate the implementation classes of the *java.sql.Connection* interface that should be instrumented. For example, to instrument calls to Times Ten (an unsupported database), you could set this property to the following:

```
com.timesten.jdbc.JdbcOdbcConnection
```

Type: String

Default value: none

Platform: Java

jdbc-dbcam-integration-enabled

Description: *New for AppDynamics for Databases 2.8* Use this property to integrate the App Agent for Java with AppDynamics for Databases (AppD4DB). When this property is enabled, you can link to AppDynamics for Databases from a Transaction Snapshot Flow Map where an exit call is to an Oracle database, and analyze the exact SQL statement that was running at the time of the snapshot. This property works in conjunction with the AppD4DB license and Oracle collector that has been previously set up. Integration must also be setup from the Admin pages of the Controller UI. For more information, see Integrate and Use AppDynamics for Databases with AppDynamics Pro.

Changes to this property do not require a JVM restart for JDK 1.6 and higher. Older 1.5 JVMs do not support class reloading, so for those environments a restart is required.

Type: String

Default value: false

Platform: Java

jdbc-prepared-statements

Description: Use this property to indicate the implementation classes of the *java.sql.PreparedStatement* interface that should be instrumented. For example, to instrument calls to Times Ten (an unsupported database), you could set this property to the following:

```
com.timesten.jdbc.JdbcOdbcPreparedStatement
```

Type: String

Default value: none

Platform: Java

jdbc-statements

Description: Use this property to indicate the implementation classes of the java.sql.Statement interface that should be instrumented. For example, to instrument calls to Times Ten (an unsupported database), you could set this property to the following:

```
com.timesten.jdbc.JdbcOdbcStatement
```

Type: String

Default value: none

Platform: Java

jmx-appserver-mbean-finder-delay-in-seconds

Description: When an app server starts up, the associated MBean server starts and the MBeans are discovered. The timing of these activities varies by app server and by configuration. If this activity is not completed in the time that the AppD agent is expecting to discover the MBeans, then the MBean Browser will not show them. Using this node property, you can delay the discovery of MBeans to make sure that agent discovers all the domains after complete start up of the app server. For example, you can set the delay to a time which is 1.5 times of the server startup time. The default delay for AppD agent is two minutes.

Type: Numeric, seconds

Default value: 200

Platform: Java

jmx-rediscover-mbean-servers

Description: When an app server starts up, the associated MBean server starts and the MBeans are discovered. The timing of these activities varies by app server and by configuration. If this activity is not completed in the time that the AppD agent is expecting to discover the MBeans, then the MBean Browser will not show them. Using this node property, you can trigger the rediscovery of MBeans to make sure that the agent discovers all the domains after complete start up of the app server. Set this property to 'true' and restart the JVM/server.

Type: Boolean

Default value: false

Platform: Java

leak-diagnostic-interval-in-minutes

Description: Interval at which diagnostic data, content summary and activity trace, is captured for leaking collections.

Type: Integer

Default value: 30

Range: Minimum=2; Maximum=N/A

Platform: Java

log-request-payload

Description: Set this property to true to log the request payload (HTTP parameters/cookies /session keys etc) as part of a transaction snapshot.

Type: Boolean

Default value: false

Platform: Java, .NET

max-business-transactions

Description: Sets a limit on the number of business transactions discovered once an agent is started. The limit helps to ensure that the Controller I/O processing capability and agent memory requirements are appropriate for a production environment.

⚠ Warning: Contact AppDynamics Support before changing this setting.

See All Other Traffic Business Transaction.

Type: Integer

Default value: 50

Range: Minimum=N/A; Maximum=200

Platform: Java, .NET

max-call-elements-per-snapshot

Description: This property represents the maximum number of elements that are collected for any call graph for a snapshot. When the limit is reached, the agent stops collecting more data for this call graph, reports what has been collected to that point, and marks the call graph with a warning that the limit was reached. it is not recommended to dramatically increase this number as it may have overhead implications.

Type: Integer

Default value: 5000

Platform: Java, .NET

max-concurrent-snapshots

Description: The maximum number of total snapshots that are allowed, including continuing transactions. When the queue goes over the value set, additional snapshots are dropped. This property is ignored while in Development mode.

Type: Integer

Default value: 20 (v.3.5.x)

Range: Values must be positive integers. No other constraints.

Platform: Java, .NET

max-jdbc-calls-per-callgraph

Description: Maximum number of JDBC/ADO.NET exit-call stack samples per call graph. Only queries taking more time than the value of min-duration-for-jdbc-call-in-ms are reported. Changing the value does not require a restart.

Type: Integer

Default value: 100

Range: Minimum=1; Maximum=1000

Platform: Java, .NET

max-jdbc-calls-per-snapshot

Description: Maximum number of JDBC/ADO.NET exit calls allowed in a snapshot. Calls after the limit are not recorded. Changing the value does not require a restart.

Type: Integer

Default value: 500

Range: Minimum=1; Maximum=5000

Platform: Java, .NET

max-service-end-points-per-node

Description: *New in 3.8* Maximum total number of service endpoints that can be defined on a single node. Increasing the value of this property enables more service end points to be defined on a particular node. This number ensures a maximum limit on overhead due to service endpoints on an agent such that only the service endpoints that are within the limit are evaluated.

Type: Integer

Default value: 100

Range: Minimum=0; Maximum=N/A

Platform: Java

max-service-end-points-per-thread

Description: *New in 3.8* Maximum number of service endpoints that can defined on a single thread of transaction execution. If this property is set to the default value of one and there are two service endpoints defined that impact one specific transaction, at any point in time only the one service endpoint will be evaluated. If a second service endpoint is detected in context of the first one, the second is ignored. But, if the second service endpoint starts after the first one ends, the second service endpoint will be evaluated. Increase this property to monitor additional service endpoints on a thread. This number ensures a maximum limit on overhead and number of metrics due to service endpoints on each thread execution.

Type: Integer

Default value: 1

Range: Minimum=0; Maximum=N/A

Platform: Java

max-urls-per-error-code

Description: Increases the number of URLs the agent can track that produced a certain error. Once the maximum has been reached, all remaining errors are classified as "unknown."

Type: Numeric

Default value: 50

Platform: Java

min-duration-for-jdbc-call-in-ms

Description: A JDBC/ADO.NET call taking more time than the specified time (in milliseconds)is captured in the call graph. The query continues to show up in a transaction snapshot. Setting this value too low (< 10ms) may affect application response times. Changing the value does not require a restart

Type: Integer

Default value: 10

Range: Minimum=0; Maximum=N/A

Platform: Java, .NET

min-load-per-minute-diagnostic-session-trigger

Description: Indicates the number of requests per Business Transaction to evaluate before triggering a diagnostic session. This is useful to prevent diagnostic sessions when there is not enough load.

Type: Integer

Default value: 10

Range: Minimum=N/A; Maximum=N/A

Platform: Java, .NET

minimum-age-for-evaluation-in-minutes

Description: Automatic Leak Detection (ALD) tracks all frequently used Collections. For a Collection object to be identified and monitored it must meet the conditions defined by the ALD properties. This property is the first criteria that needs to be met. The value is the minimum age of the Collection in minutes. The property takes effect after node restart.

From the point the collection is captured, it is monitored if it is still available for the specified period without getting
garbage collected. If it survives then it is evaluated for size checks and if it meets the criteria then it is monitored for long term growth in size.

⚠ Warning: If you reduce the default there may be a performance hit on the CPU and memory because AD needs to process more collections.

Type: Integer

Default value: 30

Range: Minimum=5; Maximum=N/A

Platform: Java

minimum-number-of-elements-in-collection-to-deep-size

Description: Automatic Leak Detection (ALD) tracks all frequently used Collections. For a Collection object to be identified and monitored for it must meet the conditions defined by the ALD properties. This property sets the number of elements threshold.

⚠ Warning: If you reduce the default there may be a performance hit on the CPU and memory because AD is processing more collections.

Type: Integer

Default value: 1000

Platform: Java

minimum-size-for-evaluation-in-mb

Description: Automatic Leak Detection (ALD) tracks all frequently used Collections. For a Collection object to be identified and monitored it must meet the conditions defined by the ALD properties. This property sets the minimum initial size in megabytes for a collection to qualify for monitoring. The collection must also survive for the period specified in the minimum-age-for-evaluation-in-minutes property.

⚠ Warning: If you reduce the default there may be a performance hit on the CPU and memory because AD is processing more collections.

Type: Integer

Default value: 5

Range: Minimum=1; Maximum,=N/A

Platform: Java

on-demand-snapshots

Description: Collect snapshots for all business transactions executed in this node. Does not need a restart. This property is ignored while in Development mode.

Type: Boolean

Default value: false

Platform: Java, .NET

percentile-method-option

Description: You can choose one of two different algorithms to calculate percentiles in AppDynamics:

P Square algorithm (default): This option consumes the least amount of storage and incurs the least amount of CPU overhead. The accuracy of the percentile calculated varies depending on the nature of the distribution of the response times. You should use this option unless you doubt the accuracy of the percentiles presented.

Quantile Digest algorithm: This option consumes slightly more storage and CPU overhead but may offer better percentiles depending on how the response times are distributed.

This property is used in conjunction with the disable-percentile-metrics and the percentiles-to-report node properties.

Changes to this property do not require that you restart the agent.

Type: Numeric

Values: 1 or 2

Default value: By default this property is set to 1, which uses the P Square algorithm to calculate the percentile.

Platform: Java

percentiles-to-report

Description: Percentile metrics for business transaction response time can be displayed in the Metric Browser or Custom Dashboard. By default, the system will capture the 95th percentile metrics. You can change the percentile captured here. If you change this value from its default, you will need to register the property as described in To register a property. This property is used in conjunction with the disable-percentile-metrics node property which must be set to 'false' in order to capture the percentile metrics. Under certain circumstances, you may want to change how the percentile is calculated. For more information, see the percentile-method-optionnode property.

Changes to this property do not require that you restart the agent.

Type: Numeric

Range: Minimum=1; Maximum=99

Default value: By default this property is set to 95.

Platform: Java

rest-num-segments

Description: The property, rest-num-segments specifies the *n* in the first-*n*-segments parameter in rest-uri-segment-scheme. If this property is 0 or less, then the value of this property is ignored. The value of this property is also ignored if rest-uri-segment-scheme=full.

Type: String

Default value: 2

Platform: Java

rest-transaction-naming

Description: This node property accepts any or all of the following parameters in string form. Anything that does not match a parameter is treated as text that is appended to the business transaction name. Properties can be separated with any character.

The agent takes each parameter and fills in the proper value based on the annotations and properties of the Java class:

- {class-name}: The app agent will fill in the name of the Java class mapped to the REST resource
- {method-name}: The method being called
- {class-annotation}: Path annotation applied to the class
- {method-annotation}: Method annotation applied to the method (does not always exist)
- {rest-uri}: URI of the REST resource. The REST URI is further configured using the following properties:
    - **rest-uri-segment-scheme**

    - **rest-num-segments**
- {http-method}: HTTP method

Type: String

Default value: {rest-uri}.{http-method}

Platform: Java

Examples: Using Default Settings and Using rest-transaction-naming Properties

rest-uri-segment-scheme

Description: The property, rest-uri-segment-scheme has three valid values: first-n-segments, last-n-segments, and full. This property indicates how many segments of the URI to use for the URI in {rest-uri}. This option is case-sensitive. If the value of this property is full, then the value of rest-num-segments is ignored.

Type: String

Default value: first-n-segments

Platform: Java

rmqsegments

Description: The App Agent for .NET (agent) automatically discovers RabbitMQ remote services. Use the rmqsegments node property to refine the queue backend name to include some or all segments of the routing key. You must be familiar with your implementation RabbitMQ exchanges and routing keys. See RabbitMQ Exchanges and Exchange Types.

The RabbitMQ routing key is a string. The agent treats dot-separated (".") substrings of the routing key as segments. Set the value for rmqsegments to an integer that represents the number of routing key segments to include in the name. For more information, see To refine RabbitMQ backend naming.

Type: Numeric

Values: Integer representing the number of routing key segments to include in the name.

Default Value: 0

Platform: .NET

show-packages

Description: For the call graphs captured on this node, show the specified packages or class names in addition to the ones configured in the global call graph configuration. Does not need a restart.

Type: String

Default value: none

Platform: Java, .NET

slow-request-deviation

Description: The value in milliseconds for the deviation from the current average response time. This setting is used for evaluation of slow in-flight transactions. Also see the slow-request-threshold property for more details.

Type: Integer

Default value: 200

Range: Minimum=10; Maximum=3600

Platform: Java, .NET

slow-request-monitor-interval

Description: In-flight requests are checked for slowness in the interval specified by this property. The value is specified in milliseconds.

Type: Integer

Default value: 100

Range: Minimum=0; Maximum=3600

Platform: Java, .NET

slow-request-threshold

Description: In-flight requests taking more time than this threshold (in ms) with a deviation greater than the slow-request-deviation property from the current average response time are monitored to capture hot spots.

Type: Integer

Default value: 500

Range: Minimum=0; Maximum=3600

Platform: Java, .NET

spring-integration-receive-marker-classes

Description: Use this property to specify the class and method you have identified as suitable POJO entry points for Spring Integration.

Based on the MessageHandler interface, the App Agent for Java by default automatically discovers exits for all channels except 'DirectChannel.' In cases where a lot of application flow happens before the first MessageHandler is executed,

- set enable-spring-integration-entry-points=false,
- configure suitable POJO entry points,
- and declare each suitable POJO entry point class/method in this property spring-integration-receive-marker-classes.

If the application code polls for messages in a loop, the span of each loop iteration is tracked as a transaction. Tracking begins when the loop begins and end it when the iteration ends. To safeguard against cases where pollableChannel.receive() is not called inside a loop, specify this property for each class/method combination that polls messages in a loop.

After setting this property, restart the application server for changes to this property to take effect.

See also Spring Integration Support.

Type: Comma-separated string of fully-qualified class /method name, such as spring-integration-receive-marker-classes = <fully qualified name of class/method>,<> ....

Example: For example, to enable tracking for the following:

```
class MessageProcessor
{
void process()
{
   while(true)
   {
          Message message = pollableChannel.receive()
   }
}
}
```

set this property as follows:

```
spring-integration-receive-marker-classes = MessageProcesser/process
```

Platform: Java

thread-correlation-classes

Description: For multi-threaded applications, use this property to configure classes to be included in Java thread correlation when simple prefix-matching (matching on STARTSWITH) is sufficient to identify the classes.

The thread-correlation-classes property specifies the classes to include for thread correlation. This property can be used together with the thread-correlation-classes-exclude property.

The configured correlation takes effect without requiring a restart of the managed application. Also see: Configure Multi-Threaded Transactions for Java.

Type: Comma-separated string of fully-qualified class names or package names

Default value: none

Platform: Java

thread-correlation-classes-exclude

Description: For multi-threaded applications, use this property to configure classes to be excluded in Java thread correlation when simple prefix-matching (matching on STARTSWITH) is sufficient to identify the classes. This property specifies the classes to exclude from thread correlation. This property can be used in conjunction with the thread-correlation-classes node property.

The configured correlation takes effect without requiring a restart of the managed application. Also see: Configure Multi-Threaded Transactions (Java only).

Type: Comma-separated string of fully-qualified class names or package names

Default value: none

Platform: Java

thread-cpu-capture-overhead-threshold-in-ms

Description: Involved in BT CPU Monitoring. The is the time allotted to collect and calculate Thread CPU Time for 1000 iterations. You might need to increase this value if you see a symptom where there is no CPU usage information in the BT overview or in any of the snapshots. Restart is needed after changing this value.

Also see: SUN JDK 1.6 on Linux.

INFO level logs similar to the following will be logged if the agent is unable to collect the Thread CPU Time during the allotted time. As you can see in the logs below, the agent stops monitoring thread CPU time when this happens:

```
====================================
[Thread-0] 23 Oct 2013 17:07:18,704 INFO XMLConfigManager - Full Agent Registration Info Resolver using
node name [app07.com]
..
[Thread-0] 22 Oct 2013 14:19:26,346 INFO JVMThreadCPUTimeCalculator - Disabling BT CPU Monitoring. Time
taken to calculate Thread CPU Time for [1000] iterations is [15 ms] which is greater than the allowed
budget of [10 ms].
..
[Thread-0] 23 Oct 2013 17:07:19,273 INFO JVMThreadCPUTimeCalculator - Disabling BT CPU Monitoring. Time
taken to calculate Thread CPU Time for [1000] iterations is [15 ms] which is greater than the allowed
budget of [10 ms].
======================================
```

Type: Integer
Default: 10 ms
Platform: Java