webMethods Integration Server 10.7 Readme

October 2020

This file contains important information you must read before using webMethods Integration Server 10.5. You can find user documentation on the Documentation website or the TECHcommunity website. At those locations, you can also find suite-related security and globalization information.

Included in this file is information about functionality that has been added, removed, deprecated, or changed for this product. Deprecated functionality continues to work and is supported by Software AG but may be removed in a future release. Software AG recommends against using deprecated functionality in new projects.

1.0 Critical Information

This section lists any critical issues for the current release that were known when this readme was published. For critical information found later, go to the Knowledge Center on the Empower website.

2.0 Known Issues

This section lists any issues for the current release that were known when this readme was published. For known issues found later, go to the Knowledge Center on the Empower website.

- PIE-67716
  A secure outbound call made using Apache Commons HttpClient fails with error "javax.net.ssl.SSLException: Connection reset".
  Integration Server 10.7 uses OpenJSSE to provide TLSv1.3 support. When OpenJSSE is in use and a layered product, such as CloudStreams, uses the Apache Commons HttpClient to make outbound calls, the calls sometimes fail with the error: javax.net.ssl.SSLException: Connection reset. The outbound call fails when the request is sent on a connection that was previously established. That is, the outbound call fails only when reusing a keep-alive connection.
  There are two options for working around this issue:
  Option 1. Change the keep-alive timeout to a low value for those connections from the component that is logging the error. A low keep-alive timeout reduces the chance of reusing a stale connection -- the stale connection is what causes this issue.
  If the component happens to be a webMethods CloudStreams connector, use advanced connection configuration section to adjust keepalive timeout in one of the following ways:
  - Set the keepalive timeout to a really low value to essentially turn off reusing connections.
  - Set the keepalive timeout to a low value such that the connection is closed before it becomes stale (i.e., before the other Server or network layer closes the connection). Typically, server endpoints, network firewalls, etc., close the keepalive connections after a configurable time if they do not receive successive requests. The goal here is to close the connection from client side (i.e., the Integration Server/CloudStreams connector making that request) before the connection gets closed by the target server.
  Option 2. Remove the OpenJSSE provider from Integration Server. To do this:
  1. Shut down Integration Server.
  2. Open custom_wrapper.conf located under the directory: <Software AG_directory>/ profiles/IS_<instance-name>/configuration
  3. Comment out or remove the following line:
  wrapper.java.additional.105=-XX:+UseOpenJSSE
  4. Save custom_wrapper.conf.
  5. Restart the Integration Server.

- PIE-65289
  Integration Server Administrator shows a scheduled task in the running state even when the underlying service is not executing.
  To work around this issue, recreate the scheduled tasks.

- OGI-2585
  When using console.sh to start Integration Server, parameters passed through console.sh do not take effect.
  In addition, Docker images created for Integration Server will not generate stdout logs.
  To work around this issue, open console.sh file in a text editor and change
  $OSGI_INSTALL_AREA/bin/startup.sh

to
$OSGI_INSTALL_AREA/bin/startup.sh $*

- PIE-64352
  Scheduled tasks in Integration Server fail to execute after reactivating the associated solution in webMethods Cloud Container.
  The target node of a scheduled task retains the IP address of the associated solution despite reactivating the solution in Cloud Container. When a solution is reactivated in Cloud Container, a new hostname or IP address is allocated to it and the target node in the scheduled task is expected to change accordingly. However, the scheduled task fails to execute as its target node still points to the previous IP address.
  To work around this issue, update the scheduled task's target node with the new hostname or IP address of the solution on each reactivation.

- PIE-66089
  Validation of an SFTP User Alias fails in Command Central.
  When creating or editing an SFTP User Alias in Command Central, validation of the user credentials using the "Test" button fails even with the correct Password and PassPhrase.
  Currently, there is no workaround for this issue.

- PIE-66447
  Deleting a symbolic link to a Git package in Designer, deletes the Git package as well.
  Local Service Development projects in Designer that use Git, utilize a symbolic link to represent the Git package on the local system. If you delete this symbolic link, the Git package and its contents are also deleted. Committing or merging the local copy with the repository after this could cause the package to be deleted from the Git repository.
  Currently, there is no workaround for this issue.

- PIE-62959
  User Interface issue in Integration Server Administrator accessed from webMethods Cloud Container.
  If you are using Chrome and access Integration Server Administrator from webMethods Cloud Container, a user interface issue causes the page to automatically scroll upon clicking the “Settings” menu.
  To work around this issue, use a different browser.

- PIE-63099
  Integration Server returns a misleading message when you update a KEX algorithm for an SFTP server alias.
  When you update a KEX algorithm for an SFTP server alias, the algorithm is updated correctly. However, Integration Server returns a message indicating that a different algorithm is updated. You can ignore the message as it is incorrect.

- PIE-8533 (was 1-1Z6J9O)
  Integration Server does not shut down if an audit logging queue contains records waiting to be written to a destination.
  To work around this issue, wait for the records in the queue to be written to the destination.

- PIE-8045 (was 1-1Y1BZ7)
  Installing a package that contains a schema with a target namespace that is the same as an existing schema on the Integration Server may result in two sets of definitions or declarations for the same components.
  If you only need the schema definitions or declarations contained in one schema, delete the other schema. However, if you need definitions from both schemas, there is no workaround for this issue.

- WSC-515
  Known issue in the JVM Cipher implementation for certain JVM builds may cause decryption failures in Integration Server and other products that use the outbound password manager provided in Integration Server.
  A change in the PBEWithHmacSHA256AndAES_256 implementation causes Strings encrypted with the cipher in Open JDK builds 1.8.0_64 and before do not decrypt in later builds, beginning around build number 1.8.0_71. If your Integration Server runs on a JVM from Open JDK builds 1.80_64 up to, but not including 1.8.0_71, and then upgrade to Open JDK build 1.8.0_71 and after, passwords stored in the Integration Server outbound password manager cannot be decrypted.
  To avoid this issue, change the encryptor before upgrading the JVM. That is, store the password using the same handle but a different encryption code such as EntrustPbePlus.
  For information about changing the encryption method for the outbound password file and resetting outbound passwords, see the "Master Passwords and Outbound Passwords" chapter in the *webMethods Integration Server Administrator's Guide*. You can also change an outbound password using the pub.security.outboundPasswords:updatePassword service which is documented in the *webMethods Integration Server Built-In Services Reference*.
  A description of this Open JDK defect can be found here:
  https://bugs.openjdk.java.net/browse/JDK-8162690
  Note that the Open JDK defect description has some ambiguity regarding the build number in which this issue was introduced. The issue is observed in build number 1.8.0_74 but may have been introduced in earlier builds. Builds 1.8.0_64 through 1.8.0_74 were not tested by the defect reporter. The WSC-515 description cites build 1.8.0_71 as the Open JDK defect identities 71 as the "Introduced in version" value.

- PIE-8185 (was 1-1YBYQD)
  Web service connector ends with the error [ISC.0082.9034] Field is absent, field must exist.
  If the output signature of a service used as an operation in a provider web service descriptor (WSD) contains a field that has a namespace URI without a prefix, Integration Server adds a prefix when generating a WSDL document for the provider WSD. In the consumer WSD created from the WSDL, the web service connector that corresponds to the operation (IS service) specifies a prefix for the field in the service output. However, the web service provider does not include a prefix with the field in the response. As a result, the contents of the SOAP response cannot be mapped to the web service connector output and the web service connector ends with the error [ISC.0082.9034] Field is absent, field must exist.
  To avoid this issue, if a service will be exposed as a web service, always associate a prefix with a namespace URI for fields in the service signature.

- PIE-8494 (was 1-1Z342R)
  The xsi:nil attribute in an element does not convert properly when generating an IS document from

an XML document.
If an XML document has an element containing only xsi:nil as an attribute and an IS document is generated from that XML document using the pub.xml.xmlNodeToDocument service, the xsi:nil attribute is generated as an @xsi:nil field for the element in the resulting IS document. This occurs even if the element with the xsi:nil attribute has a simple type string; however, the document type that is created from the XML schema (which is used by the XML document) has a string field for the xsi:nil element instead of the IS document. There is a type difference between the generated document and the document type.
To work around this issue, manually edit the generated IS document to remove the @xsi:nil attribute and then convert the IS document to string field.

- PIE-16451
  WSDL generated for a web service descriptor with a service signature, header document type, or a fault document type containing derived document types does not contain the schema definitions for the derived document types.
  To work around this issue, create a WSDL with the schema definitions for the derived document types and then create a WSDL first web service descriptor.

- PIE-18649
  When creating a WSDL first provider web service descriptor, Integration Server does not preserve the original service name from the WSDL document.
  When Integration Server generates a WSDL document for the provider web service descriptor, the service name will not match the service name in the source WSDL document.
  There is currently no workaround for this issue.

- PIE-19157
  IMAP email listener does not start.
  This issue occurs when an IMAP e-mail port is configured to receive requests from an e-mail server that uses NTLM for authentication. With this configuration, the following error is returned when the port is enabled:
  "Failed to start EmailListener:imap: <UserName>@<HostName>: [ISS.0070.9003] Enable failed: Could not log into account <UserName>@<HostName>"
  To resolve this issue, do one of the following:
  If you want to disable NTLM authentication, follow these steps:
  1. Open custom_wrapper.conf located under
     <Software AG_directory>/ profiles/IS_default/configuration directory.
  2. Add the following property:
     wrapper.java.additional.n=-Dmail.imap.auth.ntlm.disable=true
     where n is the next unused sequential number in the file.
  3. Restart Integration Server.
     Note that this behavior is consistent with the past releases of Integration Server.

  If you want to enable NTLM authentication, follow these steps:
  1. Download jcifs-1.3.15.jar file from http://jcifs.samba.org/src/ into the
     <IntegrationServer_directory>/instances/<instanceName>/lib/jars or
     <IntegrationServer_directory>/lib/jars directory.
  2. Restart Integration Server.

- PIE-22556
  Java service throws java.lang.reflect.InvocationTargetException when attempting to use jars from <JRE_directory>/lib/ext directory, such as classes in the com.sun.crypto.provider package.
  If you plan to use jars from the <JRE_directory>/lib/ext directory, you can avoid this issue by modifying the config.ini as follows:
  1. Open the config.ini file located in <Software AG_directory>/profiles/IS_<instanceName>/configuration.
  2. Add the following line:
     osgi.parentClassloader=app
  3. Restart Integration Server.

  Note: This issue does not apply to Microservices Runtime.

- PIE-58945
  Integration Server does not support HTTP request or response compression for MTOM streaming and HTTP chunked transfer encoding.
  Unexpected results may occur when using HTTP request or response compression function with MTOM streaming and HTTP chunked transfer encoding.
  There is no workaround for this issue. Software AG recommends against using the HTTP compression function along with MTOM streaming and HTTP chunked transfer encoding.

- PIE-65633
  Importing an OpenAPI document fails when the format is "date" and also contains an enum property with a date value.
  When you create a provider REST API descriptor using an OpenAPI document that contains a property of type "string" and format as "date", and also contains an enum property with a date value, Integration Server throws an exception and does not import the OpenAPI document.
  There is no workaround for this issue.

- PIE-65385
  Integration Server throws an exception while importing an OpenAPI document containing a property of array type that holds an enum value.
  While creating a provider REST API descriptor using an OpenAPI document that contains a property of type "array" having an enum value, Integration Server throws an exception and does not import the OpenAPI document.
  There is no workaround for this issue.

  PIE-66057
  Integration Server may execute an incorrect service when a REST API containing ambiguous URL templates is invoked.
  When a client invokes a REST API that conforms to the Swagger specification but defines multiple ambiguous URL templates, then Integration Server may invoke an incorrect service. For example, consider the following two URL templates:
  test/1/{id}/{name} -> For this the mapped service is s1.
  test/1/2/3 -> For this the mapped service is s2.
  When a client invokes "test/1/2/3", Integration Server must invoke s2 service.
  However, Integration Server invokes s1 service.

There is no workaround for this issue.

- PIE-63433
  Integration Server incorrectly formats a multidimensional array variable in a REST API response of type 'application/xml'.
  While sending a REST API response in XML format for a service containing a multidimensional array variable in the output service signature, Integration Server does not format the variable correctly.
  There is no workaround for this issue.

- PIE-66393
  Integration Server does not honor the value of watt.server.stats.logFilesToKeep.
  Integration Server keeps an unlimited number of stats.log files regardless of the value of watt.server.stats.logFilesToKeep.
  To work around for this issue manually delete unwanted stats.log files.

- PIE-65416
  When running Integration Server or Microservices Runtime in an Alpine OS container, it may not be possible to create, edit, or synchronize a publishable document type with Universal Messaging. Requests to create or update the provider definition on Universal Messaging may fail with the following error:
  UM admin operation failed [create channel]: com.wm.util.coder.ProtoBufException: [ISC.0076.9021] Error invoking the protocol buffer compiler. Error: java.io.IOException: Cannot run program "/xxx/common/bin/protoc" (in directory "/xxx/IntegrationServer/bin/../packages/xxx/ns/yyy): error=2, No such file or directory.
  Where "xxx" and "yyy" in the message will contain one or more directory names.
  This issue occurs because the protoc executable distributed in the common\bin directory is not compatible with an Alpine OS container.
  To work around this issue, in the generated Dockerfile under the installation directory, after this line:
  ENV SAG_HOME /opt/softwareag.
  Add the following lines:
  RUN wget -q -O /etc/apk/keys/sgerrand.rsa.pub https://alpine-pkgs.sgerrand.com/sgerrand.rsa.pub && \
  wget -O /tmp/glibc-2.32-r0.apk https://github.com/sgerrand/alpine-pkg-glibc/releases/download/2.32-r0/glibc-2.32-r0.apk && \
  apk add /tmp/glibc-2.32-r0.apk && \
  rm /tmp/glibc-2.32-r0.apk

- PIE-63239
  Hostnames added to Global IP Access Restrictions and port IP Access do not have effect on access whether added to allow or deny list.
  When Integration Server determines whether a client is allowed to access a port, it uses the client's fully qualified hostname provided by the operating system's network implementation. Some operating systems do not return a fully qualified host name. They only return the simple host name without the subdomain.domain suffix. To make sure IP access checks work correctly, use simple

host names with a wildcard in the IP access allow list or deny list for a port and in the watt.server.hostAllow and watt.server.hostDeny properties. For example, rather than use "host01.east.mycompany", use "host01*".

- PIE-66274
  Importing a WSDL with many operations results in a StackOverFlow error.
  Issues in the internal XML serialization of a WSDL cause the service to run out of stack space when importing a WSDL with many operations. This results in a StackOverFlow error.
  There is no workaround for this issue.

- PIE-65618
  An Integration Server port should be enabled or disabled only using the API: POST /admin/port/internalserver/{alias}?action=(enable|disable). However, Integration Server incorrectly allows you to enable and disable a port using POST and PUT at /admin/port/internalserver as well. Ensure you use the correct API to enable and disable ports.

- PIE-66117
  The pub.client:smtp and pub.client:http services cannot process a javax.mail.internet.MimeMessage object.
  The pub.client:smtp and pub.client:http services cannot process a javax.mail.internet.MimeMessage object that is generated by the pub.mime:getEnvelopeStream service when the payload exceeds the watt.server.mime.largeDataThreshold.
  To work around this issue, the javax.mail.internet.MimeMessage object must be converted to a java.io.InputStream object before passing it to these services. Alternatively, consider increasing the value of the watt.server.mime.largeDataThreshold parameter to such an extent that the pub.mime:getEnvelopeStream service returns only InputStream objects.
  To convert a MimeMessage object to an InputStream, use the writeTo() method of MimeMessage to write to a file and then read the file as an InputStream.

- PIE-66046
  The pub.mime:getEnvelopeStream service could generate an out of memory error when adding a large number of files.
  There is no workaround for this issue.

- PIE-65268
  The pub.json:documentToJSONString service returns Boolean, null, integer and string values as strings.
  To work around this issue, check the returned value and convert it to the required data type.

- PIE-63000
  External port in API Gateway cannot be updated after changing the registration port.
  In an API Gateway instance that has an external port, a registration port, and an internal port; any attempt to update the external port after updating the registration port generates an error.
  There is no workaround for this issue.

- PIE-65481
  Some System tasks in the Integration Server may show a negative time interval.

There is no workaround for this issue.

- PIE-62179
  Updating the Integration Server license key in Command Central is not reflected in Integration Server Administrator.
  Integration Server Administrator displays "License Key is Expired or Invalid" even after a valid license key is added on the IS Configuration page of Command Central.
  There is no workaround for this issue.

- PIE-66119
  An input file path for the pub.parquet:write service that contains a consecutive backslash and forward slash may cause unexpected results. For example, the path "packages/replicatePackage/resources\/testService2.parquet" in Linux creates a directory named "resources\" that cannot be traversed.
  To work around this issue, ensure the paths passed to the service are in the appropriate format for the operating system.

- PIE-65750
  The pub.json:documentToJSONString service does not handle the escape character correctly.
  There is no workaround for this issue.

- PIE-66452
  When the pub.mime:getEnvelopeStream service is invoked with the createMultipart parameter set to "no", a java.io.InputStream object is returned. Such invocations may generate an out of memory error if the body parts of the MIME data are large.
  There is no workaround for this issue.

- PIE-64674
  Integration Server Administrator shows the hostname of the proxy server instance of Integration Server during reverse invoke.
  If an Integration Server instance is configured as a proxy, then all requests are directed to the target Integration Server instance. However, Integration Server Administrator shows the hostname of the proxy server instance.
  There is no workaround for this issue.

- PIE-65372
  The pub.json:documentToJSONString service converts strings that contain Unicode values to the corresponding Unicode characters. For example, if the document contains the string "\u2605", the documentToJSONString service converts it to "★".
  To work around this issue, use an extra backslash with the Unicode value. For example, "\\u2605".

- PIE-61906
  The preferred algorithm sections for an SFTP server alias are empty when you try to include the excluded algorithms. Additionally, the algorithms are not sorted in any specific order.
  In the Create Server Alias page, for Version 2 SFTP server alias, when you click the left arrow button to include a set of excluded algorithms, the corresponding preferred algorithm section is blank. Secondly, the algorithms are not sorted.

To work around the first issue, select the set of excluded algorithms and click the left arrow button. However, there is no workaround to sort the algorithms.

- PIE-65498
  Integration Server generates an incorrect output service signature on importing an OpenAPI document.
  When you create a provider REST API descriptor using an OpenAPI document that contains an operation response referencing an empty schema, Integration Server generates an incorrect document structure in the output service signature.
  There is no workaround for this issue.

- DTS-1149
  The filter in the Show Columns dialog of the data table on the Dashboard > APIs tab of the new Administration Console does not work as expected.
  The data table on the Dashboard > APIs tab of the new Administration Console can be configured to show or hide an available column by clicking the settings icon > Show Columns. The filter available in this dialog does not correctly filter the list of columns displayed in the dialog.
  To work around this issue: 1) clear the filter control, and 2) select or deselect the columns that you want to display or hide respectively.

- DTS-1212
  The Security menu item in the navigation panel of the new Administration Console has not been localized and is displayed in English in the localized versions of the Integration Server Administrator as well.

- DTS-1235
  The behavior and appearance of the Common Directory Services (CDS) pages in the new Administration Console varies slightly from other pages in the new Administration Console. For example, some of the dialog overlays, tooltips, and icons in CDS pages may behave and look different from similar controls on other pages of the new Administration Console.
  You can refer to the Integration Server documentation for the CDS pages if the behavior of any control is not comprehensible.

- DTS-1270
  The Date Time control for selecting the period for which data is presented in the Service Errors row/Current column of the Request Summary table in the Dashboard > Usages tab of the new Administration Console has not been localized. The names of the months and labels on the buttons are displayed in English in the localized versions of the Integration Server Administrator as well.

- DTS-1283
  Selecting an adapter in the navigation panel of the new Administration Console (Integration Server Administrator > Try the New Administration Console > Adapters > <adapter_name>), should open the adapter details page in a new browser tab. While Integration Server Administrator opens a new browser tab with the adapter details, it also displays the adapter details in the original browser tab (the tab from which the new Administration Console was opened). The same issue is also observed for solutions (Integration Server Administrator > Try the New Administration Console > Solutions > <solutions_name>).

3.0 Usage Notes

This section provides any additional information you need to work with the current release of this product.

- Beginning with the 10.7 release, webMethods Integration Agent is no longer available. Use webMethods Microservices Runtime instead.

- Integration Server 10.7 supports TLS v1.3 for secure inbound and outbound connections that use JSSE. Integration Server 10.7 no longer includes TLSv1.0 as a default enabled protocol. To support this change, Integration Server removes the server configuration parameters watt.net.jsse.client.enabledProtocols and watt.net.jsse.server.enabledProtocols, replacing them with watt.net.jsse.client.disabledProtocols and watt.net.jsse.server.disabledProtocols, respectively. During migration of an earlier version of Integration Server to version 10.7, Integration Server detects the current values of watt.net.jsse.client.enabledProtocols and watt.net.jsse.server.enabledProtocols. Integration Server then uses those values to build the values for watt.net.jsse.client.disabledProtocols and watt.net.jsse.server.disabledProtocols.

- TLS v1.3 support is provided via the OpenJSSE from Azul. OpenJSSE is available in the Zulu 8 JDK. Adding TLS v1.3 as a supported protocol is possible by using java system property -XX:+UseOpenJSSE. Integration Server adds this system property to the profiles/IS_<instanceName>/configuration/custom_wrapper.conf when creating an Integration Server instance. Microservices Runtime startup scripts server.bat|sh have been modified to include this parameter as well. If you decide to use a JDK that is not Zulu, the OpenJSSE functionality, and therefore TLS v1.3, may not be available. To use a JVM from another provider, you may need to either remove or comment out the -XX:+UseOpenJSSE option from the custom_wrapper.conf file in Integration Server and/or from the server.bat|sh file for Microservices Runtime.

- When creating a keystore alias for a PKCS12 type keystore in version 10.7, Integration Server lists BC (Bouncy Castle) and OpenJSSE as possible providers for the keystore. In versions of Integration Server prior to 10.7, Integration Server listed SunJSSE as a possible provider for a PKCS12 type keystore. However, when using OpenJSSE, the SSL provider named SunJSSE is not available. During migration to Integration Server 10.7 that uses OpenJSSE, the migration utility changes existing keystores that used SunJSSE as the provider to use BC (Bouncy Castle).

- XML Schema 1.1 is not supported on AIX.
  The Xerces library in AIX supports XML Schema 1.0 only. Attempts to use XML Schema 1.1 features on an Integration Server running on AIX results in schema syntax errors and the schema does not compile.

- Use of MD5 signing algorithm is not supported when Integration Server runs in FIPS mode.

- Beginning in Integration Server 10.7, the pub.mime services have been enhanced to support large payloads. If any of your existing flow services use the pub.mime:createMimeData or the pub.mime:getEnvelopeStream services and you want to take advantage of the new functionality, you need to modify the follow services to use a javax.mail.internet.MimeMessage object instead of an InputStream. This would involve using the pipeline to change mapping for the service in

question. Further, the watt.server.mime.largeDataThreshold value determines if the pub.mime:getEnvelopeStream service returns a javax.mail.internet.MimeMessage object instead of an InputStream. If you do not want invocations of pub.mime:getEnvelopeStream service to return a javax.mail.internet.MimeMessage object, increase the value of watt.server.mime.largeDataThreshold. You could also use convert a javax.mail.internet.MimeMessage to an InputStream using the javax.mail.internet.MimeMessage.writeTo().

- Statistics Data Collector, which is the Integration Server component responsible for the capture and retrieval of statistical information for its service invocations, performs a scheduled restart every 24 hours. The purpose of the restart is to minimize the disk file consumption produced by its document index from an embedded third-party library (org.apache.lucene). Restart of the Statistics Data Collector does not affect a running Integration Server.

- Integration Server does not support HTTP request or response compression for dynamic server pages (DSPs).

- webMethods Broker is deprecated.

- Before importing a Swagger document into Integration Server, validate the document using an external tool.

- In Universal Messaging version 10.0 the implementation of shared durables was rewritten to improve stability and performance. However, these changes were not compatible with the pub.publish:deliver* services which relied on the previous implementation of shared durables to ensure that delivered documents were routed by Universal Messaging to the intended destination. As of version 10.3 (and in fixes delivered on earlier releases), Universal Messaging implemented subscriber name filtering for its durable subscriptions. In subscriber name filtering, when a publisher designates a message for a specific subscriber, Universal Messaging routes the message to a consumer whose durable subscription name matches the designated subscriber name in the message. For Integration Server, the subscription name filtering requires some modifications, specifically:

  - Enable the subscription name filtering feature on Universal Messaging To enable the feature through Universal Messaging Enterprise Manager, navigate to the realm and select the Config tab. In the realm server configuration panel, click Show Advanced Config. Expand Advanced Configuration - Durable Config and set Durable Name Filtering to true.

  - Change the destID input parameter value in invocations of pub.publish:deliver and pub.publish:deliverAndWait, if necessary. This may only be necessary when the trigger to which the message is to be delivered contains an underscore in its name. For more information about client IDs for triggers and how this change may impact existing services that invoke pub.publish:deliver*, see the *Publish-Subscribe Developer's Guide*.

- PIE-58995
  Previously, when creating a Dockerfile for an Integration Server that runs on Windows, the -Dimage.name parameter did not need to be specified because the is_container scripts referenced the base image tagged "latest" for Windows server containers. However, Microsoft now provides

Windows image tags per OS version instead of a “latest” base image. This change prevents a Dockerfile from being created for an Integration Server running on Windows when the -Dimage.name parameter was not specified.
For an Integration Server running on Windows, the -Dimage.name parameter should be set as follows:
-Dimage.name=mcr.microsoft.com/windows/servercore:YourOsImageTag
Where YourOsImageTag is the tag for the Windows image created for the OS version on which Integration Server runs.

- Integration Server 10.4 and previous versions to which a fix including PIE-51049 was applied provide stricter validation for an inbound SOAP request. Validation of an inbound SOAP request message fails if the SOAP request body includes input fields that are not part of the service input signature. This behavior change may be problematic for existing solutions. Integration Server includes a server configuration parameter named watt.server.soap.validateInput that can be used to control whether or not a validation error occurs when the inbound SOAP request includes fields that are not declared in the service input signature. Set this parameter to false to obtain the behavior that existed prior to 10.4 or applying a fix that includes PIE-51049.

- Take care when setting the watt.security.max* server configuration parameters. If the parameters values are set too low, Integration Server rejects legitimate requests. It is also possible to lock yourself out of Integration Server Administrator by setting the parameters too low. If this happens, you can try changing the configuration parameter values by invoking the wm.server.admin:setSettings service with a command-line HTTP client. If you are still unable to change the value of these parameters, you need to kill the Integration Server process, manually edit the parameters in the Integration Server_directory/instances/instanceName/config/server.cnf file, and restart Integration Server.

- PIE-32205
  Integration Server does not provide Java-based NTLM (Windows NT LAN Manager) support for proxy servers that support NTLM authentication. You can only use the NTLM authentication support in Integration Server to allow clients to access resources on web servers that support NTLM authentication, such as Microsoft Internet Information Server (IIS).

- PIE-16497
  Integration Server does not generate the *doctype field for IS document types generated from derived document types in a schema, when:
  - Deriving a complex type from an empty complex type by extension.
  - Deriving a complex type from a simple type by extension.

- PIE-42419
  As of Integration Server 10.0, the Allow List for any new port with an Access Mode of “Deny by Default” now includes the wm.server.csrfguard:getCSRFSecretToken service and the wm.server.csrfguard:isCSRFGuardEnabled service. When migrating to Integration Server 10.0 or higher from a version lower than 10.0, for any existing ports with an Access Mode of “Deny by Default”, you must manually update the Allow List to include the wm.server.csrfguard:getCSRFSecretToken and wm.server.csrfguard:isCSRFGuardEnabled services.

- PIE-48484
  A web service request that specifies a content type of application/xml succeeds in versions of Integration Server prior to version 9.12. However, the web service request fails in Integration Server version 9.12. Prior to Integration Server 9.12, Integration Server did not provide out of the box support to directly handle the content type application/xml. If a content handler was not specified for application/xml, Integration Server used the default content handler. The default handling is the same as the handling of text/html, resulting in XML being placed in the pipeline as an inputStream . In version 9.12, Integration Server added a content handler named ContentHandler_XML to support the content type application/xml, resulting in the content being placed in the pipeline as an XML node. This behavior change causes the web service request to fail in version 9.12 because the web service expects an inputStream, not an XML node. To address this issue and avoid having to change the service to accept an XML node instead of an inputStream, you can use the watt.server.content.type.mappings server configuration parameter to map the application/xml content type to the text/xml content type. Integration Server will use the text/xml content handler when receiving a request that specifies a content type of application/xml. Specify the following: watt.server.content.type.mappings=application/xml text/xml

- If your solution includes publishable document types with an encoding type of protocol buffers, you need to edit, save, and synchronize the publishable document types to Universal Messaging while the queues of subscribing triggers are empty. This ensures that subscribing triggers will not fail with a NullPointerException if you later choose to edit the publishable document type while queues of subscribing triggers contain published documents that conform to earlier iterations of the publishable document type.

- Integration Server 10.0 introduces the ability to enable or disable follow the master behavior on a per messaging connection alias basis. Prior to Integration Server 10.0, follow the master was enabled globally for webMethods messaging. The only way to disable follow the master behavior was to modify the custom_wrapper.conf file to include the wrapper.java.additional.n=-DFollowTheMaster=false where n was the next available wrapper.java.additional number. Do not use the DFollowTheMaster parameter to control the follow the master behavior for Integration Server. This parameter setting will override the follow the master behavior set in Universal Messaging connection aliases and JMS connection aliases.

- When securing web services using policies based on WS-SecurityPolicy, you cannot alter an inbound message before the security processing executes or alter an outbound message after the security processing completes. For inbound messages, Integration Server always performs the security processing first upon receiving the message. As a result, Integration Server cannot invoke custom handlers before the security processing of an inbound message. For outbound messages, Integration Server always performs the security processing last, right before it sends the message. As a result, Integration Server cannot invoke custom handlers after the security processing of an outbound message.

- Integration Server uses Xerces Java parser version Xerces-J 2.12.1-xml-schema-1.1. Limitations for this version are listed at https://xerces.apache.org/xerces2-j/.

- If you want to use WS-SecurityPolicy to secure a web service and want to use MTOM streaming, be

aware that if the fields to be streamed are being signed and/or encrypted, Integration Server cannot use MTOM streaming because Integration Server needs to keep the entire message in memory to sign and/or encrypt the message.

- Web services security implemented using WS-Security facility in Integration Server does not support partial message operations (Sign/Encrypt). Integration Server allows only the body of the SOAP message to be signed and encrypted.

- Do not modify the following file unless instructed to do so by Software AG:
  <IntegrationServer_directory>\instances\<instanceName>\config\wss\axis2.xml
  Changes to this file may result in an unstable configuration. Software AG will not support issues that arise because of changes to this file that were not authorized by Software AG.

- Software AG does not support the deployment of custom handlers or modules via placement of an Axis Module (*.mar) file in the following directory:
  <IntegrationServer_directory>\instances\<instanceName>\config\wss\modules
  Unexpected behavior that arises due to the manual deployment of mar files directly to this location is the responsibility of the user and will not be addressed by Software AG.

- Software AG does not support the deployment of web services via placement of an Axis Archive (*.aar) file in the following directory:
  <IntegrationServer_directory>\instances\<instanceName>\config\wss\services
  Unexpected behavior that arises due to the manual deployment of aar files directly to this location is the responsibility of the user and will not be addressed by Software AG.

- In Integration Server version 10.1, the default behavior of the enhanced XML parser now prohibits the support of DTDs and resolution of external entity references. For the pub.xml:loadEnhancedXMLNode and pub.xml:XMLStringToEnhancedXMLNode services the inputProcessing\supportDTD input parameter and the inputProcessing\isSupportingExternalEntities now have a default value of false. If an existing solution relies in the previous default behavior (inputProcessing\supportDTD and inputProcessing\isSupportinExternalEntities were set to true), after migrating to Integration Server 10.1 or higher, you must update invocations of pub.xml:loadEnhancedXMLNode and pub.xml:XMLStringToEnhancedXMLNode to set inputProcessing\supportDTD and\or inputProcessing\isSupportinExternalEntities to true.

- Now, when you start Integration Server, Integration Server receives configuration settings (for example, the size of the Java heap) from the wrapper.conf and custom_wrapper.conf files located in the *Software AG_directory*\profiles\IS_*instance_name*\configuration directory. Integration Server no longer obtains settings from setenv.bat/sh or server.bat/sh.
  If you need to modify the default property settings for Integration Server, you can override the settings using the custom_wrapper.conf file. The following table shows the settings formerly set in the setenv.bat/sh file that are now set using properties in the custom_wrapper.conf file:

This setting in setenv.bat/sh…	**Is replaced with the following property in custom_wrapper.conf…**

APPENDCLASSES/ APPEND_SYSTEM_CLASSPATH	wrapper.java.additional.203=-Dwatt.server.append.classes=
JAVA_CUSTOM_OPTS	wrapper.java.additional.*n*
JAVA_MAX_DIRECT_SIZE	wrapper.java.additional.*n*=XX:MaxDirectMemorySize=
JAVA_MAX_MEM	wrapper.java.maxmemory
JAVA_MAX_PERM_SIZE	wrapper.java.additional.*n*=XX:MaxPermSize=
JAVA_MIN_MEM	wrapper.java.initmemory
PREPENDCLASSES/ PREPEND_SYSTEM_CLASSPATH	wrapper.java.additional.202=-Dwatt.server.prepend.classes=

The following table shows settings you can change that were formerly in the setenv.bat/sh file, but are now located in other files or removed:

Setting	**New location (if applicable)**
DEBUG_ENABLED	*Software AG_directory*\profiles\IS_*instance_name*\bin\startDebugMode.bat/sh
JAVA_PROFILER_OPTS/ PROFILER_ENABLED	Removed.
Java location	wrapper.java.command in *Software AG_directory*\profiles\IS_*instance_name*\configuration\wrapper.conf
JMX_ENABLED	Removed. JMX monitoring is enabled by default and cannot be disabled.
JMX_PORT	The port property in *Software AG_directory*\profiles\IS_*instance_name*\configuration\com.software.jmx.connector.pid-*port*.properties

The startup.bat/sh and shutdown.bat/sh scripts contained in the *Integration Server_directory*\instances*instance_name*\bin and *Integration Server_directory*\bin directories are deprecated. You should use the scripts contained in the *Software AG_directory*\profiles\IS_*instance_name*\bin directory to start and stop Integration Server. If you will manage Integration Server through Command Central, you *must* use the scripts located in the *Software AG_directory*\profiles\IS_instance_name\bin directory.

The installSvc.bat file located in *Integration Server_directory*\instances*instance_name*\support\win32 directory is also deprecated. You should use the service.bat file from the *Software AG_directory*\profiles\IS_*instance_name*\bin directory to register or remove an Integration Server instance as a Windows service.

For complete instructions for using any of the features affected by these changes, see *webMethods Integration Server Administrator's Guide* and *Working with the webMethods Product Suite and the Java Service Wrapper.*

4.0 Fixes Included in Each Release

This section lists the latest fix level that has been included in each release for each product component. A release is listed in this section only if changes occurred in that release. Go to the Knowledge Center on the Empower website for detailed information about fixes.

Release 10.7

- IS_9.5_SP1_Core_Fix23
- IS_9.6_Core_Fix23
- IS_9.7_Core_Fix34
- IS_9.8_Core_Fix30
- IS_9.9_Core_Fix36
- IS_9.10_Core_Fix24
- IS_9.12_Core_Fix28
- IS_9.12_SPM_Fix9
- IS_9.12_WmCloud_Fix5
- IS_10.1_Core_Fix20
- IS_10.1_SPM_Fix7
- IS_10.1_WmCloud_Fix5
- IS_10.3_ABE_Fix2
- IS_10.3_Core_Fix13
- IS_10.3_Migration_Fix1
- IS_10.3_SPM_Fix8
- IS_10.3_WmCloud_Fix4
- IS_10.5_ABE_Fix2
- IS_10.5_Core_Fix6
- IS_10.5_Database_Fix1
- IS_10.5_Migration_Fix2

- IS_10.5_SPM_Fix5
- WAR_9.8_Fix5
- WAR_9.10_Fix5
- WAR_9.12_Fix5
- WAR_10.1_Fix6
- WAR_10.3_Fix13
- WAR_10.5_Fix4

Release 10.5

- IS_9.5_SP1_Core_Fix21
- IS_9.6_Core_Fix21
- IS_9.7_Core_Fix29
- IS_9.8_Core_Fix25
- IS_9.9_Core_Fix32
- IS_9.10_Core_Fix19
- IS_9.10_SPM_Fix3
- IS_9.12_Core_Fix22
- IS_9.12_SPM_Fix7
- IS_10.1_Core_Fix13
- IS_10.1_SPM_Fix5
- IS_10.3_Core_Fix5
- IS_10.3_SPM_Fix2
- IS_10.4_Core_Fix2
- IS_10.4_SPM_Fix1
- WAR_9.12_Fix4
- WAR_10.4_Fix2
- WFF_9.12_Fix4

- WFF_10.1_Fix6

Release 10.4

- IS_9.5_SP1_Core_Fix20
- IS_9.6_Core_Fix20
- IS_9.7_Core_Fix28
- IS_9.8_Core_Fix25
- IS_9.9_Core_Fix29
- IS_9.10_Core_Fix17
- IS_9.12_Core_Fix21
- IS_9.12_SPM_Fix6
- IS_9.12_WmCloud_Fix3
- IS_10.1_AssetPublisher_Fix1
- IS_10.1_Core_Fix11
- IS_10.1_MobileSupport_Fix1
- IS_10.1_WmCloud_Fix1
- IS_10.3_Core_Fix3
- IS_10.3_SPM_Fix2
- IS_10.3_WmCloud_Fix1
- WAR_9.8_Fix4
- WAR_9.9_Fix3
- WAR_9.12_Fix3
- WAR_10.1_Fix4
- WAR_10.3_Fix4
- WFF_9.7_Fix7
- WFF_10.1_Fix3
- WFF_10.3_Fix2

Release 10.3

- IS_9.5_SP1_Core_Fix19
- IS_9.7_Core_Fix25
- IS_9.8_Core_Fix21
- IS_9.9_Core_Fix26
- IS_9.10_Core_Fix15
- IS_9.12_Core_Fix17
- IS_9.12_SPM_Fix5
- IS_10.1_Core_Fix5
- IS_10.1_SPM_Fix3
- IS_10.2_Core_Fix2
- MIG_9.10_MigrationFramework_Fix1
- MIG_9.12_MigrationFramework_Fix4,
- WFF_10.1_Fix2

Release 10.2

- IS_8.2_SP2_Core_Fix24
- IS_9.0_SP1_Core_Fix21
- IS_9.5_SP1_Core_Fix18
- IS_9.6_Core_Fix18
- IS_9.7_Core_Fix22
- IS_9.8_Core_Fix18
- IS_9.9_Core_Fix22
- IS_9.10_Core_Fix11
- IS_9.12_Core_Fix14
- IS_9.12_SPM_Fix4
- IS_10.0_Core_Fix3

- IS_10.0_WmCloud_Fix3
- IS_10.1_Core_Fix3
- IS_10.1_SPM_Fix1
- MIG_9.12_MigrationFramework_Fix3
- MIG_10.1_MigrationFramework_Fix1
- WFF_9.7_Fix6
- WFF_9.9_Fix4
- WFF_9.12_Fix3

Release 10.1

- IS_8.2_SP2_Core_Fix23
- IS_9.0_SP1_Core_Fix20
- IS_9.5_SP1_Core_Fix17
- IS_9.6_Core_Fix16
- IS_9.7_Core_Fix19
- IS_9.8_Core_Fix15
- IS_9.8_Tanuki_Fix1
- IS_9.9_Core_Fix15
- IS_9.9_SPM_Fix4
- IS_9.10_Core_Fix9
- IS_9.10_SPM_Fix2
- IS_9.10_Tanuki_Fix
- IS_9.12_Core_Fix7
- IS_9.12_SPM_Fix2
- IS_9.12_WmCloud_Fix2
- IS_10.0_Core_Fix1
- IS_10.0_WmCloud_Fix2

- MIG_9.12_MigrationFramework_Fix1
- WAR_9.7_Fix3
- WAR_9.8_Fix4
- WAR_9.9_Fix2
- WAR_9.10_Fix3
- WAR_9.12_Fix1
- WFF_9.6_Fix5
- WFF_9.9_Fix3
- WFF_9.10_Fix1
- WFF_9.12_Fix1

Release 10.0

- IS_8.2_SP2_Core_Fix22
- IS_9.0_SP1_Core_Fix18
- IS_9.5_SP1_Core_Fix14
- IS_9.6_Core_Fix13
- IS_9.7_Core_Fix14
- IS_9.7_SPM_Fix3
- IS_9.7_SubVersion_Fix1
- IS_9.7_Tanuki_Fix1
- IS_9.7_VCS_Fix1
- IS_9.7_WmCloud_Fix4
- IS_9.8_Core_Fix11
- IS_9.8_SPM_Fix2
- IS_9.8_VCS_Fix1
- IS_9.8_WmCloud_Fix1
- IS_9.9_Core_Fix9

- IS_9.9_SPM_Fix3
- IS_9.9_Tanuki_Fix1
- IS_9.9_VCS_Fix1
- IS_9.9_WmCloud_Fix1
- IS_9.10_Core_Fix6
- IS_9.10_SPM_Fix1
- IS_9.10_VCS_Fix1
- IS_9.10_WmCloud_Fix1
- IS_9.12_Core_Fix4
- IS_9.12_WmCloud_Fix1
- WAR_9.6_Fix5

Release 9.12

- IS_8.0_SP1_Core_Fix32
- IS_8.2_SP2_Core_Fix20
- IS_9.0_SP1_Core_Fix14
- IS_9.5_SP1_Core_Fix11
- IS_9.7_Core_Fix8
- IS_9.7_Tomcat6_Fix2
- IS_9.8_Core_Fix5
- IS_9.9_Core_Fix2
- IS_9.9_SPM_Fix2
- IS_9.10_Core_Fix1
- IS_9.10_OData_Fix1
- WAR_9.7_Fix2
- WAR_9.8_Fix3
- WAR_9.10_Fix2

- WFF_8.2_SP2_Fix6
- WFF_9.5_SP1_Fix4
- WFF_9.6_Fix4
- WFF_9.8_Fix1
- WFF_9.9_Fix1

Release 9.10

- IS_7.1.3_Core_Fix28
- IS_8.2_SP2_Core_Fix19
- IS_9.0_SP1_Core_Fix13
- IS_9.5_SP1_Core_Fix10
- IS_9.5_SP1_Portal_Fix3
- IS_9.6_Core_Fix9
- IS_9.7_Core_Fix5
- IS_9.7_Portal_Fix4
- IS_9.8_SPM_Fix1
- IS_9.9_Core_Fix1
- IS_9.9_SPM_Fix1
- WAR_9.5_SP1_Fix6
- WAR_9.8_Fix2
- WFF_8.2_SP2_Fix5
- WFF_9.0_SP1_Fix2
- WFF_9.5_SP1_Fix3
- WFF_9.6_Fix1
- WFF_9.7_Fix4

5.0 Other Resolved Issues

This section lists the issues that were resolved in each release but were not part of the fixes listed in the previous section. A release is listed in this section only if changes occurred in that release.

Release 10.7

- PIE-55371
  Creating a JSON document type fails if the enum keyword contains an empty array as one of its elements in JSON schema.
  When creating a JSON document type if JSON schema contains enum keyword with an empty array as one of its elements, then Integration Server writes a service exception to the error log.
  This issue is resolved.

- PIE-55402
  Integration Server does not create the URL alias automatically while creating a REST API descriptor (RAD) using resource first approach.
  While creating a RAD using the resource first approach, Integration Server does not create the URL alias automatically if the basepath is updated.
  This issue is now resolved.

- PIE-55771
  JSON schema validation fails but error messages are not clear as to the cause.
  If validation fails for the JSON schema constructs 'anyOf', 'allOf' and 'oneOf' keyword, the error messages do not clearly indicate the cause or location of the validation failure.
  This issue is now resolved.

- PIE-57239
  Integration Server generates invalid Swagger document after creating a provider REST API descriptor (RAD).
  When user creates a provider RAD using a Swagger document that contains an inline schema in the body or in the response, then Integration Server generates an invalid Swagger document while creating the provider RAD.
  This issue is resolved.

- PIE- 57903
  Attempting to create a JSON document type from a JSON schema that contains a cyclic reference at the property level might cause out of memory errors.
  This issue is now resolved.

- PIE-58826
  When deploying the JNDI configuration for ActiveMQ to Cloud, the provider URL is incorrectly changed to point to Universal Messaging.
  This issue is now resolved.

- PIE-58587

Designer should not allow adding an OAuth type security definition for a REST API descriptor if the scope name added in Integration Server contains a large string. Currently Designer does not display any error while adding the security definition.
Desginer now displays the following message when the scope length is too long: The scope name, *scopeName* exceeds the maximum length.

- PIE-58525
  Integration Server allows to access a resource with incorrect authentication type.
  After assigning OAuth type security definition for a resource, Integration Server allows accessing the resource using basic authentication without any error.
  This issue is resolved.

- PIE-58673
  Integration Server adds incorrect output to the REST connector service while executing it.
  While executing a REST connector service, if the REST API returns an undefined data in the response, then Integration Server adds that data in the connector service output.
  This issue is resolved.

- PIE-58658
  Integration Server throws an exception while creating a REST API descriptor (RAD) from a Swagger file containing a path defined as '/test/{id}-{name}'.
  While importing a valid Swagger file to create a RAD, if the Swagger file contains a path having multiple path parameters in a single URI segment, Integration Server throws a validation error.
  This issue no longer occurs.

- PIE-58847
  Integration Server throws NullPointerException while calling a connector service in a consumer REST API descriptor (RAD).
  If a consumer RAD is created and then renamed, an attempt to run any of the connector service, ends with a NullPointerException.
  There is no workaround for this issue.

- PIE-58862
  Using the Integration Server Administrator UI to set the Kerberos "Use Subject Credentials Only" property (which corresponds to the "javax.security.auth.useSubjectCredsOnly" field) on the Security > Kerberos > Kerberos Settings screen sets an incorrect property.
  This issue is now resolved.

- PIE-58919
  A StringIndexOutOfBoundsException occurs when the root service wm.server.apigateway:getApiData tries to fetch details of published APIs from an API Gateway instance running on a different machine.
  This issue is now resolved.

- PIE-58996
  Consumer endpoint aliases with Kerberos as the authentication method are not editable.
  This issue is now resolved.

- PIE-59175
  Command Central displays incorrect status of the OSGI-IS_{is_instanceName} component.
  Command Central incorrectly displays the status of the OSGI-IS_{is_instanceName} component in the Instances tab as “STARTING” even when the child nodes do not have a Platform Manager plugin Offline implementation. This issue occurs only after an Integration server restart.
  This issue is now resolved.

- PIE-59266
  Designer displays the erroneous input parameter 'secure' under the service signature of pub.client:restClient service.
  This issue is resolved.

- WFF-612
  Composite definitions for fixed-length record parsers without delimiters do not get parsed.
  This issue is resolved. Support for composite definitions for fixed-length record parsers without delimiters can now be done through flat file schema generation wizard.

- WFF-636
  In Integration Server, when the newly added "recordIdentifierPre105Version" property is set to true the record identifier will not be added to pub.flatFile:convertToString service output for composite fields.

- WFF-664
  There is no input to handle line separators other than modifying the flat file schema record delimiter based on the Operating System where Integration Server is running.
  To resolve this, a new input 'lineSeparator' is added to the delimiters input document in the pub.flatFile:convertToString service to specify how handle CRLF/LF/CR line separators in the service output.

Release 10.3

- PIE-48958 [IS_10.1_Core_Fix5]
  When creating a document type from XML string in Integration Server, an XML external entity attack was possible during the validation of XML string.
  The issue is now resolved. After the fix, Integration Server rejects such XML with the following message - "XML provided is not well-formed or external references encountered while parsing XML - <specific error message>". - "

- PIE-47905 [IS_10.2_Core_Fix2, IS_9.10_Core_Fix12, IS_9.12_Core_Fix15]
  Application Platform does not initialize properly when Integration Server startup does not complete within 5 minutes.
  To resolve this issue, Integration Server registers the OSGi services before continuing with the rest of startup. This allows Application Platform to initialize properly even if Integration Server takes longer than 5 minutes to complete startup.

- PIE-50428 [IS_10.2_Core_Fix2, IS_9.12_Core_Fix16]
  Element in outbound SOAP message is associated with the incorrect namespaces.
  When an attribute on an element redefines the XML namespace prefix for an element to be a different URL from the one defined in the WSDL document and WS-Security processing is engaged, the element in the outbound SOAP message is associated with the incorrect namespace.
  This issue is now resolved.

- PIE-50562 [IS_10.2_Core_Fix2, IS_9.12_Core_Fix16]
  Generation of a consumer web service descriptor or WSDL first provider web service descriptor fails with a WSDLException.
  If a WSDL document contains an empty targetNamespace attribute, an attempt to create a web service descriptor from the WSDL document fails with the following exception:
  WSDLException: faultCode=PARSER_ERROR: Problem parsing '- WSDL Document -'.: org.xml.sax.SAXParseException: The value of the attribute "prefix="xmlns",localpart="tns",rawname="xmlns:tns"" is invalid. Prefixed namespace bindings may not be empty.
  Now, when creating a web service descriptor from a WSDL document that contains an empty targetNamespace attribute, Integration Server no longer generates the xmlns:tns= attribute.
  Note: This fix resolves the issue for consumer web service descriptors completely. However, to fully resolve the issue for WSDL first provider web service descriptors, you must also install a Web Services Stack fix that includes WSSTACK-2783.

- PIE-51285 [IS_10.2_Core_Fix2, IS_9.12_Core_Fix17]
  The enhanced XML parser throws an ArrayIndexOutOfBoundsException when more than 20 attributes are present on a node.
  With this fix, there is longer a limit to the number of attributes on a node which prevents the ArrayIndexOutOfBoundxException.

- PIE-50185 [IS_10.2_Core_Fix2]
  Integration Server does not close a socket connection after SSL handshake fails.
  If Integration Server establishes the socket connection using JSSE and then if the SSL handshake fails, the socket connection remains open.
  The issue is now resolved. Integration Server now closes the socket connection after SSL handshake fails.

- PIE-51486 [IS_10.2_Core_Fix2]
  When entering quiesce mode Integration Server shuts down all cache managers available in a JVM instead of just the cache managers started by Integration Server.
  This issue is now resolved.

- PIE-51047 [IS_9.10_Core_Fix14, IS_9.12_Core_Fix17, IS_9.7_Core_Fix24, IS_9.8_Core_Fix21, IS_9.9_Core_Fix25]
  When editing existing keystore and truststore aliases, the configured “Type” and “Provider Type” fields show incorrect input.
  When editing existing keystore and truststore aliases properties, the values
  or the “Type” and “Provider” fields are reset to “JKS” and “SUN” instead of the configured values

set during the creation of keystore and truststore aliases.
The issue is resolved. Integration Server now preserves the settings used during the creation of keystore and truststore aliases.

- PIE-51909 [IS_9.10_Core_Fix14, IS_9.12_Core_Fix17, IS_9.7_Core_Fix24, IS_9.8_Core_Fix21, IS_9.9_Core_Fix25]
  Long latencies observed when using SFTP.
  Concurrent execution of the pub.client.sftp:login service and/or methods in the underlying class com.wm.app.b2b.server.sftp.client.SFTPSessionManager may result in long latencies which are caused by synchronization.
  To address this issue and improve multi-threading support, synchronization statements deemed unnecessary were removed.

- PIE-49804 [IS_9.10_Core_Fix14, IS_9.12_Core_Fix17, IS_9.9_Core_Fix25]
  Enhancements in web service connectors to process messages in an order.
  When creating a consumer web service descriptor from a WSDL document, Integration Server creates a web service connector for each operation contained in the WSDL document. These connectors do not provide a facility to process messages in an order.
  The issue is now resolved. Integration Server now uses message numbers that are unique identifiers used by the reliable messaging sequence.

- PIE-51892 [IS_9.10_Core_Fix14, IS_9.12_Core_Fix17, IS_9.9_Core_Fix25]
  FTP thread hangs during FTP put operation.
  During an FTP file transfer using the FTP put command from client to server, where Integration Server acts as the server, if a network issue occurs, then the FTP thread hangs and it never times out. This issue happens only in Passive mode.
  The issue is now resolved.

- PIE-51663 [IS_9.10_Core_Fix14, IS_9.12_Core_Fix17]
  Integration Server acting as FTPS client cannot connect to an FTPS server.
  An Integration Server acting as FTPS client cannot connect to an FTPS server using JSSE.
  Integration Server uses JSSE when the secure/useJSSE input parameter for the pub.client.ftp* service is set to yes.
  The issue is now resolved.

- PIE-52091 [IS_9.10_Core_Fix14, IS_9.8_Core_Fix21, IS_9.9_Core_Fix26]
  When watt.net.http401.throwException is false and the pub.client:http service does not get a response from the server to which it is trying to connect, Integration Server throws a NullPointerException instead of a ServiceException.
  This issue is now resolved.

- PIE-51877 [IS_9.10_Core_Fix14, IS_9.9_Core_Fix25]
  Integration Server throws a ConcurrentModificationException when initializing a connection from a JDBC connection pool.
  This issue is resolved by updating methods in an internal class to use locking.

- PIE-49739 [IS_9.10_Core_Fix14]

Public service setKeyAndChainFromBytes does not work properly if the SSL key differs from the default SSL key in Integration Server Administrator.
If an SSL key is configured in the Integration Server Administrator (Security > Certificates) is different from the one set using the setKeyAndChainFromBytes service, then Integration Server does not send the SSL key set by setKeyAndChainFromBytes service for connections using JSSE. Normally, SSL key set using the setKeyAndChainFromBytes service takes precedence over the one set in the Integration Server Administrator.
This issue is resolved now.

- PIE-50123 [IS_9.10_Core_Fix15, IS_9.12_Core_Fix17]
  ODATA $filter query fails when there is an ampersand (&) in the search string.
  When using the Integration Server OData implementation to query the OData entity with a filter containing the ampersand character (&). For example: $filter=SAMPLE eq 'Text1 & Text2'), Integration Server returns an incorrect result.
  Integration Server now returns a proper result; however, the query string that uses & must be properly encoded before Integration Server receives the query. For example: $filter=SAMPLE%20eq%20%27Text1%20%26%20Text2%27

- PIE-51631 [IS_9.10_Core_Fix15, IS_9.7_Core_Fix25]
  The pub.document:searchDocuments service behaves differently than expected.
  The pub.document:searchDocuments service functions differently from the expected behavior, which is documented in the webMethods Integration Server Built-In Services Reference.
  This issue is now resolved.

- PIE-51101 [IS_9.12_Core_Fix16, IS_9.12_Core_Fix17]
  No option to set custom truststore while invoking pub.client:http service.
  The pub.client:http service uses the default truststore alias (DEFAULT_IS_TRUSTSTORE) while establishing a secure connection with HTTP server. Integration Server does not provide the option to set custom truststore while invoking this service.
  This issue is resolved. Now, a new input parameter "trustStore" is included in pub.client:http service to set the custom truststore alias while invoking this service.

- PIE-49598 [IS_9.12_Core_Fix17]
  Integration Server does not validate the SSL certificate expiry while using JSSE.
  Integration Server does not validate the SSL certificate expiry in a certificate chain of an incoming connection request from an Internet resource while it is configured to use JSSE. Hence, even if the certificate has expired, Integration Server does not reject the request.
  This issue is now resolved.

- PIE-50404 [IS_9.12_Core_Fix17]
  Input parameter value of scheduled task is overwritten with the latest value of input parameter in flow service.
  An assigned input value in scheduler task is overwritten with the latest value set for the corresponding flow service. This impacts the value of the input parameter in the scheduled task.
  This issue is now resolved.

- PIE-51293 [IS_9.12_Core_Fix17]

Execution of a web service connector ends with a NullPointerException.
Execution of a web service connector for a consumer web service descriptor generated from a WSDL document containing multiple service definitions ends with a NullPointerException. This occurs because Integration Server did not create the BRANCH steps correctly when creating the connector.
Now, Integration Server generates the correct BRANCH steps for web service connectors created from a WSDL document with multiple service definitions.
Note: After applying this fix, you must recreate or refresh the consumer web service descriptor to generate web service connectors with the correct BRANCH steps.

- PIE-51300 [IS_9.12_Core_Fix17]
  webMethods messaging trigger ends with the exception com.google.protobuf.InvalidProtocolBufferException: Protocol message contained an invalid tag (zero).
  If a publishable document type specifies an encoding type of Protocol buffers and saving changes to the document type results in warning messages about field names not being valid for protocol buffer encoding, trigger processing fails for instances of the publishable document type with the following exception:
  com.wm.app.b2b.server.dispatcher.exceptions.MessagingCoderException: java.io.IOException: com.google.protobuf.InvalidProtocolBufferException: Protocol message contained an invalid tag (zero)
  This issue occurs because of an error in how Integration Server encodes the UnknownFieldSet when publishing a document. The UnknownFieldSet contains fields that cannot be represented in a protocol buffer message descriptor.
  This issue is resolved by changing the way in which Integration Server encodes the UnknownFieldSet at the time of publication. This encoding change affects only those documents published after the fix is installed. Existing, unprocessed documents that use the improperly encoded UnknownFieldSet must be removed to prevent the occurrence of InvalidProtocolBufferException: Protocol message contained an invalid tag (zero). Make sure the client side queue for the Universal Messaging connection alias that publishes instances of the document type is empty. Additionally, make sure to use Universal Messaging Enterprise Manager to delete the channel associated with the publishable document type. Deleting the channel removes any instance documents encoded with the invalid zero key. Integration Server recreates the channel when it is restarted after fix installation.

- PIE-51613 [IS_9.12_Core_Fix17]
  Integration Server does not start if the distributed service results cache contains invalid entries.
  If the distributed service results cache contains invalid entries, Integration Server exits with an error similar to the following.
  Error occurred while initializing server: java.lang.StringIndexOutOfBoundsException: String index out of range: -1 at java.lang.String.substring(String.java:1967) at com.wm.app.b2b.server.ostore.ServiceCacheImpl.getKeySvc(ServiceCacheImpl.java:350) at com.wm.app.b2b.server.ostore.ServerCache.load(ServerCache.java:110) com.wm.app.b2b.server.CacheManager.startSweeper(CacheManager.java:128) at com.wm.app.b2b.server.Server.run(Server.java:644)
  This issue is now resolved.

- PIE-52025 [IS_9.12_Core_Fix17]
  Unable to restart Integration Server in a clustered environment.
  After installing a fix that contains PIE-45163, restarting Integration Server in a clustered environment fails if there is scheduled task with target set to 'All'. Also, the pub.scheduler:getTaskInfo service returns incorrect status of a running task.
  These issues are now resolved.

- PIE-50807 [IS_9.7_Core_Fix24]
  Integration Server fails to read the restore file.
  Integration Server throws an exception while restoring pipeline from file using the pub.flow:restorePipelineFromFile service. The issue has occurred because an incorrect data type was defined to one of the input fields in the file while saving pipeline.
  This issue is now resolved.

- PIE-50102
  Integration Server fails to update the REST API descriptor(RAD) containing REST V2 resources.
  In Designer, while making changes to the REST V2 resources or the services associated with the operations of the resources, Integration Server fails to update the RAD that contains the REST V2 resources.
  The issue is now resolved.

- PIE-50180
  Decoding a dynamic parameter with unicode characters fails to decode accurately.
  Integration Server fails to correctly decode the unicode characters in the dynamic parameter of a REST request.
  This issue is now resolved.

- PIE-50348
  Integration Server fails to add comments to the generated document.
  When creating a REST API descriptor from a Swagger document, if a definition in the Swagger document contains comments, Integration Server fails to add those comments to the corresponding Comments tab of generated docTypes.
  This issue is now resolved.

- PIE-50352
  In a REST API descriptor, variables of type File get converted to type String.
  When creating a REST API descriptor from a Swagger document, Integration Server represents a file type as a String instead of a document.
  This issue is resolved. Since Integration Server does not support the File type, variables of type File are converted to Object type.

- PIE-50359
  Integration Server clears the response code from a REST API descriptor.
  When adding a new response code to a resource in REST API descriptor and then moving the resource from one package to another, Integration Server removes all the new response codes from REST API descriptor.

This issue is now resolved.

- PIE-50616
  Unable to create a REST API Descriptor for REST resources with documents with cyclic references.
  Creating a REST API descriptor for REST resources with services that have documents with cyclic references fail.
  This issue is now resolved.

- PIE-50639
  Clicking the Log Off link in Integration Server Administrator causes Internet Explorer to log the user out of all Integration Servers in all Internet Explorer tabs.
  This issue is resolved. Clicking the Log Off link in an Internet Explorer tab now logs the user out of the Integration Server in the current tab only. If the user is logged in to the same Integration Server through multiple tabs, logging off the Integration Server on one tab logs the user off of that same Integration Server on the other tabs as well.

- PIE-50642
  Integration Server displays '405 method not found' error while executing a URL at run time that is added to a REST API descriptor(RAD).
  While adding two similar URLs to a RAD with different HTTP methods, Integration Server might display '405 method not found' error while executing one of the
  URLs during run time.
  The issue is now resolved.

- PIE-50780
  The pub.client:http service ends with a NullPointerException if useJSSE is set to yes and the default truststore is not configured.
  If the default truststore is not configured, the watt.security.cert.wmChainVerifier.trustByDefault parameter is set to false, the pub.client:http service sets useJSSE to yes, the pub.client:https service ends with a NullPointerException but should not.
  This issue is now resolved.

- PIE-51052
  When migrating JDBC connection pools, Integration Server migration utility reports the following error. "Error message: String index out of range: -1"
  This issue occurs because of the non-XML files present in the <IntegrationServer_installDirectory>\instances\<instance_name>\config\jdbc\pool folder.
  This issue is now resolved. Integration Server migration utility correctly migrates JDBC connection pools.

- PIE-51268
  Integration Server displays incorrect error messages when certain services belonging to pub.string folder are invoked without the required inputs.
  Integration Server throws incorrect error messages when the pub.string:tokenize , pub.string:numericFormat ,pub.string:padLeft , pub.string:padRight services are invoked.

- When invoked without the required inputs com.wm.app.b2b.server.ServiceException:

java.lang.NullPointerException is thrown and com.wm.app.b2b.server.ServiceException: java.lang.NumberFormatException is thrown when invoked with invalid inputs
This issue is now resolved. Now, Integration Server displays the following appropriate error messages:
com.wm.app.b2b.server.ServiceException: [ISS.0086.9249] Missing Parameter for incomplete inputs
com.wm.app.b2b.server.ServiceException: [ISS.0086.9250] Parameter is not of type for invalid inputs

- PIE-51641
  Package deployment fails with a NullPointerException when the Compile Java Services option is set to Yes.
  Deploying a package that contains a Java service fails with a NullPointerException similar to the following when the Compile Java Services option is set to Yes:
  java.lang.NullPointerException: null
  at com.wm.lang.ns.NSService.setValues(NSService.java:717)
  at com.wm.lang.ns.NSNode.<init>(NSNode.java:93)
  at com.wm.lang.ns.NSService.<init>(NSService.java:149)
  at com.wm.app.b2b.server.BaseService.<init>(BaseService.java:201)
  This issue is now resolved.

- PIE-51723
  Enterprise Gateway blocks the IP address of the proxy used.
  When a proxy is used between a client and Enterprise Gateway, then Enterprise Gateway blocks the IP address of the proxy. This is because, Enterprise Gateway ignores the X-Forwarded-For header in the request and hence during a Denial of Service attack check, instead of using the client IP address it uses the IP address of the proxy server.
  This issue is resolved. Now, Integration Server checks for X-Forwarded-For header in request and if it is present, then Enterprise Gateway uses the client IP address from this header.

- PIE-52012
  Invoking pub.string:numericFormat service with a decimal value for the 'num' parameter fails with error.
  Invoking the pub.string:numericFormat service by providing a decimal value for the input parameter 'num' fails with following error: [ISS.0086.9250] Parameter [num] is not of type: [Integer]
  This issue is now resolved.

- PIE-52017
  Integration Server supports string constraints in Swagger document.
  Integration Server supports to add string field constraints such as MinLength, MaxLength, and Pattern to the Swagger document generated for REST V2 resources. Integration Server generates the constraints along with the Swagger document that are set as part of the fields in the referred definition or service.
  This is a new feature.

- PIE-52089
  Integration Server rolls back the messages to Universal Messaging if message processing fails with

a decoding error.
While processing a message using webMethods messaging trigger that receives documents from Universal Messaging, if Integration Server encounters any decoding errors, Integration Server rolls back the message to the message provider (in this case, Universal Messaging). As the decoding errors are non-transient, the same message rolls back each time Integration Server starts processing that message.
This issue is now resolved. If a webMethods messaging trigger encounters a decoding exception while processing a message, Integration Server acknowledges the message to messaging provider. The message text for [ISS.0153.0089C] has also changed to "Trigger <trigger_name> failed because of message decoding exception: <exception_message>. Message has been acknowledged to messaging provider."

- PIE-52220
  Passing an empty string as an input in pub.string:tokenize service fails with an error.
  In the pub.string:tokenize service when an empty string is passed as an input for the "inString" parameter, Integration Server throws the following error:
  com.wm.app.b2b.server.ServiceException: [ISS.0086.9272] inString cannot be empty when empty string is passed as an input for parameter "inString".
  This issue is now resolved. Now, pub.string:tokenize service accept emptry string as an input parameter.

- PIE-52300
  After applying a fix that includes PIE-36402 or migrating to a version of Integration Server that includes PIE-36402, execution of a web services that use multi-level document type references exhibits a performance degradation in Integration Server as compared to older releases.
  Web services that contain multiple levels of document type references (in which a document reference field includes a field that is a document reference field which in turn also includes a field that is a document reference field, and so forth) exhibit a decrease in performance. This issue is observed after applying a fix containing PIE-36402 (IS_9.8_Core_Fix2) on Integration Sever 9.8 or in Integration Server versions 9.9 and higher.
  The performance degradation has been remedied by identifying and removing performance bottlenecks.

- PIE-52360
  An enabled webMethods messaging trigger does not have a registered listener on Universal Messaging but the trigger cannot be restarted.
  In some situations, a webMethods messaging trigger that receives messages from Universal Messaging does not create the associated listener on the Universal Messaging server as part of trigger startup. Without the listener, the trigger will not receive messages from Universal Messaging. Even though there is not a listener for the trigger, the trigger remains in an enabled state which prevents Integration Server from restarting the trigger.
  This issue is observed in the following scenarios:
  - During startup of the webMethods messaging trigger, the connection to Universal Messaging becomes unavailable but is quickly re-established. However, the listener for the trigger is not created on the Universal Messaging server. In Integration Server Administrator, the trigger state indicates that it is enabled but not connected (On the Settings > Messaging > webMethods

Messaging Trigger Management screen, the Active property displays Yes (Not Connected) for the trigger.) In this state, the trigger cannot be re-enabled manually. The Universal Messaging connection alias used by the trigger must be restarted for the trigger to be enabled. This scenario is most frequently observed when Integration Server connects to an active-active Universal Messaging cluster.
- During startup of the webMethods messaging trigger, Universal Messaging throws an exception while creating the listener for the trigger but the Universal Messaging connection alias remains active. This can occur for a variety of reasons, but it most often occurs when the trigger request to create the listener times out or when the Universal Messaging connection alias and the trigger are connecting to different realms.
To resolve the scenarios in which a listener is not created on Universal Messaging as part of startup of the webMethods messaging trigger, Integration Server now attempts to restart any webMethods messaging trigger that is enabled, associated with an active Universal Messaging connection alias, and does not have a registered listener on Universal Messaging server.

- PIEAR-1086
Assertion errors related to Adapter Connections are logged when releasing the connection or making the connection available. This happens when the parent service has set a timeout in the flow services.
This issue is now resolved.

- PIEAR-1093
Error copying an existing connection as the default values for the connection properties are not copied.
The issue is now resolved.

Release 10.2

- PIEAR-1052
In webMethods Adapter for SAP, the connection is not enabled, as the script file was not loaded. The issue is resolved. Now the connection is enabled and ART retains the backward compatibility of the adapters which are referring to the java scripts with the extension .js.txt.

- PIEAR-1056
In webMethods Adapter for Runtime, an error occurs when you view the listeners in Integration Server Administrator.
This issue occurs when the listener is not loaded in the Integration Server which is referred by one of the notifications.
This issue is now resolved.

Release 10.1

- PIE-45041
The Integration Server server log contains the active Session ID even though masking of Session IDs is enabled.

The Integration Server server log contains the active Session ID when the maskSession ID is set to “true” and logging level is set to “Trace”. The active Session ID is mentioned even though masking of Session IDs is enabled.
This issue is now resolved.

- PIE-45064
  Moving or renaming folders containing flow services results in broken references.
  When a folder containing a flow service is moved or renamed, any document references present in the flow service are not correctly updated. This results in the flow service containing broken references.
  The issue is now resolved.

- PIE-45087
  Designer does not honor the protocol buffer encoding type for a publishable document type created in an earlier version of Integration Server.
  When using Designer 9.10 or higher to view or edit a publishable document type created on an Integration Server version 9.10 or earlier, Designer may display the IData as the Encoding type even though the assigned encoding type is protocol buffers. Additionally, if the Encoding type is changed to Protocol buffers, Designer reverts the Encoding type to IData upon save.
  This issue is now resolved.

- PIE-45125
  Java services that reference Ehcache classes directly cannot be compiled.
  This issue is resolved.

- PIE-45131
  Issues with configuring a restV2 resource.
  When configuring a restV2 resource, Integration Server does not display any error on specifying a URL format containing only dynamic parameters for a REST resource. This is incorrect because a URL format must include at least one static parameter.
  Integration Server throws an exception on specifying a URL format with only dynamic parameters.

- PIE-45172
  From Command Central, Integration Server instance creation using a YAML template and specifying the dbtype in lower case prevents any update to Audit Logging Destination.
  When using Command Central to create an Integration Server instance using a YAML template and specifying the dbtype in lowercase prevents any attempt to change the Audit Logging Destination to 'Database' for an Audit Logger in the Edit Session Logger Details page (Settings> Logging> Edit Session Logger Details). Additionally, the following error occurs on changing the Session Logger: RESTART is required due to inability to update in Integration Server.
  This issue is resolved. Now, the dbtype is case insensitive in the YAML template, allowing the choice of "Database" as the destination when the ISCoreAudit Functional Pool is set to a valid JDBC pool.

- PIE-45177
  When using Command Central to create an Integration Server instance using a YAML template, specifying the dbtype in lower case causes the functional description to be unavailable.

When creating an Integration Server instance with Command Central, using YAML template and specifying the dbtype in lowercase causes the functional description of functional alias to be unavailable for viewing or editing on the Settings >JDBC Pools > Functional Definitions page.
This issue is resolved. Now, the dbtype is case insensitive in the YAML template and hence, the function description appears correctly.

- PIE-45320
  Unable to add IPv6 address in Integration Server to access a port.
  While updating the IP access of a port in the Integration Server Administrator on the Security > Ports > IP Access screen, Integration Server does not allow a user to add an Ipv6 address in the allow/deny lists of IP address.
  This issue is now resolved. Integration Server now allows an IPv6 address in the allow/deny lists.

- PIE-45366
  Searching for Integration Server elements results in a NullPointerException.
  Searching for Integration Server elements using the Service Development perspective of Designer in a package for which the manifest.v3 file has the system_package attribute missing results in a NullPointerException.
  This issue is now resolved.

- PIE-45543
  Executing the is_container script with the createLeanDockerfile option creates a Docker image that does not include the WmCloud package.
  This issue is now resolved.

- PIE-45588
  When using a stand-alone Java client created for Integration Server, the client ignores client certificates when establishing an outbound HTTP connection using JSSE.
  When a stand-alone Java client uses the Integration Server method com.wm.app.b2b.client.BaseContext.setSSLCertficates() to provide client certificates for the purpose of establishing an SSL connection using JSSE, the client did not provide the destination HTTP server with the client certificates needed to establish the connection.
  This issue is now resolved.

- PIE-45797
  Java service compilation fails with an error if the service uses the Common Directory Services API.
  If a Java service uses the Common Directory Service API, compilation of the Java service fails with the following error:
  class file for com.webmethods.sc.LocalizedException not found
  This issue is now resolved.

- PIE-46024
  Integration Server now supports migration when source and target version are same.
  The migration utility can now be used for migration where the source and target versions of Integration Server are the same. This can be useful for data center or machine moves.

- PIE-46407

Unable to change the logging level for Integration Server by using composite templates in Command Central.
When using a composite template in Command Central to change the logging level for an Integration Server, Command Central throws a ConfigurationValidationException and the logging level does not change.
This issue is now resolved.

- PIE-46654
  The Integration Server migration utility does not migrate the quiesce port settings.
  This issue is now resolved.

- PIEAR-1002
  In webMethods Adapter Runtime 10.1, when you create an adapter connection, the Pool Increment Size value in connection management property cannot be greater than Maximum Pool Size value.
  This issue is resolved. Pool Increment Size is now validated with Maximum Pool Size value.

- PIEAR-1004
  In webMethods Adapter Runtime 10.1, the connection pool reset does not happen when adapter connection exception occurs in the design time while creating or configuring the adapter services.
  This issue is now resolved.

- PIEAR-1012
  In webMethods Adapter Runtime 10.1, when you navigate to the Connections screen from the About screen, the left navigation pane of the Connections screen still shows the information related to About screen.
  This issue is now resolved.

Release 10.0

- WFF-259
  An exception causes file polling ports to automatically disable.
  When processing large number of files, the file polling port is automatically disabled and the following exception is displayed in Integration Server Administrator.
  "java.lang.IllegalArgumentException: Comparison method violates its general contract!"
  The disabled port can only be enabled by reloading the package.
  This issue occurs because of the change in the sorting algorithm in JDK 1.7.
  This issue is resolved.

- WFF-261
  In webMethods Integration Server, the pub.flatFile:convertToString service eliminates the values or fields that contain only white spaces which results in incorrect outputs.
  This issue is resolved.

Release 9.12

- PIE-39870 (IS_9.0_SP1_Core_Fix14, IS_9.5_SP1_Core_Fix11, IS_9.9_Core_Fix2, IS_9.10_Core_Fix1)

Service fails to run automatically after the cache expires when the Prefetch parameter is set to True. When the Prefetch parameter is set to True and the cache expires, the service fails to run automatically. The following exception is displayed in the server logs. "[ISS.0086.9249] Missing Parameter:num1" .
This issue is resolved.

- PIE-39177 (IS_9.0_SP1_Core_Fix14, IS_9.9_Core_Fix2)
  Integration Server logs contain duplicate entries after a port is enabled and disabled.
  When a port is enabled and disabled, Integration Server creates two entries for each action in the server logs. This issue occurs if the logging level is set to Debug. Integration Server logs the same information in the Debug log and the Info log which causes duplicates.
  This issue is resolved. Now, if the logging level is set to Debug Integration Server logs information related to enabling and disabling of ports in the Info log.

- PIE-36924 (IS_9.5_SP1_Core_Fix11)
  An additional prefix is included in an XML document input when restored from the pipeline.
  Integration Server includes an additional prefix dx: at the root of an XML document instance.
  This issue occurs when the XML document is updated as specified in the following sequence of steps:
  (1) The XML document is provided as input to an operation in a provider web service descriptor.
  (2) The document is saved to a pipeline.
  (3) The pipeline is restored.
  The issue is now resolved.

- PIE-39114 (IS_9.7_Core_Fix8, IS_9.9_Core_Fix2)
  Discrepancy in session count on Server > Statistics page and Server > Statistics > Session page.
  In Integration Server Administrator, the current total sessions displayed on the Server > Statistics page does not match the number of rows for Current Sessions displayed on the Server > Statistics > Sessions page.
  This issue is now resolved.

- PIEAR-746 (IS_9.7_Core_Fix8)
  The threads waiting for the connection and the threads releasing the connection from the connectionPool are blocked in the ConnectionPool.
  When the connection creation thread acquires a lock on the ConnectionPool, the thread goes to the wait state and the state of this thread does not change. This prevents the connection from being released.
  To resolve this issue, Integration Sever now includes the following server configuration parameter: watt.server.jca.connectionPool.createConnection.interrupt.waitTime Specifies the wait time, measured in milliseconds, that elapses before Integration Server interrupts a connection creation thread that is in a wait state. There is no default value. You must restart Integration Sever for changes to this parameter to take effect.

- PIE-39355 (IS_9.7_Core_Fix8)
  Integration Server does not indicate when it is disconnected from a cluster.
  Integration Server does not log an error message when it is disconnected from a cluster, preventing

automatic detection of the situation.
Now, Integration Server logs the following error message when it is disconnected from the cluster.
[ISS.0033.151] The cluster is now not operational.
Additionally, Integration Server logs the following error message when it rejoins the cluster.
[ISS.0033.152] The cluster is now operational.

- PIE-34518 (IS_9.8_Core_Fix5)
  Addition of CDATA block support to outbound SOAP processing.
  When processing an outbound SOAP message, Integration Server ignores CDATA delimiters in String fields.
  If a String field contained text in a CDATA block, Integration Server treats the text as regular text instead of character text and url-encodes special characters in the delimiters and in the text block.
  Integration Server now provides CDATA block support for processing of outbound SOAP messages only when Integration Server hosts the web service provider. When a service used as an operation in a web service provider returns String values containing CDATA blocks, when encoding the IData into a SOAP message, Integration Server places the CDATA text in the outbound SOAP message in a CDATA section and does not url-encode special characters in the delimiters or text block.
  Keep the following information in mind:
  - A CDATA block begins with <![CDATA[and ends with]]> - Multiple CDATA blocks may be used in a single String value.
  - CDATA blocks may not overlap or be nested.
  Note: When Integration Server is acting as the web service client, Integration Server does not provide CDATA block support when processing outbound SOAP messages. If a String value containing the request is passed to the web service connector and the string contains CDATA, the contents of CDATA block will be treated as regular text and special characters in the delimiters and text block will be url-encoded in the outbound SOAP request.

- PIE-35132 (IS_9.8_Core_Fix5)
  Changes to Integration Server to address the security vulnerabilities identified during internal security testing.
  This fix resolves the security issues found during internal security testing.

- PIE-38427 (IS_9.8_Core_Fix5)
  A NullPointerException occurs when modifying or saving a flow service if dependency checking features are disabled.
  A NullPointerException occurs in Software AG Designer when modifying or saving a flow service if the server configuration parameter watt.server.ns.dependencyManager is set to false.
  This issue is resolved.
  PIE-38649 (IS_9.8_Core_Fix5)
  Host name of an email port is displayed incorrectly in the View Email Client Details page in Integration Server Administrator.
  The host name of an email port is incorrect and contains garbled characters when the email port is viewed in the Security > Ports > View Email Client Details page in Integration Server Administrator.
  This issue is resolved. The host name is now displayed correctly in the Ports > View Email Client

Details page in Integration Server Administrator.

- PIE-38920 (IS_9.8_Core_Fix5)
  Certification of Integration Server with MySQL version 5.6.
  Integration Server 9.8 is now certified for use with MySQL version 5.6
  PIE-39009 (IS_9.9_Core_Fix2)
  Messages are rolled back to Universal Messaging incorrectly if the webMethods messaging trigger is configured to Suspend and Retry Later.
  If a webMethods messaging trigger specifies Suspend and Retry Later for On Retry Failure, Integration Server rolls back a message to Universal Messaging if a transient error causes the trigger to fail. However, instead of rolling back a single message, Integration Server rolls back all of the received but unacknowledged messages for the trigger. For concurrent triggers, this can cause messages to be reprocessed. For serial triggers, this can cause messages to be lost.
  Now, when a webMethods messaging trigger specifies Suspend and Retry Later and a transient error prevents the trigger from executing successfully, Integration Server rolls back a single message at a time to Universal Messaging. Additionally, Integration Server clears the trigger queue on the Integration Server immediately after suspending the trigger. Clearing the queue removes any messages that the trigger received but did not process. These messages will be redelivered.

- PIE-39287 (IS_9.9_Core_Fix2)
  Integration Server displays the predefined Client Prefix instead of the user-defined Client Prefix after a restart.
  After restarting Integration Server, the user-defined Client Prefix for a new Broker Connection Alias is replaced with the predefined Client Prefix that Integration Server creates.
  This issue is resolved. Now, Integration Server displays the user-defined Client Prefix correctly after a restart.

- PIE-39295 (IS_9.9_Core_Fix2)
  An attempt to create a web service descriptor form a WSDL document fails with a NullPointerException if the WSDL document contains multiple services but the services are not associated with all of the binding and portType definitions in the WSDL document.
  Creating a consumer web service descriptor or a WSDL first provider web service descriptor from a WSDL document fails with a NullPointerException if the WSDL document defines multiple services and every service is not associated with every binding and portType defined in the WSDL document.
  This issue is resolved. Each web service descriptor generated from a service in a WSDL document exposes only the binding and portTypes associated with the service.

- PIE-40157 (IS_9.10_Core_Fix1)
  Correction to a spelling error in Queue Provider field.
  On the Settings > Logging > Edit <loggerName> Details page, Integration Server Administrator lists Universal Messaging instead of Universal Messaging as one of the Queue Provider values.
  The spelling error is now corrected.

- PIE-37272
  Deadlock among threads in XA recovery store prevents XA transactions from completing.

The XA recovery store contains information about each XA transaction, including the transaction ID (XID), the global state of the transaction at each point in the transaction, and the state of each resources participating in the transaction. During an XA transaction, a deadlock between threads accessing the XA recovery store occurred because an exception occurred after a thread acquired a lock. The exception prevented the release of the lock.
This issue is now resolved. In addition to resolving the issue described above, Integration Server now includes server configuration parameters to control recovery logging.
watt.server.transaction.xastore.performXALogging
Specifies whether Integration Server writes transaction information to the XA recovery store. Set to true to instruct Integration Server to log information about the state and progress of each XA transaction. Set to false to instruct Integration Server to skip logging XA transaction information. The default is true.
Important! If you set watt.server.transaction.xastore.performXALogging to false, Integration Server does not log any information to the XA recovery story while processing a transaction, making transaction recovery impossible. If you want Integration Server to automatically resolve incomplete transactions or you want to manually resolve incomplete transactions, Integration Server must perform XA logging.
You must restart Integration Server for changes to this parameter to take effect.
watt.server.transaction.xastore.maxTxnPerFile
Specifies the maximum number of unique XA transactions in an XA recovery log file. When the XA recovery log file reaches the maximum number of transactions, Integration Server creates a new file. The default is 2000 transactions.
Consider increasing the maximum number of unique XA transactions for the XA recovery log file if there are more than 2000 active XA transactions and Integration Server exhibits a performance delay due to input/output. Increasing the number of unique XA transactions allowed per file decreases the number of files used for the XA recovery log, which, in turn, may result in fewer files for Integration Server to search when performing XA recovery.
Decreasing the number of unique XA transactions stored in file may help during transaction recovery as it might decrease the time to read and consolidate open transactions.

- PIE-37566
  Executing anonymous services creates persistent sessions.
  Some anonymous services in the WmRoot package create persistent sessions. Repeated execution of these services can consume all of Integration Server's licensed sessions.
  This has been fixed. Anonymous services in WmRoot are now stateless which means that Integration Server creates sessions to execute these services and discards the sessions immediately
- PIE-38112
  Integration Server does not restart after improper shut down.
  If Integration Server does not shut down properly, which can be caused by a JVM crash or a machine crash, configuration files such as acls*.cnf become corrupted or are reduced to a size of zero. In turn, this causes a restart of Integration Server to fail. To recover, data must be restored manually from a backup directory. This issue results from how Integration Server persists data into the file system. Prior to this fix, Integration Server tried to overwrite the existing file in the file system. If Integration Server shut down improperly right after the file was recreated but before the new content was written to the file system, the file content became corrupted or the file was

reduced to a size of zero.
With this fix, Integration Server changes how it saves changes to configuration files. When saving changes to configuration files, Integration Server first saves the configuration changes in a temporary file in the IntegrationServer_directory/instances/instanceName/config/work directory. After saving the changes in a temporary file, Integration Server moves the temporary file to the actual configuration file. This ensures that if unexpected behavior occurs before the configuration changes are initially saved to the temporary file, only the temporary file is impacted. The actual configuration file is not corrupted. At start up, if Integration Server detects files in in the IntegrationServer_directory/instances/instanceName/config/work, it suggests that changes to a configuration file were not saved to the temporary file and that, therefore, the changes were not made to the actual configuration file. Use the contents of the IntegrationServer_directory/instances/instanceName/config/work directory to determine which configuration files were in the process of being changed. Decide whether you want to redo the changes to the configuration files. Delete the files under the directory and redo the configuration change if you so choose.

- PIE-38470
  The xsd:any element and attributes are not generated for a web service descriptor when the Allow Unspecified Fields property is true.
  The WSDL document for a web service descriptor does not include the xsd:any element and attributes when the web service descriptor uses a Document variable for which the Allow unspecified fields property is set to true. This issue occurs even when the server configuration parameter watt.core.schema.createSchema.omitXSDAny is set to false.
  To address this issue, the web service descriptor now includes a property, Omit xsd:any from WSDL. When set to false, the WSDL document generated by Integration Server includes an xsd:any element. When set to true, the WSDL document does not include the xsd:any element. The default value is true. For changes to this property to take effect, after saving the change to the property, either refresh the web service descriptor or reload the package that contains the web service descriptor.

- PIE-38658
  When a client sends multiple HTTP PUT requests to Integration Server, every second request fails with a 400 Bad Request response.
  Now, alternating HTTP PUT requests from the same client will not fail. Integration Server handles multiple HTTP PUT request from the same client correctly.

- PIE-38686
  GET request mandatorily adds a default value to the Content-Type header.
  If Content-Type header is not specified in a GET request, the pub.client.http service adds a default Content-Type:"application/x-www-form-urlencoded". According to HTTP 1.1, it is not mandatory to specify the Content-Type header. If the Content-Type header is not specified, the pub.client.http service should not add a default Content-Type.
  Now, if no Content-Type header is specified, pub.client.http service does not add a default Content-Type to the request. However, if you pass the data using 'args' or 'table' keys, default Content-Type: "application/x-www-form-urlencoded" is added.

- PIE-38751
  In Integration Server, sometimes publishing a guaranteed document to Universal Messaging server fails due to transaction failure.
  This issue occurs when the Universal Messaging server does not respond to the transaction within the specified EvenTimeout.
  This issue is resolved. A new server configuration property, watt.server.um.producer.transaction.commitRetryCount is introduced in the server configuration file. This property specifies the number of retry attempts Integration Server makes to publish the message to the Universal Messaging server. The maximum number of retries is 9. If you try to assign a value greater than 9, Integration Server automatically sets the value of the property to 9. The default is 0. When setting a retry value, you must ensure that the total transaction time does not exceed the MaxTransactionTime. The total transaction time is calculated by multiplying the total number of attempts with the EvenTimeout. For example, if the retry value is set to 9, and the EvenTimeout is set to 60s, the total transaction time is 60,000(9+1) = 600s.

- PIE-38932
  Integration Server loads a package more than once during deployment.
  During deployment of multiple packages at the same time, Integration Server reloads the packages more than once.
  Now, when a deployment operation includes multiple packages, Integration Server loads each package only once.

- PIE-39017
  In a clustered environment, Integration Server sometimes does not create child tasks when a new server is added to the cluster or when an existing server is restarted.
  When a task is scheduled to run on all servers in a clustered environment, Integration Server creates a parent task and a child task for each server in the cluster. When a new server is added to the cluster or when an existing server in the cluster is restarted, Integration Server creates a corresponding child task upon server restart. However, Integration Server sometimes does not create the child task for the newly added server or for the server that was restarted. As a result, the complete information for all servers in the cluster is not available on the Scheduler screen.
  This issue is resolved.

- PIE-39164
  SSL handshake fails as no client certificate is sent when a new connection is opened through proxy.
  When Integration Server tries to open a connection to web server that requires client certificate through proxy, the SSL handshake using JSSE fails as no client certificate is sent. This happens only when the keyStoreAlias and keyAlias parameters are set using pub.security.keystore:setKeyAndChain service and not when they are set using the Integration Server Administrator (Security > Certificates). This issue occurred because the client certificate attachment was missing in the request.
  This issue is now resolved. Now, client certificate is sent to web server when a connection is opened through proxy.

- PIE-39178
  Repository-based deployment fails when deploying locally publishable documents on the target

Integration Server.
When deploying locally publishable documents on the target Integration Server using a repository-based deployment, Integration Server cannot find a messaging connection alias for the locally publishable document. This causes the deployment to fail. When the publishable document type is set to locally only, there is no messaging connection alias attached to it.
This issue is resolved. Now, Integration Server does not check for a messaging connection alias during deployment when the publishable document type in Integration Server is set to locally only.

- PIE-39210
  Integration Server experiences memory leak due to orphaned entries of type com.wm.app.b2b.server.ostore.FileCache$InvokeCounter located in com.wm.app.b2b.server.ostore.FileCache.
  This issue is now resolved.

- PIE-39268
  Enhancement to Integration Server to use a file for specifying allowed cipher suites.
  The following server configuration properties identify the list of cipher suites for use with inbound and outbound SSL connections.
  watt.net.ssl.client.cipherSuiteList
  watt.net.ssl.server.cipherSuiteList
  watt.net.jsse.client.enabledCipherSuiteList
  watt.net.jsse.server.enabledCipherSuiteList
  Prior to this enhancement, the properties accepted a comma-separated list of cipher suites. However, specifying a long list of cipher suites can be cumbersome. To make it easier to specify a long list of cipher suites, Integration Server now allows specifying a file as the value for the cipher suite server configuration properties. In the file, specify each cipher suite on a different line. For each cipher suite server configuration property for which you want to specify a file instead of a list of cipher suites, specify the following as the value of the property: file:directoryName\filename For example: watt.net.jsse.server.enabledCipherSuiteList=file:c:\ssl\ciphers.txt.
  Note: You can set the value of a cipher suite server configuration property to a comma-separated list, default, or the absolute path to a file. You cannot specify a combination of these. Integration Server loads the file and its list of supported cipher suites at start up. Changes to the contents of the file that are made after Integration Server starts will not take effect until the next time Integration Server starts. Integration Server throws the following exception if it cannot find the specified file: ISS.0070.9048E Integration Server cannot find the file {0} specified for the server configuration parameter {1}.
  Cause The specified server configuration parameter uses a file to identify the allowed cipher suites; however, Integration Server cannot find the file.
  Action Make sure the server configuration parameter specifies the correct location of an existing file.

- PIE-39278
  Message not found for messageKey 68.29 and 68.30.
  If an email port has "Log out after each mail check=Yes" , Integration Server fails to retrieve the error messages for messageKey 68.28 and messageKey 68.30.
  Following messages appear in the server log:

For IMAP: ISP.0068.0029I] Message not found for messageKey 68.29. For POP3: ISP.0068.0029I] Message not found for messageKey 68.30.
This issue is now resolved. Now, correct messages are displayed for messageKey 68.28 and messageKey 68.30.

- PIE-39298
  Error in getting SFTP server host key, if a key exchange algorithm is not supported by SFTP server. Jsch has a default key exchange algorithm order.
  During the handshake, jsch checks the client key exchange algorithms one by one with SFTP server key-exchange algorithms. The first matching algorithm is used as the key-exchange algorithm between SFTP client and SFTP server. However, Jsch does not support 2048-bit keys for diffie-hellman-group-exchange-sha256 and diffie-hellman-group-exchange-sha1 key exchange algorithms in Java 1.7 and earlier versions. Consequently, if the Integration Server runs with Java 1.7 or earlier version and SFTP server expects 2048-bit keys for these algorithms, then the handshake between SFTP server and SFTP client fails.
  To avoid this issue, the order of these key exchange can be changed using the watt.ssh.jsch.kex parameter so that, any other matching key exchange algorithm can be selected as the key exchange algorithm between the SFTP client and SFTP server.
  To address this issue, Integration Server now include a server configuration parameter to change the order of the key exchange algorithm.
  watt.ssh.jsch.kex Specifies the order of the key exchange algorithm for Jsch. The specified order overrides the default key exchange algorithms order supported by Jsch.
  For example: watt.ssh.jsch.kex=diffie-hellman-group-exchange-sha1, diffie-hellman-group1-sha1,diffie-hellman-group14-sha1 If the SFTP server expects 2048-bit keys for diffie-hellman-group-exchange-sha1, then the order of this algorithm can be changed so that, any other matching algorithm can be selected as key exchange algorithm between the SFTP client and SFTP server.
  watt.ssh.jsch.kex=diffie-hellman-group1-sha1, diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1 Integration Server must be restarted for changes to this parameter to take effect.

- PIE-39357
  Integration Server fails to retrieve the SFTP Get Host Key during the SFTP Server Alias creation. SFTP Server Alias creation fails with 'Auth Cancel' exception, as Integration Server fails to retrieve the SFTP Get Host Key.
  The issue is now resolved.

- PIE-39380
  Integration Server issues an Access Denied error message during the logon process and logs an authentication exception in the server log.
  Integration Server issues the following error message during the logon process and logs the exception in the server log:
  Access to Integration Server is denied because of the following authentication exception:
  LoginModule "com.wm.app.b2b.server.auth.jaas.X509LoginModule" unavailable.
  The issue is now resolved.

- PIE-39574
  Web service connector does not capture all the Set-Cookie HTTP response headers sent back by the

web service provider.
When a web service connector sends a request to a web service provider that sends back multiple Set-Cookie response headers, the connector captures only one Set-Cookie response header.
This issue is resolved. Now, if the web service provider sends back multiple Set-Cookie HTTP response headers, the connector returns all of the headers in its output.

- PIE-39585
  Missing fields in an IS document type generated from an XML Schema Definition (XSD) that has complex type definitions referring to group elements defined later in the schema.
  When an Integration Server document type is generated from an XSD where the complex type definitions contain references to group elements defined later in the schema, the generated document type does not contain the fields corresponding to such references.
  This issue is now resolved.

- PIE-39701
  Integration Server migration utility reports an exception during migration of Trading Networks package.
  During migration of Integration Server, if the source Integration Server contains a Trading Networks package, the migration utility throws an exception if JDBC pool information was not selected during migration of configuration files and JDBC pool information was specified during installation of Trading Networks.
  Trading Networks data and configuration file Migrator : Unable to locate the database configuration information. File <IntegrationServer_installDirectory>\bin\migrate\..\..\instances\default\config\jdbc\pool\<poolName>.xml does not exist. Where poolName refers to the pool name specified for the Trading Networks functional alias.
  This issue is now resolved. Integration Server migration utility correctly migrates Trading Networks data.

- PIE-39735
  Synchronization of publishable documents with the messaging provider fails when using a startup service to run pub.publish:syncToProvider.
  When the pub.publish:syncToProvider service runs as part of a startup service, Integration Server reloads the webMethods messaging triggers but fails to update the state of the triggers.
  This issue is resolved.

- PIE-39762
  REST API descriptor treats parameter of type byte[] as byte.
  When creating a REST API descriptor, if a selected REST resource contains a service with byte[] in the service signature, the REST API descriptor treats the parameter as a byte instead of byte[].
  This issue is now resolved.

- PIE-39819
  An error occurs when you create or edit a file polling port if the path to the monitoring directory contains a backslash '\'.
  When you create or edit a file polling port and the path to the monitoring directory contains a

backslash '\', the following error message is displayed. "No listener defined with the specified key: FilePollingListener: c: mp" This issue occurs because the Integration Server Administrator uses the wrong encodingType.
This issue is resolved.

- PIE-39825
  A synchronous or asynchronous publish and wait does not receive a reply.
  If a publishable document type is empty, that is, it does not contain any fields besides the _env field, and values are mapped to fields the _env field before publishing the document using pub.publish:publishAndWait, Integration Server does not receive a reply to the published request. This occurs for synchronous and asynchronous publish and wait requests.
  This issue is now resolved.

- PIE-39839
  Universal Messaging does not reuse the session on Integration Server even though watt.server.trigger.reuseSession is set to true.
  When the server configuration property, watt.server.trigger.reuseSession is set to true and Broker is the messaging provider, Broker reuses the session on Integration Server. Whereas, Universal Messaging as the messaging provider does not reuse the session.
  This issue is resolved.

- PIE-39847
  JDBC connection pools are blocked after the database restarts.
  After the database used by a JDBC pool restarts, Integration Server does not release connections from the pool because of an unexpected exception thrown by the JDBC driver.
  Now, Integration Server exception handling prevents the exception thrown by the JDBC driver from blocking the release of connections from the JDBC pool.

- PIE-39848
  After migrating to Integration Server 9.5 or later, Integration Server displays an exception when calling a web service and using MTOM streaming.
  After migrating 2 to Integration Server 9.5 or later, sending a web service request that uses MTOM streaming for which more than one chunk is sent, the following exception occurs: "Exception --> org.apache.axis2.AxisFault: Connection reset by peer: socket write error"
  This issue is now resolved.

- PIE-39853
  Unable to view assigned input values for services in Integration Server Administrator when the server configuration property watt.server.disableXSSFilter=true.
  When creating a scheduled task, assigning input values using the "Assign Inputs" button displays an empty input field even though the assigned value is saved in the input field. Since the XSS filter is disabled by setting the server configuration property watt.server.disableXSSFilter=true, Integration Server in unable to encrypt quotation marks (") in the input values. Therefore, the assigned input values are not displayed in the input field in Integration Server Administrator.
  This issue is resolved. Integration Server is now able to encrypt quotation marks (") in input values when the server configuration property watt.server.disableXSSFilter=true.

- PIE-39876
  Swagger document generated by Integration Server does not represent REST resources with required list parameters correctly.
  If a REST API descriptor contains REST resources that refer to a list parameter, such as String list, Document list, Document Reference list, and Object list, and the list parameter is required, the definitions section of the generated Swagger document does not indicate that the parameter is required.
  This issue is now resolved.

- PIE-39877
  If a REST API descriptor contains a REST resource that refers to a parameter of type String table, the generated Swagger document does not indicate the type of the parameter as "array".
  This issue is now resolved.

- PIE-39938
  Reliable messaging sequences on Integration Server are timed out when the inactivity timeout interval is set to a value of -1.
  This issue is now resolved. The reliable messaging sequences do not get timed out when the inactivity timeout interval is set to -1 in the Integration Server Administrator (Settings > Web Services > Reliable Messaging > Edit Configuration).

- PIE-39967
  Service error statistics counter causes memory leak.
  An internal mechanism used to report the last 60 seconds worth of service failures has a memory leak. The memory leak occurs when a steady stream of exceptions, at least one every 60 seconds, are thrown without interruption.
  To resolve the memory leak, Integration Server clears the service error statistics counter when new entries are added.

- PIE-40014
  Selected Authentication Type does not get populated under HTTP Transport Properties in consumer web service descriptor endpoint alias.
  When creating a consumer web service descriptor endpoint alias, the selected Authentication Type is not populated under HTTP Transport Properties.
  The issue is now resolved.

- PIE-40027
  Integration Server logs the wrong message when it cannot find the encoder "esapi".
  When Integration Server cannot find the encoder "esapi", it logs the following wrong error message:
  [ISC.0072.0023E] Couldnt initialize encoder "{0}".
  Whereas, Integration Server should log the following message:
  ISC.0072.0023E] Failed to initialize the bundle for encoder 'esapi'.
  This issue is resolved now.

- PIE-40048
  In the Service Usage screen, some of the values are displayed as   instead of the actual value.

In the Service Usage screen of Integration Server Administartor, some of the values in the Count and Last Run columns are displayed as   instead of the actual value. This issue occurs because of encoding issues in Integration Server.
This issue is resolved. Now, the values in the Count and Last Run columns are displayed correctly.

- PIE-40109
  The pub.xml:getNextXMLNode service incorrectly returns a NullPointerException when the NodeIterator does not have XML nodes to return.
  This issue is resolved. The pub.xml:getNextXMLNode service returns null when there are no XML nodes to return.

- PIE-40126
  Results window does not show parameter names when using Natural CALC subprogram.
  From the Natural CALC subprogram, the IDL and IS connection are created. When using Integration Server and Adapter 9.9, and running the example CALC flow service in Software Designer from the Service Development perspective, the results window does not show the parameter names. The field name abbreviator (as used for XmlData) truncates field names that start or end with #.
  With this fix, the Field name abbreviator no longer truncates field names that start or end with #.

- PIE-40139
  Integration Server becomes unresponsive due to an OutOfMemoryError caused by lack of PermGen space. A memory leak in the package classloader may result in a java.lang.OutOfMemoryError: PermGen space, which causes Integration Server to become unresponsive.
  The memory leak in the package classloader is now repaired.

- PIE-40141
  The pub.event.routing* services are not working properly in Integration Server .
  The pub.event.routing* services of send, subscribe, and unsubscribe are not working properly. For example, the send service throws the following ServiceException: Could not run 'send' com.wm.app.b2b.server.ServiceException: com.wm.app.b2b.server.ServiceException: Event Routing service not available.
  This issue is resolved now.

- PIE-40183
  Different error messages for primary and regular HTTP ports.
  When Integration Server receives a client request for a web service that does not exist, it returns different error messages depending on whether the request was received through the primary port or an HTTP port.
  Now, Integration Server returns the same error message regardless of the port through which it receives the request.

- PIE-40189
  Missing schema definition for the envelope (_env) field in the WSDL document when a service containing reference to a publishable document type is exposed as a web service.
  When a service containing reference to a publishable document type is exposed as a web service,

the resulting WSDL document does not contain the schema definition for the envelope (_env) field. This issue is now resolved.

- PIE-40194
  A webMethods messaging trigger that receives messages from a Universal Messaging cluster becomes disconnected and does not reconnect.
  When a Universal Messaging connection alias that specifies a cluster of Universal Messaging servers and the network between the Universal Messaging servers fails, one or more webMethods messaging triggers that use the alias might disconnect and then not reconnect. This might happen even if the trigger can successfully connect to one of the Universal Messaging servers but not the other servers in the cluster.
  Now, if a webMethods messaging trigger becomes disconnected from a cluster of Universal Messaging servers, the trigger reconnects only when the cluster has a quorum. In addition to resolving the issue described above, Integration Server now redirects the Universal Messaging client log entries from the default stdout to the Integration Server's log directory, Integration Server_directory/instances/instanceName/logs/umClient.log. The umClient.log file can be used when debugging Universal Messaging issues.

- PIE-40195
  Integration Server does not validate a Kerberos ticket sent by an Internet browser.
  If Integration Server receives a request on a port configured to accept Kerberos tickets, Integration Server returns a 401 status code with a "Negotiate" header challenge to the Internet browser. Upon receiving the Negotiate header challenge, a properly configured Internet browser generates a Kerberos ticket and sends the request, along with the Kerberos ticket, to Integration Server. Integration Server should validate the Kerberos ticket and either allow or deny the client request based on the ACL settings. However, Integration Server could not process the Kerberos ticket because a Kerberos ticket sent by an Internet browser is structurally different from tickets generated by a Java client. Because Integration Server could not process the ticket, Integration Serer could not validate the Kerberos ticket. Instead, Integration Server returned a 401 status code and prompted the client for username and password credentials.
  Integration Server can now process a Kerberos ticket generated by an Internet browser.

- PIE-40202
  JDBC connection pool for ISInternal functional alias is blocked.
  Integration Server does not release connections from the JDBC connection pool used by the ISInternal functional alias because of an unexpected exception thrown by the JDBC driver.
  Now, Integration Server exception handling prevents the exception thrown by the JDBC driver from blocking the release of connections from the JDBC connection pool.

- PIE-40212
  Enhancement for using a public cache, including a distributed cache, for service results caching.
  Previously, Integration Server stored cached service results in the ServiceResults system cache which is part of the SoftwareAG.IS.Services system cache manager.
  With this enhancement, a local or distributed public cache can now be used for service results caching. By using a distributed cache as the service results cache, multiple Integration Servers can access cached service results.

Keep the following information in mind when using a public cache for service results caching:
-The cache must be configured such that elements do not expire once they are placed in the cache. That is, the Eternal check box must be selected for the cache. The duration of cached service results depends on the Cache expire property value for the service and the watt.sever.cache.flushMins server configuration parameter value.
-When using a distributed cache as a service results cache, make sure that all the Integration Servers that share the service results cache have the same packages.
-When returning cached results for a service, by default, Integration server returns a reference to the cached results instead of the actual value of the cached results. If you want Integration Server to return the actual value instead of a reference, make sure that the Copy on Read and Copy on Write check boxes are selected for the public cache.
-If you want to be able to use Integration Server Administrator or the pub.cache.serviceResuls:listCache service to view the cached elements for a service, make sure that the service results cache is searchable and that the cache allows automatic indexing for keys. That is, for the service results cache, the Searchable check box must be selected and the Key check box next to Allow Automatic Indexing must be selected.
-When a public cache used for service results caching is disabled or a public cache manager that contains the cache used for service results caching is shut down, Integration Server does not cache or retrieve service results. Instead of using cached results, Integration Server executes the service.
-When a public cache used for service results caching is enabled or the public cache manager that contains the public cache used for service results is started, Integration Server re-initializes the cache.
For detailed information about creating a public cache, public cache manager, and a distributed cache, see the webMethods Integration Server Administrator's Guide.
Note: In Integration Server Administrator, the Server > Service Usage page displays statistics about service results. For a distributed cache, this statistics corresponds to the current Integration Server instance only. Integration Server Administrator does not provide aggregated statistics for all the Integration Servers using the same distributed cache for service results caching.
To configure Integration Server to use a public cache for service results caching, you specify the name of the cache and cache manager in the server configuration parameters watt.server.serviceResults.cache and watt.server.serviceResults.cacheManger, respectively.
watt.server.serviceResults.cache
Specifies the name of the public cache to use for service results caching. If no value is specified for the watt.server.serviceResults.cache parameter, Integration Server uses the ServiceResults cache in the cache manger specified in the watt.server.serviceResults.cacheManager parameter as the service results cache. If the cache manager in the watt.server.serviceResults.cacheManager parameter does not contain a cache named ServiceResults, Integration Server throws an error at start up and does not cache service results. You must restart Integration Server for changes to this parameter to take effect.
watt.server.serviceResults.cacheManager
Specifies the name of the public cache manager that contains the public cache to use for service results caching. If no value is specified for this parameter, Integration Server uses the SoftwareAG.IS.Services system cache manager as the service results cache manager. If the SoftwareAG.IS.Services system cache does not contain a cache with a name that matches the value of the watt.server.serviceResults cache, Integration Server throws an error at startup and does not

cache service results. You must restart Integration Server for changes to this parameter to take effect.
Note: To use the ServiceResults system cache located in the SoftwareAG.IS.Services system cache manager as the service results cache, do not specify a value for watt.server.serviceResults.cache or watt.server.serviceResults.cacheManager.

- PIE-40246
  The pub.client:soapClient service does not use the private key and certificate chain set using the pub.security.keystore:setKeyAndChain.
  This issue is now resolved.

- PIE-40259
  Using the delete icon to delete an OpenID Provider on the Security > OpenID page does not delete the OpenID Provider.
  In the Security > OpenID page of Integration Server Administrator, clicking the delete icon in the Delete Provider column invokes an unknown service, resulting in the display of the following message:
  [ISC.0072.9001] Unknown service: wm.server.security.openid:deleteProvider
  Furthermore, the OpenID Provider is not deleted.
  Now, clicking the delete icon in the Delete Provider column on the Security > OpenID page results in Integration Server deleting the OpenID Provider.

- PIE-40261
  Use JSSE option is missing on external port if the internal port is set to HTTP.
  On Integration Server Administrator, when you view a port for which the Enterprise Gateway External Port is set to HTTPS and Enterprise Gateway Registration Port is set to HTTP, the "Use JSSE" option does not appear. The "Use JSSE" option appears only if the Enterprise Gateway Registration Port is set to HTTPS.
  This issue is now resolved.

- PIE-40275
  Enhancement to allow the default locale for pub.date* services to be the server locale instead of the client locale.
  When executing a pub.date* service with a locale input parameter for which no locale is specified Integration Server uses the locale from the session used by the client that invoked the service. Integration Server now includes a server configuration parameter that you can use to specify that the default locale for pub.date* services should be the server locale.
  watt.server.session.locale.ignore
  Specifies whether the default locale for the pub.date* services is the server locale or the locale from the session used by the client that invoked the service. When set to true, when executing a pub.date* service for which a locale input parameter value is not specified, Integration Server uses the server locale as the value of the locale parameter. When set to false, when executing a pub.date* service for which a locale input parameter value is not specified, Integration Server uses the locale from the session used by the client that invoked the service. The default is false.
  You do not need to restart Integration Server for changes to this parameter to take effect.

- PIE-40300
  Under heavy load, Integration Server might generate duplicate contextIDs for audit log messages resulting in an AuditLogManagerException.
  Under heavy load, Integration Server might generate duplicate contextIDs for audit log messages, causing Integration Server to write the following AuditLogManagerException to the server.log:
  [ISS.0095.9998] AuditLogManager Exception:
  com.webmethods.sc.auditing.api.logger.WmAuditLoggerException: Null or duplicate context given to logger 'Error Logger'
  Now Integration Server generates unique contextIDs even under heavy load.

- PIE-40324
  Integration Server throws an error when an incoming message contains a pre-populated contextID and audit contextID.
  Integration Server writes the following error to the server.log when an incoming message contains a pre-populated context ID and audit contextID.
  [BAA.0000.0061] This context has already been used in the current context stack.
  Integration Server now handles a message with a pre-populated contextIDs and audit contextID and no longer logs the above error.

- PIE-40362
  Unable to publish a package to multiple subscribers.
  When publishing a package to multiple subscribers, clicking the 'Send Release' button in Integration Server Administrator prompts you to select a subscriber even though subscribers are selected.
  This issue is resolved.

- PIE-40363
  UserInfo service does not receive error information.
  If an OpenID Provider's UserInfo Endpoint returns an error, information about the error should be passed to the userInfoError input variable in the UserInfo service registered for the OpenID Provider. This is not happening.
  This issue is now resolved.
  Note: Previously, the userInfoError input variable was named error.

- PIE-40372
  Reply messages are sometimes not received with asynchronous request-reply using webMethods messaging and Universal Messaging as the messaging provider.
  In an asynchronous request-reply for webMethods messaging, the pub.publish:publishAndWait service is first invoked followed by the pub.publish:waitForReply. When the messaging provider is Universal Messaging, some reply messages are not received if Integration Server receives the reply message before invocation of the pub.publish:waitForReply service.
  This issue is now resolved.

- PIE-40408

Enhancements for managing a service results cache. Integration Server now includes built-in services that you can use to manage a service results cache, including:
pub.cache.serviceResults:resetServerCache resets the entire cache.
pub.cache.serviceResults:resetServiceCache resets the cache for a particular service.
pub.cache.serviceResults:listServiceCache lists the cached results for a particular service.
In addition to the above services, Integration Server Administrator now includes a way to view the cached service results for a particular service. To use Integration Server Administrator to view the cached service results, do the following:
1. Open Integration Server Administrator if it is not already open.
2. In the Packages menu of the Navigation Panel, click Management.
3. In the Packages list, click the package containing the service for which you want to view cached service results.
4. Click Browse Services in packageName.
5. In the Services list, click the name of the service for which you want to view the cached service results.
6. On the Packages > Management > packageName > Services > serviceName page, click List Service Cache. Integration Server displays a list of the cached results for the service. For each cached element, Integration Server displays the key for the cached element, the cached service inputs, and the associated cached service outputs.
Notes:
- You can also access the Packages > Management > packageName > Services > serviceName page in the following way: Under Server, click Service Usage. On the Server > Service Usage page, click the name of the service for which you want to list cached results.
- To return a list of cached result for a service, the services results must be searchable and that the cache allows automatic indexing for keys. That is, for the service results cache, the Searchable check box must be selected and the Key check box next to Allow Automatic Indexing must be selected. Note that exposing the contents of the service results cache may represent a security risk.
- You can view cached service results for a public service results cache only. You cannot view cached service results in the ServiceResults system cache.
Below is more information about each of the new built-in services.
pub.cache.serviceResults:listServiceCache WmPublic.
Returns a list of the cached service results for a particular service.
Input Parameters
serviceName String. Name of the service for which you want to view the cached service results.
Output Parameters elements Document List. Conditional. A document list (IData) containing the cached service results for the specified service. The pub.cache.serviceResults:listServiceCache service returns the elements output only if the service results cache contains cached results for the specified service. For each set of cached service results, the pub.cache.serviceResults:listServiceCache returns the following information:
key String. Optional. Key for the cached element containing the service results.
input. Document. Optional. An IData containing the key/value pairs for the cached service input.
output Document. Optional. An IData containing the key/value pairs for the cached service output
Usage Notes:
- If serviceName specifies a service that does not exist or a service for which Integration Server does not cache service results, Integration Server returns an empty elements output parameter.

- You can use the pub.cache.serviceResults:listServiceCache with a public service results cache only. You cannot use the pub.cache.serviceResults:listServiceCache with the ServiceResults system cache.
- For the pub.cache.serviceResults:listServiceCache service to return results, make sure that the service results cache is searchable and that the cache allows automatic indexing for keys. That is, for the service results cache, the Searchable check box must be selected and the Key check box next to Allow Automatic Indexing must be selected.
- Exposing the contents of the service results cache via the listCacheService may represent a security risk.

pub.cache.serviceResults:resetServerCache

WmPublic. Resets the cache for all services in the service results cache, resulting in the removal of all cached service results for all services from the service results cache.

Input Parameters None.

Output Parameters message String. Message indicating whether or not Integration Server cleared the service results cache successfully.

- "Cache reset successfully" indicates that Integration Server cleared the entire service results cache successfully.
- "Error resetting cache!" indicates that Integration Server did not clear the entire service results cache successfully. Review the sever log and error log to determine why Integration Server did not clear the service results cache successfully.

pub.cache.serviceResults:resetServiceCache

WmPublic. Resets the cache for specific service, resulting in the removal of cached service results for the service.

Input Parameters serviceName String. Fully qualified name of the service in the format folder.subfolder:serviceName for which you want to remove cached service results.

Output Parameters message String. Message indicating whether or not Integration Server cleared the service results cache for the specified service successfully.

- "Cache reset successfully" indicates that Integration Server cleared the cached service results for the specified service successfully.
- "Error resetting cache!" indicates that Integration Server did not clear the cached service results for the specified service successfully. Review the sever log and error log to determine why Integration Server did not clear the service results cache successfully.

Note: To avoid revealing whether or not a service exists, if serviceName specifies a service that does not exist, the message output parameter returns the message "Cache reset successfully"

- PIE-40422

  Integration Server grants access to a user account disabled by the external user manager.

  Integration Server caches information for users who provide valid credentials. If the user later provides invalid credentials, Integration Server denies access but does not remove the cached information about the user. Consequently, if the user later provides valid credentials, Integration Server grants access even if an external user manager disabled the user account.

  Now, when a user provides invalid credentials, Integration Server removes the user information from the cache.

- PIE-40454

  The pub.storage:get service fails in a cluster.

  Integration Server improves the performance of the pub.storage* services by caching the

DATASTORE_ID values from the IS_DATASTORE table. However, in a cluster, a cached DATASTORE_ID value might become stale due to actions taken by other Integration Servers in the cluster. A pub.storage* service fails when it attempts to use a stale DATASTORE_ID.
Now, when a pub.storage* service attempts to use a stale DATASTORE_ID, Integration Server replaces it with the current DATASTORE_ID.

- PIE-40467
  Corrections for cosmetic issues in Integration Server Administrator.
  Integration Server Administrator displays some cosmetic issues, such as empty cells, deeply nested tables, and missing graphics, that are now repaired.
- PIE-40480
  When restarting Integration Server fix after applying a fix, expected changes are not made to the embedded database.
  If the ISInternal functional alias uses the embedded, internal database and application of an Integration Server fix made changes made to the database structure, Integration Server should perform some database updates upon restart of Integration Server. However, this was not happening.
  This issue is now resolved.
- PIE-40504
  A NullPointerException occurs when Integration Server enters quiesce mode and the JMS subsystem has not been initialized.
  This issue is resolved. Integration Server can now enter quiesce mode even if the JMS subsystem has not been initialized.
- PIE-40505
  Memory leak when invoking pub.jms:reply Integration Server did not correctly release JMS Session and MessageProducer objects when invoking the pub.jms:reply service, resulting in a memory leak. The issue is now resolved.
- PIE-40516
  In a cluster of Integration Servers, the scheduled task for which the target node is set to "Any Node" and "Repeat After Completion" executes the same service simultaneously on different cluster nodes when the Terracotta Server Array becomes unavailable.
  This issue is now resolved. Now, in a cluster of Integration Servers, a scheduled task with target node set to "Any Node" and "Repeat After Completion", executes the service only on one of the cluster nodes at a time when the Terracotta Server Array becomes unavailable.
- PIE-40517
  Specifying a value of true for the watt.net.ssl.client.useJSSE server configuration property does not use JSSE for all of the outbound HTTPS connections from Integration Server.
  After applying a fix or upgrading to a release that includes PIE-38599, Integration Server no longer honors the value of watt.net.ssl.client.useJSSE for outbound HTTPS connections. This occurs because PIE-38599 changed the default value of the useJSSE input parameter in the pub.client:http service to “no” from a null value. A value of “yes” or “no” for the useJSSE input parameter overrides the value of the watt.net.ssl.client.useJSSE server configuration property. A null value for the useJSSE input parameter causes the service to defer to the value of the watt.net.ssl.client.useJSSE server configuration property.

Now, the default value of the useJSSE input parameter for the pub.client:http service is null, which allows the watt.net.ssl.client.useJSSE server configuration property to determine the default behavior for outbound HTTPS connections.

- PIE-40548
  Performance degradation observed in Integration Server.
  Integration Server exhibits performance degradation in the GA versions 9.8, 9.9, and 9.10.
  Integration Server versions 9.6 and 9.7 exhibit a similar performance degradation after applying the latest fix.
  The cause of the apparent performance degradation is now resolved.
- PIE-40561
  During the migration of Integration Server from 9.0 to a later release, the migration utility also migrates the server configuration parameters watt.core.schema.validateIncomingXSDUsingXerces and watt.server.wsdl.validateWSDLSchemaUsingXerces.
  The migration utility need not migrate these server configuration parameters during the migration of Integration Server from 9.0 to a later release because the functionalities provided by the two parameters are replaced in releases starting with 9.0.
  - The 'Validate schemas using Xerces' option in the New Web Service Descriptor wizard replaces the functionality of the watt.core.schema.validateIncomingXSDUsingXerces parameter.
  - The value of the 'Validate schemas using Xerces' option is stored as the web service descriptor property 'validateSchemaUsingXerces' (which replaces the functionality of the watt.server.wsdl.validateWSDLSchemaUsingXerces parameter).
  The issue is resolved. The specified server configuration parameters are removed from Integration Server starting with release 9.0.
- PIE-40569
  Integration Server logs a message about JDBC connection pool waiting threads threshold count being exceeding erroneously.
  Integration Server should log the following message only when a thread is waiting for the JDBC connection pool to have an available connection.
  [ISS.0096.0009I] JDBC Connection pool waiting threads threshold exceeded, <number> threads waiting for connection for pool <JDBCPoolName> Integration Server should not log the above message if the JDBC pool is empty or the JDBC driver threw an exception when Integration Server attempts to create a connection. In these cases, there is not actually a waiting thread.
  This issue is now resolved.
- PIE-40576
  SFTP transfers fail due to timestamp parsing failure.
  A change in the Log Timestamp Format (Settings > Logging > View Server Logger Details > Log Timestamp Format) changes the timestamp format returned by pub.client.sftp:ls service. This causes the SFTP transfers to fail due to the timestamp parsing failure.
  To address this issue, Integration Server now includes a server configuration parameter which specifies the timestamp format for SFTP client public services.
  watt.server.sftp.dateStampFmt
  Specifies the format of date to be used in the SFTP client public services.
  To specify the date format to use, you can use any format that is supported by the Java class java.text.SimpleDateFormat. For example, to display the date with the format 08-12-02

14:44:33:1235, specify dd-MM-yy HH:mm:ss:SSSS. The default format for watt.server.sftp.dateStampFmt property is yyyy -MM -dd HH:mm:ss z.
It is not necessary to restart Integration Server after you change the value of this parameter.

- PIE-40578
  Reloading the Packages > Management page in Integration Server Administrator results in a re-execution of the last command on the page.
  The Packages > Management page in Integration Server Administrator encodes the last action as part of the URL. Web browsers remember the last action as part of the history, resulting in the re-execution of the last action when reloading the frame.
  To resolve this issue, for HTML5-compatible browsers, Integration Server alters the browser history so that that the URL for the Packages > Management page does not include the last action as part of the URL. As a result, when reloading the frame, the last action will not be re-executed.
- PIE-40629
  A concurrent webMethods messaging trigger may not utilize all available threads.
  When a concurrent webMethods messaging trigger uses the maximum available threads for message processing, the next request for a thread must wait for a thread to be returned to the thread pool before another message can be processed. Integration Server returns a thread to the thread pool once message processing completes. When a thread becomes available, the waiting request should begin executing. However, instead of the request using a thread as soon as it becomes available, the request waits until the number of threads used by the trigger reaches 0 (zero). This results in periods of time when the trigger is not using all the available threads.
  Note: A trigger uses the maximum available threads when the number of messages being processed by the trigger equals the Max execution threads property value. For example, if Max execution threads is set to 10, the trigger uses the maximum available threads when it is processing 10 messages.
  Now, a concurrent webMethods messaging trigger will not wait for the number of the threads used by the trigger reaches zero before processing a waiting request. Instead, the trigger begins processing the waiting request as soon as a thread becomes available.
- PIE-40664
  The JSSE-enabled HTTPS ports fail to use the updated certificate.
  After adding a new certificate to the truststore and then reloading the truststore using the Integration Server Administrator (Security > Keystore), the JSSE-enabled ports do not use the added certificate.
  This issue is now resolved. The JSSE-enabled HTTPS ports now use the updated certificate after the corresponding truststore is reloaded using Integration Server Administrator.
- PIE-40684
  Integration Server Administrator sometimes displays two scrollbars right next to each other instead of a single scrollbar.
  This issue is now resolved.

- PIE-40687
  Display non-ASCII Unicode characters in the body of a Java service.
  When you persist Java services, Integration Server encodes non-ASCII Unicode characters as ASCII Unicode escape sequences (for example, the Korean character '?' is saved as '\uD55C'). By default,

Integration Server does not decode the escape sequences. Therefore, when Integration Server sends Java service code to Software AG Designer, escape sequences appear in Designer instead of the original Unicode characters.
This fix lets you configure Integration Server to decode the escape sequences. In Integration Server Administrator, go to Extended Settings and add "watt.server.ns.decodeJavaService=true." The change will take effect immediately (no server restart necessary).
Note: After this change, Unicode escape sequences will no longer display in the Java service code in Integration Server. For example, if you use the string "\uD55C" in Java service code, the string will save correctly, but the editor will display it as "?".

- PIE-40688
  Invoking pub.storage:get causes a pub.storage lock to be released immediately.
  When using pub.storage:get to retrieve an item, Integration Server released the lock on the item immediately. However, when using pub.storage:get to retrieve an item, the lock on the item should remain until the lock is removed by a call to pub.storage:unlock, a call to pub.storage:put, or the lock expires.
  This issue is now resolved.

- PIE-40693
  Slow file transfer speed using pub.client.ftp:put.
  While transferring files with large size (around 1 GB) using pub.client.ftp:put service, the file transfer speed is observed to be slow. Hence, it takes longer to complete the file transfer. This issue was due to the buffer limit in the amount of data that can be read and written to the stream (1KB). This issue is now resolved. Now, the buffer size is increased (32 KB) and hence, the transfer time for large files is less and is comparable with the FTP client on Microsoft Windows.

- PIE-40702
  Change to Integration Server to address security vulnerability.
  A certain form of URL, if executed, can cause a redirection that allows for the possibility of credential theft. The form of URL no longer causes a redirection. Instead, the Integration Server user receives an HTTP 404 response.

- PIE-40742
  Package deployment fails because some packages are partially loaded.
  When using Deployer to deploy IS packages, some packages only partially load. This causes deployment to fail.
  This issue is now resolved. Packages are no longer partially loaded.

- PIE-40867
  Modification to Integration Server to control the logging of messages about duplicate universal names for document types.
  Whenever Integration Server loads a document type with an explicit universal name (such as when a package is loaded), Integration Server registers the universal name. If Integration Server loads a document type whose explicit universal name is a duplicate of one that is already registered,

Integration Server logs the following error message:
"[ISC.0081.0002E] Error registering universal name <universal_name>, already registered to <document_type>" filling the IS logs
A user cannot use Designer or Developer to assign the same universal name to a document type. However, if the same WSDL document is used to generate multiple consumer Web service descriptors and/or multiple WSDL first provider Web service descriptors, Integration Server creates multiple document types with the same universal name explicit. When loading a package containing these document types, Integration Server logs the error message mentioned above multiple times which can clutter the log file. To suppress the error message about registering a duplicate universal name, add the following to extended settings on the Settings > Extended > Edit Extended Settings page
watt.server.ns.logErrorsOnRegisteringMultipleDocTypesForAUniversalName=false
After you save your changes, you do need to restart Integration Server for these changes to take effect. If you want Integration Server to resume logging message about registering duplicate universal names, set watt.server.ns.logErrorsOnRegisteringMultipleDocTypesForAUniversalName to true. The default value is true.
Note: Setting watt.server.ns.logErrorsOnRegisteringMultipleDocTypesForAUniversalName to false suppresses the error messages about duplicate universal names only. It does not resolve the duplicate names.

- PIE-40940
  OAuth authentication is not disabled for requests that contain an Auhorization header with a Bearer access token even after setting the watt.server.auth.oauth.accessToken.useHeaderFields parameter to false.
  When watt.server.auth.oauth.accessToken.useHeaderFields is set to false, Integration Server should not perform OAuth authentication if Authorization header field contains an access_token. However, if the Authorization header field of an incoming request contains a Bearer access token and the watt.server.auth.oauth.accessToken.useHeaderFields is set to false. Integration Server writes the following messages to the server log:
  [ISS.0012.0012W] Authentication of user "bearer" failed with exception: [ISS.0010.8044] Integration Server rejected the request to access this resource. The access token is either invalid or expired.
  [ISS.0053.0002C] Access denied for user bearer on port 5555 -> ...
  This issue is now resolved. If watt.server.auth.oauth.accessToken.useHeaderFields is set to false and a request contains an Authorization header field with a Bearer access token, Integration Server disables OAuth authentication for the request.

- PIE-40950
  Failure to close JSSE SSL connections after handling stateless requests may negatively impact performance.
  When Integration Server finishes handling a stateless request received on a port that uses JSSE for SSL, Integration Server does not close the SSL connection. This causes Integration Server to build up stale connections and slows down port allocation, which affects performance.
  This issue is now resolved.

- PIE-40962
  The encode method of IDataXMLCoder class closes the OutputStream after encoding the IData object.
  The encode method of IDataXMLCoder class, encode(java.io.OutputStream os, IData data) encodes the IData object and closes the OutputStream even though the OutputStream is created by the caller.
  This issue is resolved.

- PIE-40984
  Configuration property to determine whether missing SOAP headers result in an error message. Prior to Integration Server version 9.0, when processing a SOAP received by a web service descriptor, Integration Server did not properly validate that all of the SOAP headers were present. As a result, Integration Server did not throw an error when a SOAP response was missing a SOAP header. Note that all headers in a web service descriptor, whether generated from the original WSDL document or added to the web service descriptor, are treated as required. This is an application of the WS-I basic profile rules declaring that all headers in a WSDL be treated as required. Beginning in Integration Server version 9.0, Integration Server properly validates SOAP responses for required headers. If a required header is not present, Integration Server throws the following error: [ISS.0088.9443] One or more required Headers <headerName> are not present in the SOAP response."
  After upgrading to Integration Server 9.0 or higher from an earlier version, web service connectors that previously succeeded when a required header was missing from a SOAP response now fail. While failure when a required header is missing is the correct and documented behavior, Integration Server now provides a configuration property to control whether missing required headers in a SOAP response results in an error. This can be useful in migration situations. When watt.server.SOAP.ignoreMissingResponseHeader is set to true, a SOAP response that does not contain a required header will not result in an error. The default is false, a SOAP response that does not contain a required header results in an error. Integration Server restart is not required for changes to take effect.

- PIE-40985
  SOAP web service fixes for backward compatibility.
  This fix bundles four separate fixes relating to upgrade and migration of web services. The fixes allow backward compatibility of behavior that has since been prohibited through stricter validation in newer releases.
  (1) Allow 7.1.3 web service providers to process requests with mismatched SOAP versions.
  A range of fix levels in 7.1.3 allowed web service providers to process SOAP requests with a SOAP version that did not match the SOAP version that was declared in the WSDL binding. This fix adds a new extended server setting to emulate the old, less strict behavior so that consumers will not have to update their SOAP requests. The setting only affects web service providers that were created in releases older than 8.2.2 and for which the property "Pre-8.2 compatibility mode"=true. The new parameter is watt.server.SOAP.pre82WSD.ignoreVersionMismatch. By default, this parameter is set to false. To use the parameter to emulate the old behavior, go to Integration Server Administrator and add watt.server.SOAP.pre82WSD.ignoreVersionMismatch=true to the extended

settings.
(2) Correctly perform document type-directed decoding of a SOAP message when the corresponding namespace qualified document type field does not have a namespace prefix.
Starting with release 8.2.2, any document type field that contains a namespace must have a namespace prefix in its name in order to be used with web services. In 7.1.x, this prefix was not required. With this fix, document types migrated from 7.1.x that do not meet the 8.2.2 and later IS web service requirements will match the decoded IData representations of SOAP messages as they did in 7.1.x. Specifically, this issue applies in cases where the Decoder needs to choose between converting a simple type value into a string field or into a mixed content DocType with a *body field.
(3) Allow empty or absent *body values to pass validation.
A validation error occurred in some 9.x releases during decoding of mixed content elements that have an enumeration restriction. The error occurred when optional missed content was absent, because an empty value was not listed as an option in the enumeration set for the *body field. This fix adds an extended setting to skip this validation and treat the *body as completely optional.
The new parameter is watt.core.validation.skipAbsentStarBody. By default, this parameter is set to false.
To use the parameter to skip the new validation, go to Integration Server Administrator and add watt.core.validation.skipAbsentStarBody=true to the extended settings. (4) Process one-way SOAP request without a Null Pointer Exception. After receiving a SOAP request, some one-way web service providers return an error containing a Null Pointer Exception. This fix handles the condition that resulted in an error when processing the SOAP request.

- PIE-41066
  When executing the pub.soap.wsrm:createSequence service without specifying the value of the sequenceKey input parameter, Integration Server incorrectly reports the messaging sequence as null.
  This issue is now resolved. When the pub.soap.wsrm:createSequence service is executed without specifying the value of the sequenceKey input parameter, Integration Server returns the server Sequence ID associated with the messaging sequence.

- PIE-41090
  Enhancement to Integration Server to use regular expressions for specifying the URIs for allowed origin servers.
  When CORS is enabled, the watt.server.cors.allowedOrigins server configuration parameter indicates the origin servers from which Integration Server will accept requests. The watt.server.cors.allowedOrigins value can be a comma-separated list identifying the specific origin servers or a "*", meaning any origin server is allowed. A comma-separated list can become long and difficult to maintain.
  To make the list of origin servers easier to maintain, Integration Server now supports the use of regular expressions in the comma-separated list of allowed origin servers. Integration Server treats any value in the comma-separated list that begins with "r:" as a regular expression. Integration Server treats any value that does not begin with "r:" as a simple string. The server configuration parameter uses the Java regular expression syntax, as documented at

https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html. A regular expression value must match the entire value of the Origin header in the HTTP request for it to be considered a match.
Example:
watt.server.cors.allowedOrigins=http://test1.domain.com,r:https?://.*.test2.domain.com:[0-9]+,r:.+\.[a-zA-Z]*-int.domain.com
Integration Server treats the first value, "http://test1.domain.com", as a simple string. If an Origin header contains this value, it will be allowed.
The second value, "r:https?://.*.test2.domain.com:[0-9]+", contains a regular expression. The "r:" is not part of the regular expression. The actual regular expression used to match supplied Origin headers is "https?://.*.test2.domain.com:[0-9]+".
The third value, "r:.+\.[a-zA-Z]*-int.domain.com", contains a regular expression. The "r:" is not part of the regular expression. The actual regular expression used to match supplied Origin headers is ".+\.[a-zA-Z]*-int.domain.com".
"Origin: http://test1.domain.com" will be allowed because it is equal to the first value.
"Origin: http://my.test2.domain.com:8080" will be allowed because it matches the second value.
"Origin: https://my.test2.domain.com:8088" will be allowed because it matches the second value.
"Origin: http://my.test2.domain.com" will not be allowed. If it had a port number, it would match the second value.
"Origin: nbps://example.prod-int.domain.com" will be allowed because it matches the third value.
"Origin: example.qa.staging-int.domain.com" will be allowed because it matches the third value.
"Origin: example.dev1-int.domain.com" will not be allowed. If the second token of the host name did not include any digits, it would have matched the third value.
watt.server.cors.allowedOrigins=*
The simple wildcard "*" continues to work as before. It indicates that any origin server will be allowed.
Regular expressions that match any host name, IP address and port (e.g. "r:.+" and "r:.*") will have the same effect as "*".
Note that when CORS is enabled, Integration Server evaluates the list of regular expressions in watt.server.cors.allowedOrigins sequentially for every request. Integration Server performs a regular expression match operation on each regular expression until a match is found or all regular expressions in the list have been evaluated. Software AG recommends that you put the more frequently matched regular expressions at the beginning of the comma-separated list.

- PIE-41109
  Swagger document and REST Resources tab for REST API descriptor does not represent service input correctly when the service signature includes one or more Document, Document List, Document Reference, or Document Reference List field and another type of field.
  The Swagger document and REST Resources tab for a REST API descriptor that includes a REST resource (service) whose service signature includes one or more Document, Document List, Document Reference, or Document Reference List fields in addition to another type of field does not correctly represent the service signature. Integration Server places all of the input parameters for that resource into a REST Definition and sets the source to Body which is incorrect. Integration Server should place the Document, Document List, Document Reference, or Document Reference List fields in the REST Operation as Body parameters that reference the REST Definitions and place

the parameters that are not documents in the REST Operation as Header parameters.
This issue is now resolved.

- PIE-41121
  Integration Server does not show reliable messaging sequence reports in certain situations.
  When Integration Server exchanges messages with a reliable messaging client where the web service operation uses In-Only Message Exchange Pattern (MEP), the reliable messaging sequence reports are not displayed on Integration Server.
  The issue is now resolved. The reliable messaging sequence reports specifying the Server Sequence ID are displayed on Integration Server in situations where the web service operation uses In-Only MEP.

- PIE-41147
  Integration Server Administrator returns a blank page when you run the wm.server.query:getSettings service from the WmRoot package.
  This issue is resolved.

- PIE-41152
  When watt.net.http401.throwException is set to false and the pub.client:http service receives a 401 error, the service does not include the body of the response in the output.
  If the watt.net.http401.throwException server configuration property is set to false, when the pub.client:http service receives a 401 error response, the service places the HTTP response header and body in the output pipeline. However, the pub.client:http service was not placing the response body in the output pipeline.
  This issue is now resolved.

- PIE-41155
  Integration Server fails to act as an FTP server. An attempt by an FTP client to connect to an FTP port on Integration Server fails with a NullPointerException.
  This issue is now resolved.

- PIE-41176
  Web service fails with a RampartException while handling a holder-of-key SAML 2.0 assertion.
  When handling a holder-of-key SAML 2.0 assertion, a web service fails with the following exception: RampartException: No crypto property file supplied for decryption.
  This issue is now resolved.

- PIE-41281
  Integration Server Administrator displays incorrect values for fields in Security > Ports.
  Integration Server Administrator displays incorrect values (default values) for "Use JSSE" and "Client Authentication" fields in the Security Configurations section of Security > Ports.
  This issue is now resolved.

- PIE-41315
  When Integration Server receives a request with a REST-style URL and an OAuth Bearer token, Integration Server rejects the request with a 403 HTTP status code.
  This issue is now resolved. An OAuth Bearer token can be used to access a REST resource that is in the scope for which the token was issued.
  Note: Because the service name does not appear in a REST-style URL, do not include the service name when defining the OAuth scope. A scope for an Integration Server REST resource must include the full name of the resource but not the name of the service.

- PIE-41320
  Memory leak in transaction manager.
  Transaction manager retains an association to com.wm.app.b2b.server.ServerThread which is not being removed during transaction manager cleanup. Consequently, the association between the transaction and the server thread is not being cleaned up, which prevents the transaction manager entry from being removed.
  This issue is now resolved.

- PIE-41324
  Integration Server Administrator displays a Package Load Error loading a package created on Integration Server 7.x or earlier.
  Using Integration Server Administrator to load a package created on Integration Server 7.x might result in a Package Load Error caused by a NullPointerException.
  This issue was originally resolved in fixes that included PIE-33326. Unfortunately, recent revisions of some Integration Server fixes introduced a regression that allowed the problem to reappear. This fix resolves the regression.

- PIE-41345
  The pub.utils.messaging:migrateDocTypesTriggersToUM service does not migrate filters with parentheses properly.
  When migrating a trigger filter condition that contains parentheses, the pub.utils.messaging:migrateDocTypesTriggersToUM service removes the parentheses from the filter. The pub.utils.messaging:migrateDocTypesTriggersToUM service now preserves parenthesis when migrating trigger filter conditions.

- PIE-41375
  Integration Server does not indicate when it is disconnected from a cluster.
  Integration Server does not log an error message when it is disconnected from a cluster, preventing automatic detection of the situation. Now, Integration Server logs the following error message when it is disconnected from the cluster.
  [ISS.0033.151] The cluster is now not operational,
  Additionally, Integration Server logs the following error message when it rejoins the cluster.
  [ISS.0033.152] The cluster is now operational.

- PIE-41444
  Integration Server throws a NullPointerException instead of logging trace level information when the HTTP Header (0038) server log facility is set to Trace and an incoming request does not contain any credentials.
  This issue is now resolved.

- PIE-41464
  Exception handling for the pub.client:smtp service results in a ClassCastException being thrown for a SocketTimeoutException or a ConnectException.
  The pub.client:smtp incorrectly handles the exceptions java.net.SocketTimeoutException and java.net.ConnectException by throwing them as a java.lang.ClassCastException.
  This issue is now resolved.

- PIE-41497
  A request that includes a URL alias for a REST resource and includes an OAuth access token fails. When a URL alias exists for a REST resource and partial matching is enabled for URL aliases (watt.server.url.alias.partialMatching=true), the URL alias can be used with variable trailing tokens to create a request URL that invokes services for the REST resource and passes the $resourceID, $path, and custom variables to the services. (With the Bearer authorization scheme, the Authorization request header includes an OAuth access token instead of username and password.) However, if the request URL uses the Bearer authorization scheme, the request fails and Integration Server returns an HTTP 401 status code.
  This issue is now resolved. A request that uses the Bearer authorization scheme can also use a URL alias to request a REST resource.

- PIE-41502
  When acting as a web service client, Integration Server displays an error if the response from a web service provider contains the value of the content-encoding header as identity.
  When acting as a web service client, Integration Server displays the following error when the response from a web service provider contains the value of the content-encoding header as identity: "Received unsupported content-encoding: identity"
  This issue is now resolved.

- PIE-41529
  Info message is logged continuously in the server log.
  If you configure the email port to IMAP email server and select "Log out after each mail check" functionality, then the following Info message is continuously logged in the server log: "[ISP.0068.0029I]Logging out of IMAP Server"
  This issue is now resolved. Now, this message is moved to DEBUG level.

- PIE-41582
  Integration Server Administrator does not allow Username/Password as the Client Authentication method if Realm URL specifies nsps or nhps as the protocol.

When creating a Universal Messaging Connection Alias, Integration Server Administrator performs a UI validation to check if Realm URL specifies nsps or nhps as the protocol. Integration Server Administrator displays the following message when you select Username/Password as the Client Authentication method. "Client authentication must be set to SSL if Realm URL specifies includes nsps or nhps."
This issue is resolved. The UI validation is now disabled which allows you to select Username/Password as the Client Authentication method if Realm URL specifies nsps or nhps as the protocol.

- PIE-41611
  Client side queue for webMethods messaging does not drain completely.
  When Integration Server drains the client side queue by publishing or delivering messages from the client side queue to the webMethods messaging provider, some failures may occur. Integration Server handles the failures based on the exception type. If the exception type is fatal, Integration Server writes the message to the audit log and removes it from the client side queue. If the exception type is transient, Integration Server retries publishing or delivering the message. However, if the client side queue contains published and delivered messages and fatal exception occurs during publish or delivery of a message, Integration Server might add the message back to the client side queue instead of writing it to the audit log. This prevents the client side queue from fully draining and may cause the queue to drain slowly.
  This issue is now resolved.

- PIE-41677
  Integration Server fails to clear the connections of failed remote service executions that were invoked using the pub.remote:invoke service.
  This issue is now resolved. Now, if a remote service execution fails, then the connection to remote server is closed completely.

- PIE-41782
  No option to select Java Secure Socket Extension (JSSE) socket factory for HTTPS outbound connections.
  When accessing a web service through HTTPS protocol using the pub.client:soapClient service, there is no option to select JSSE for HTTPS outbound connections. The server configuration property "watt.net.ssl.client:useJSSE" applies to all outbound HTTPS connections and hence, user cannot select JSSE for individual web services call.
  This issue is now resolved. Now, an optional input parameter "useJSSE" is added to the pub.client:soapClient service.
  If the value of "useJSSE" field is left empty or unspecified, then Integration Server uses JSSE for outbound web service call based on the server configuration property "watt.net.ssl.client.useJSSE".
  If the value of "useJSSE" field is set to "yes", then the Integration Server uses JSSE for the outbound web services call.
  If the value of "useJSSE" field is set to "no", then the Integration Server uses Entrust IAIK library for the outbound web services call.

- PIE-41794
  Changes to Integration Server to notify a caller that provides an expired session id.
  After applying a fix or installing a release that includes PIE-37166, Integration Server prompts users for credentials if no matching session object can be found for a given session ID stored in a cookie. This behavior is controlled by the watt.security.session.forceReauthOnExpiration server configuration property. When set to true, the more secure behavior, Integration Server rejects requests that include a session id mapped to an expired or invalid session object and prompts the user for credentials. Setting the server configuration property to false instructs Integration Server to create a new session object, assuming the session id is valid and trusted. In case of the server configuration property value set to true (the default), Integration Server does not notify the caller that a valid session object no longer exists. Nor does Integration Server request that the caller deletes the cookie with the expired session.
  Now, when Integration Server receives a request with a session id mapped to a session object that no longer exists, Integration Server notifies the caller and requests that the cookie containing the expired session id be deleted. It is then up to the caller, usually a browser, to react to the notification.

- PIE-41806
  Enhancement to allow loggers on multiple Integration Servers to share a Universal Messaging queue for logging across a Universal Messaging realm.
  The naming convention for the Universal Messaging queue used by a logger now includes the client prefix if the Share Client Prefix check box is selected for the Universal Messaging connection alias used by the logger. By including the client prefix in the namespace of the queue name, loggers for Integration Servers in a stateless or stateful cluster can share same Universal Messaging queue across a Universal Messaging realm. That is, loggers on multiple Integration Servers can write log entries to and read log entries from one Universal Messaging queue shared across a Universal Messaging realm.
  Previously, the Universal Messaging queue name did not include the client prefix as part of the queue namespace regardless of the state of the Share Client Prefix check box. If the Share Client Prefix check box is selected for the Universal Messaging connection alias, the Universal Messaging queue name uses the naming convention: wm/is/audit/clientPrefix/logger_nameQueue (for example, wm/is/audit/myClientPrefix/SessionQueue). If the Share Client Prefix check box is not selected for the Universal Messaging connection alias, the Universal Messaging queue name uses the naming convention: wm/is/audit/logger_nameQueue (for example, wm/is/audit/SessionQueue). If you change the state of the Share Client Prefix check box on the Universal Messaging connection alias is used by loggers, you must restart Integration Server for changes to the queue name to take effect. Additionally, you must make the same changes to other Integration Severs in the cluster that use the alias.

- PIE-41825
  Cookies not included in some HTTP requests.
  When the pub.client:http service is used to send requests to a remote server, and the server returns a cookie, in some cases the cookie is not included in subsequent calls to the server.
  With this fix, the cookies are used on subsequent calls.

- PIE-41880
  webMethods Messaging triggers do not recognize nested documents in a filter when parent and child documents have the same name.
  The name of inline document fields are represented as "Message" names in the protobuf descriptors used by Universal Messaging filters. Because of the message name conflict, the filters are never satisfied and the triggers do not fire.
  This fix corrects the problem by using the name of the parent plus the actual field as the protobuf message name. After you install this fix, edit and save the root document type, then sync it with the UM Provider. The filters will begin to work properly.

- PIE-41900
  The pub.client.http fails to set the auth token.
  If the 'auth' type is set to Bearer, then the pub.client.http service fails to set the auth token into the Authorization header while sending a request to the HTTP server.
  This issue is now resolved.

- PIE-41986
  After installing a fix or a release that includes PIE-37166, a dynamic server page (DSP) request that includes a service invoke of a stateless service prompts for credentials even though the service and DSP are secured with the Anonymous ACL.
  The resolution for PIE-37166 introduced the server configuration parameter watt.security.session.forceReauthOnExpiration, which, when set to true, causes Integration Server to reject any request that includes a cookie identifying an expired or invalid session even if the request includes valid user credentials. That is, when set to true, Integration Server does not implicitly create a new session for an expired session and Integration Server deletes the session object for the expired session. A result of this change is that a sequence a DSP requests that includes a service invoke of a stateless service prompts for credentials even though the service and DSP are secured with the Anonymous ACL.
  Note: This issue does not occur when watt.security.session.forceReauthOnExpiration is set to false. Keep in mind that setting watt.security.session.forceReauthOnExpiration to true, the default value, offers more secure behavior.
  Now, Integration Server preserves the session object.

- PIE-42132
  Enhancement to limit the number of server log files that Integration Server keeps on the file system. Integration Server writes its server log to a file named server.log. When this file is rotated, the existing contents of server.log are saved to a new file named server.log_<current-date>_<current-time>. Over time, the number of archived server log files increases and may consume large amounts of disk space.
  To address this issue, Integration Server provides the watt.server.serverlogFilesToKeep server configuration parameter to control the number of server log files that Integration Server maintains on the file system, including the current server log file. When Integration Server reaches the limit for the number of server log files, Integration Server deletes the oldest archived server log file each

time Integration Server rotates the server log.
If the parameter value is 30, for example, Integration Server keeps the current server log file and up to 29 previous server log files.
If the parameter value is less than 1, Integration Server keeps an unlimited number of server log files.
If the parameter value is 1, Integration Server keeps the current server log file (server.log) and no previous (archived) server logs.
The default for watt.server.serverlogFilesToKeep is -1, which places no limit on the number of server log files kept on the file system.
You must restart Integration Server for changes to this parameter to take effect.

- PIE-42173
  New parameter for existing service pub.cache.serviceResults:resetServiceCache and new service pub.cache.serviceResults:addCacheEntry are added to Integration Server as part of this fix.
  pub.cache.serviceResults:addCacheEntry
  WmPublic. Adds cached entry into service results for the service without executing the actual service. This service can be useful to perform bulk load of service results at Integration Server startup.
  Input Parameters <serviceName> String Required. Name of the service for which to add the cache entry into cached service results.
  <input> Document Required. An IData containing the key/value pairs for the cached service input.
  <output> Document Required. An IData containing the key/value pairs for the cached service output.
  Output Parameters None.
  Usage Notes If <serviceName> does not exist in Integration Server, or it is not enabled to cache service results, Integration Server throws an exception.
  pub.cache.serviceResults:resetServiceCache An optional input parameter is added to delete specific cached elements for a particular service.
  <input> Document Optional. An IData containing key/value pairs that indicate the cached elements to remove.

- PIE-42308
  Support for session caching for transacted JMS connection aliases.
  Previously, Integration Server provided session caching for non-transacted JMS connection aliases. Now, Integration Server provides support for session caching for transacted JMS connection aliases when the JMS provider is Universal Messaging 9.10 or WebSphere MQ 7.5. The following paragraphs provide more information about this feature.
  When sending a JMS message, Integration Server creates and closes a new JMS session object and a JMS MessageProducer object for each message. This can introduce overhead for some JMS providers. To improve performance when sending JMS messages using a transacted JMS connection alias, you can configure session caching for a JMS connection alias.
  For each JMS connection alias, Integration Server can create the following session pools:
  - A default session pool containing JMS Session objects. When a default session pool is defined for a JMS connection alias, Integration Server draws from a pool of open JMS Sessions for sending a JMS

message instead of opening and closing a JMS Session for each JMS message. Integration Server uses the default session pool only when sending a message to a destination that does not have its own pool. Integration Server creates a new MessageProducer each time it sends a JMS message.
- Destination-specific session pools containing JMS Session objects for sending JMS messages to identified destinations. Integration Server creates a pool for each specified destination. When sending a JMS message to one of the specified destinations, Integration Server uses a Session object from the pool instead of creating and closing a Session object for each JMS message. Integration Server creates a new MessageProducer each time it sends a JMS message.
Note: When using destination-specific session pools for a transacted JMS connection alias, Integration Server creates a new MessageProducer each time it sends a JMS message. This is different from the destination pools that Integration Server creates for a non-transacted JMS connection alias where each entry in the destination-specific pools contains a Session object and a Message Producer object.
Note: A transacted JMS connection alias specifies LOCAL_TRANSACTION or XA_TRANSACTION for the Transaction Type.
To configure session caching for a transacted JMS connection alias, do the following:
1. Using Integration Server Administrator, open the JMS connection alias for editing.
2. Under Producer caching, for Caching Mode, select ENABLED PER DESTINATION.
3. To configure the size of the default session pool for this alias, specify the following:
- In the Minimum Pool Size field, specify the minimum number of entries in the default session pool for this alias. The default is 1.
- In the Maximum Pool Size field, specify the maximum number of entries in the default session pool for this alias. The default is 30.
4. To configure the size of the session pools created for specific destinations, specify the following:
- In the Minimum Pool Size Per Destination field, specify the minimum number of entries in each destination- specific pool.
- In the Maximum Pool Size Per Destination field, specify the maximum number of entries in each destination-specific pool. A value of 0 (or blank) indicates that Integration Server does not create separate pools for any of the destinations associated with the JMS connection alias.
In the Destination Lookup Name List, specify a semicolon-delimited list of the lookup names for the destinations for which you want Integration Server to create separate session pools for this alias.
5. Click Save Changes.
This issue is now resolved.

- PIEAR-799
The threads which creates the connection and the threads which ends the connection from the connectionPool can go to blocked state.
To recover from such a situation, Connection pool now has a monitor thread which is a pool interrupter thread, to interrupt the server threads which are in blocked state while creating or ending the connection.
The following properties are introduced to configure the connection time outs:
1-> watt.server.jca.connectionPool.threadInterrupter.sleepTime. The default value of the watt property is 2000msec, which is the sleep time for the pool interrupter thread.
2-> watt.server.jca.connectionPool.threadInterrupt.waitTime. This property specifies the wait time, measured in milliseconds, that elapses before Integration Server Connection pool interrupts a

connection creating or ending thread which is in a wait state. The pool interrupter thread will start monitoring the server threads, only if this property is set. There is no default value.
You must restart Integration Sever for changes to this parameter to take effect.
Use the following service to manage the connections which are hung:
pub.art.connection:getInterruptedThreadStatus. This service takes the connection alias name as input and lists the connection threads which are not responding even after interrupting by the connection pool interrupter. We recommend a manual intervention for the server threads which gets hung while creating or destroying the connections and also not responding to the interrupt mechanism.

Release 9.10

- PIE-36529 (IS_8.2_SP2_Core_Fix18, IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  An outbound web service request that used MTOM attachments causes out of memory issues if the attachments are large.
  An outbound web service request incorrectly used a memory-buffered output stream when requests with large MTOM attachments were made to external endpoints.
  Outbound web service requests that use MOTM attachments now use the correct output stream and chunk the large MTOM attachment.

- PIE-36506 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  The pub.security.keystore:setKeyAndChain service clears security parameters after each outbound call instead of when the pub.security.keystore:clearKeyAndChain service executes.
  In previous releases, Integration Server did not clear the security parameters configured for the outbound call by the pub.security.keystore:setKeyAndChain service until the pub.security.keystore:clearKeyAndChain executed. Now, Integration Server clears the security parameters after each outbound call. The new behavior is not backward compatible.
  Now, the security values configured by pub.security.keystore:setKeyAndChain stay in effect until pub.security.keystore:clearKeyAndChain gets invoked.

- PIE-37676 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  An attempt to authenticate a cache manager user fails with an IllegalThreadStateException.
  When authenticating users of a cache manager, a race condition caused an IllegalThreadStateException to be thrown.
  The race condition that caused the IllegalThreadStateException is now resolved.

- PIE-36620 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Integration Server returns an HTTP 500 error in response to successive requests made on persistent HTTP connections.
  When client reuses a persistent HTTP connection to send multiple requests, Integration Server should handle the request and send a response. However, Integration Server sometimes incorrectly returned a HTTP 500 error, which caused the request to fail.

This issue is now resolved.

- PIE-37093 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  When using the VCS Integration feature in Integration Server, delete and safe delete of a version-controlled node fails with an error.
  This issue is now resolved.

- PIE-37325 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Integration Server does not send HTTP cookies after a request is redirected.
  When the pub.client:http service sends a request to an HTTP server and the HTTP server responds with a redirection (a 300-level status code) and a Set-Cookie header, Integration Server includes the cookie in the request to the HTTP server to which the request is being redirected. However, subsequent requests to the second server do not include the cookie.
  Now, when a HTTP server includes a Set-Cookie header in a redirection response, Integration Server includes the cookie in the request and subsequent requests sent to the HTTP server to which the request is redirected.

- PIE-38245 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix13)
  The pub.flow:transportInfo document type signature is incorrect as seen on the Input/Output tab. The signature for the pub.flow:transportInfo document type is missing 2 fields: filePolling/lastModified and filePolling/length.
  This issue is now resolved. The signature of the pub.flow:transportInfo document type now includes the filepolling/lastModified and filepolling/length parameters.

- PIE-38290 (IS_8.2_SP2_Core_Fix19, IS_9.0_SP1_Core_Fix13)
  Integration Server resets the central user management system when lookup for a user in the central user management fails with an error.
  The reset causes a failure of other operations on the central user management system.
  This issue is resolved.

- PIE-37898 (IS_8.2_SP2_Core_Fix19, IS_9.6_Core_Fix9)
  Integration Server becomes unresponsive during startup.
  Integration Server sometimes became unresponsive at startup when initializing the scheduler. This occurred because of a database deadlock.
  This issue is now resolved and the deadlock will no longer occur.

- PIE-36477 (IS_8.2_SP2_Core_Fix19)
  After transitioning to or from daylight savings time, scheduled tasks run twice or not at all.
  Integration Server runs scheduled tasks based on the time relative to the time zone. This caused issues when entering and exiting daylight savings time, specifically tasks ran twice or not at all.
  Now, Integration Server runs scheduled tasks based on coordinated universal time (UTC). Because Integration Server runs the tasks without regard to the relative time zone, the start and end of daylight savings time does not affect the execution of scheduled tasks.

- PIE-36386 (IS_9.0_SP1_Core_Fix11, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix8, IS_9.7_Core_Fix5)

When a flow service invokes a service that does not exist, Integration Server handles operations on the pipeline first before issuing a runtime exception.
When a flow service invokes a service that does not exist, Integration Server performs operations on the pipeline, such as removing the dropped variables from the pipeline, before issuing a com.wm.lang.flow.FlowException about the unknown service.
This issue is resolved. Integration Server now issues the runtime exception when a flow service invokes a service that does not exist and will not proceed with the pipeline operations.

- PIE-35990 (IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix8)
  When using Oracle Streams Advanced Queuing (AQ) as the JMS provider, if the JMS connection alias is disabled due to an error from the JMS provider, the JMS connection continues to show the status as "in progress".
  If a JMS connection alias, when attempting to establish a connection with Oracle Streams Advanced Queuing (AQ) as the JMS provider, is disabled due to an error from the JMS provider, the JMS connection continues to show the status as "in progress" and cannot be enabled or disabled.
  This issue is resolved.

- PIE-37022 (IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix8)
  Integration Server logs the following error twice for the same port in the server logs while shutting down:
  [ISP.0046.0011I] Disabling HTTP Listener on port <port number>
  This issue is resolved. Integration Server now logs this message only once while shutting down.

- PIE-35873 (IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  Change to allow use of forward slash in a regular expression.
  Prior to this change, regular expressions used in BRANCH steps and trigger filters could not include a forward slash. Now, a forward slash can be used. However, the forward slash must be preceded by the backslash (\) escape character. For example, to use a regular expression to match a variable whose value contains the string “a/b”, use the regular expression %variableName% = /a\/b/

- PIE-37271 (IS_9.0_SP1_Core_Fix12, IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  In certain scenarios, invoking the pub.xml:getNextXMLNode service fails with an exception.
  Invoking the pub.xml:getNextXMLNode service fails with a NullPointerException if the specified NodeIterator is returned by the pub.xml:getXMLNodeIterator service that has the criteria input parameter set to a string value that matches an element node of the XML node.
  This issue is resolved.

- PIE-34881 (IS_9.0_SP1_Core_Fix12)
  Integration Server does not behave as expected when validating an XML document against a document type created from an XML schema definition containing <xs:restriction base="xs:anyType">.
  When using the pub.schema:validate service to validate an XML document against a document type created from an XML schema definition containing <xs:restriction base="xs:anyType">, the validation does not fail even if the target element contains attributes other than the attributes defined in the XML schema definition.

This issue is resolved. Now, validation of XML fails if there are any attributes other than the defined attributes.

- PIE-36217 (IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  An outbound HTTPS web service request fails when using JSSE.
  If an outbound HTTPS web service request specifies the use of JSSE, the request fails because Integration Server does not send the client certificates to the endpoint.
  This issue is now resolved.

- PIE-37071 (IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Integration Server error log does not include the names of the affected JDBC pools in the error messages.
  When issuing error messages such as, "java.sql.SQLException: Could not get connection", Integration Server error log does not include the names of the JDBC pools to which the error message is referring.
  This issue is resolved. Integration Server error log now includes the name of the JDBC pools in its error messages.

- PIE-37551 (IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Processing of webMethods messaging triggers fails with a java.lang.ClassCastException if the watt.server.trigger.local.checkTTL server configuration parameter is set to true.
  Processing of webMethods messaging triggers fails with the following error if the watt.server.trigger.local.checkTTL server configuration parameter is set to true:
  Unable to check document ttl: java.lang.ClassCastException: [B cannot be cast to com.wm.data.IData
  This issue is now resolved.

- PIE-37645 (IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Incorrect logging level specified for error message.
  Integration Server gives the logging level of the following error message of Trace logging level incorrectly as Info:
  [ISC.0042.0001I] baseURL (Connection reset)
  This issue is now resolved. The logging level of the message is corrected to Trace.

- PIE-38560 (IS_9.0_SP1_Core_Fix13, IS_9.5_SP1_Core_Fix10)
  Integration Server logs a NullPointerException at one minute intervals.
  Once per minute, the Integration Server scheduler uses a background thread to check for scheduled tasks in an invalid state. If Integration Server is in a cluster and the watt.server.scheduler.logical.hostname property is not set, a NullPointerException occurs and is written to the error log.
  Now, the background thread used by the scheduler no longer throws a NullPointerException if Integration Server is in a cluster and the watt.server.scheduler.logical.hostname property is not set.

- PIE-35697 (IS_9.0_SP1_Core_Fix13)
  After refreshing a consumer web service descriptor, response services reference the backup folders.
  After refreshing a consumer web service descriptor, the response services contain references to the backup folders.

This issue is now resolved.

- PIE-37921 (IS_9.0_SP1_Core_Fix13)
  The pub.client:http service does not include the supplied data in PUT requests.
  When using the PUT method with the pub.client:http service, the request transmitted to the remote HTTP server does not include the value supplied for the data/args, data/string, data/table, data/bytes, data/stream or data/mimeStream input variables.
  This issue is now resolved. When the PUT method is used with pub.client:http, the value of the data/args, data/string, data/table, data/bytes, data/stream or data/mimeStream input variables is now included as the HTTP entity of the transmitted request.

- PIE-37988 (IS_9.0_SP1_Core_Fix13)
  A NullPointerException occurs when SoftwareAG-IS-Services.xml configuration file is not present in the caching directory.
  When SoftwareAG-IS-Services.xml configuration file is not present in the directory location <Integration Server_directory>\instances\<instance_name>\config\Caching, Integration Server throws a NullPointerException.
  This issue is resolved. Integration Server now logs the following error message in the server logs to indicate that the configuration file is not present in the caching directory.
  Cannot cache "{0}" service. The cache manager SoftwareAG.IS.Services is not initialized."

- PIE-38019 (IS_9.0_SP1_Core_Fix13)
  Integration Server acting as an SFTP client faces issues while attempting to connect to an SFTP server.
  When attempting to connect to an SFTP server, Integration Server acting as an SFTP client issues the following error:
  [ISS.0147.9010] Cannot get host key from server [host_X]:22.
  Details: com.jcraft.jsch.JSchException: Algorithm negotiation fail
  This issue occurs because there is no common key exchange algorithm between the SFTP client and SFTP server.
  This issue is now resolved by updating the jsch jar file from 0.1.51 version to 0.1.53 version. The 0.1.53 version supports most of the key exchange algorithms that are required to be present in the SFTP client to connect to SFTP server.

- PIE-38429 (IS_9.0_SP1_Core_Fix13)
  Integration Server logs an exception when a service is invoked through a JSSE-enabled HTTPS port with client authentication set to "Username/Password" or "Request Client Certificates".
  When a service is invoked through an HTTPS port that uses JSSE and has client authentication set to "Username/Password" or "Request Client Certificates", Integration Server logs the following exception in the error log:
  javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated.
  The issue is now resolved. Integration Server does not log any exception upon invoking a service successfully through a JSSE-enabled HTTPS port with client authentication set to "Username/Password" or "Request Client Certificates".

- PIE-38599 (IS_9.0_SP1_Core_Fix13)

The pub.client:http service does not honor the value set for the useJSSE parameter.
The useJSSE parameter of the pub.client:http service indicates whether to enable the use of the Java Secure Socket Extension (JSSE) socket factory for creating outbound HTTPS connections. However, the pub.client:http service fails to consider the value set for the useJSSE parameter and does not use JSSE for SSL connection with proxy configuration even if the "useJSSE" parameter is set to yes.
This issue is now resolved. The pub.client:http service now honors the value set for the useJSSE parameter.
- If "useJSSE" is set to yes, Integration Server uses JSSE for HTTPS connection.
- If "useJSSE" is set to no, Integration Server uses Entrust IAIK library for HTTPS connection.

- PIE-39117 (IS_9.0_SP1_Core_Fix13)
  Subsequent HTTPS requests through proxy to the server fail.
  When you send repeated outbound HTTPS requests to a server using the pub.client:http service with useJSSE=yes and HTTPS proxy alias, the first request is sent successfully but the subsequent requests fail with exception.
  The issue is now resolved.

- PIE-37937 (IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix8)
  Publishing a document as a JMS message from Designer fails with a NamingException.
  Using Designer to publish an instance of an IS document type as a JMS message fails with a NamingException. This occurs because the pub.jms:send service, which is used by Designer to publish the JMS message, includes a new input parameter named JMSMessage/header/replyTo that Designer populates with an empty String. Integration Server attempts a JNDI lookup using the empty String, which results in the NamingException.
  Now, Integration Server verifies that JMSMessage/header/replyTo parameter is not empty before performing the JNDI lookup.

- PIE-34925 (IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  When executing a web service connector, Integration Server does not check the Execute ACL of the parent consumer web service descriptor.
  When executing a web service connector contained in a consumer web service descriptor that has a Pre-8.2 compatibility mode value of false, Integration Server did not check the Execute ACL of the consumer web service descriptor. The documented behavior is to always verify the Execute ACL for a web service descriptor, but it was not being done.
  Now, when executing a web service connector, Integration Server checks the Execute ACL of the parent consumer web service descriptor.

- PIE-37804 (IS_9.5_SP1_Core_Fix10, IS_9.6_Core_Fix9)
  Integration Server KPI that Optimize uses to display lifetime statistics displays current statistics instead.
  The Integration Server KPI, getNumCompletedRequests, which Optimize uses to display the number of completed requests over the lifetime of the Integration Server, returns the number of requests in the last polling period instead.
  This issue is now resolved.

- PIE-37586 (IS_9.5_SP1_Core_Fix10)

Integration Server deletes the parent task when all the servers in a cluster are shut down. When you schedule a task to run on all servers in a cluster, and when one server in the cluster is shut down, Integration Server deletes the corresponding child task. However, when all the servers in the cluster are shut down, Integration Server deletes all the child tasks as well as the parent task. As a result, the parent task information is lost upon restart of Integration Server.
This issue is resolved.

- PIE-38131 (IS_9.5_SP1_Core_Fix10)
  Change to Integration Server to allow the pub.client:http service to return response headers and response entity when receiving a 501 to 599 level response from a remote HTTP server.
  HTTP servers indicate success or failure by including a status code in the response. The pub.client:http service returns the status code, response headers, and response body. When the HTTP response includes a status code in the 501 to 599 range, the pub.client:http service throws a ServiceException. Some HTTP servers include useful information in the response headers and entity when returning a 501 to 599 level status code. However, this information is lost when pub.client:http throws a ServiceException.
  To address this issue, Integration Server now includes a server configuration parameter, watt.net.http501-599.throwException, which you can use to indicate how the pub.client:http service handles a 501 to 599 level response.
  watt.net.http501-599.throwException
  Specifies whether or not the pub.client:http service throws a ServiceException when receiving a 501 to 599 level response from a remote HTTP server. When set to true, the pub.client:http service throws a ServiceException when it receives a 501 to 599 level response from a remote HTTP server. When set to false, the pub.client:http service does not throw a ServiceException when it receives a 501 to 599 level response from a remote HTTP server. Instead, when the pub.client:http service returns a status code in the 501 to 599 range, the service returns the status code, response headers, and response body. The default is true. You do not need to restart Integration Server for changes to this parameter to take effect.
  Note: When the remote HTTP server returns a response code of 500, the pub.client:http service returns the status code, response headers, and response body.

- PIE-38442 (IS_9.5_SP1_Core_Fix10)
  Integration Server does not handle an element with xsi:nil=1 correctly.
  If an IS document type used by the web service contains a field of type Object for which the Allow null property is set to true and the XML instance document contains a corresponding element with an xsi:nil=1 and an xmlns:xsi= attribute, Integration Server does not properly convert the element. Instead, Integration Server incorrectly converts the element to a Document field with no children instead of converting the element to a field of type Object with a “null” value.
  This issue can also occur when the pub.xml:xmlNodeToDocument service executes and all of the following are true:
  - The preserveUndeclaredNS input parameter is set to true
  - The IS document type in the documentTypeName field contains a field of type Object for which the Allow null property is set to true
  - The supplied node contains a corresponding element with an xsi:nil=1 and an xmlns:xsi=

  Now, Integration Server properly converts the xsi:nil=1 element to an field of type Object with a "null" value.

- PIE-36837 (IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  Sessions are not being shared among Integration Servers in a cluster.
  When a dynamic server page (DSP) is requested by a client, a session is created in Integration Server and a Set-Cookie response header is returned with the session ID. If a subsequent request from the client includes the session ID but a different Integration Server in the cluster receives the request, the second server should find the session in the cluster's session store and then use the session to service the request. This was not happening. The first server did not save the session to the cluster's session store. As a result, the client's session was not available to the second server. This has been fixed. When a clustered Integration Server receives a request for a DSP, the server saves the session to the cluster's session store.

- PIE-36942 (IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  Integration Server may become unresponsive when processing concurrent requests to disable file polling ports in different packages.
  If Integration Server receives concurrent requests to disable a file polling port in a package and the ports are located in different packages, Integration Server may become unresponsive. Additionally, HTTP/S ports on Integration Server might become unavailable.
  This issue is now resolved.

- PIE-37005 (IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  Enhancement to pub.jms:send service to allow specifying a destination for the JMSReplyTo header. Currently, if you want to specify the JMSReplyTo header for a JMS message that you are sending, you must use the pub.jms:sendAndWait service. The pub.jms:sendAndWait service sends a request message and waits for a response. The act of waiting for a response message comes with extra overhead for Integration Server which is unnecessary if you merely want to specify a JMSReplyTo destination but do not want the sending service to wait for a reply.
  To address this issue, the pub.jms:send service has been enhanced to include a new input parameter for setting the JMSReplyTo header. When invoking the pub.jms:send service, you can set the JMSMessage/header/replyTo field which is an optional parameter of type String. Set this parameter to one of the following:
  - If the JMS connection alias used by the pub.jms:send service connects to the JMS provider using JNDI, set JMSMessage/header/replyTo to be the lookup name of the destination lookup object name.
  If the JMS connection alias used by the pub.jms:send service connects to the JMS provider using a native Broker connection, set JMSMessage/header/replyTo to the Broker queue name. That is, if the JMS connection alias specifies the Broker as the JMS provider and uses the native webMethods API to connect directly to the webMethods Broker, specify the name of the queue on the Broker that should receive replies to the message.
  Note: When using the native webMethods API to connect to the Broker, the JMSMessage/header/replyTo destination must be a queue. Topics are not supported.
  When executing the pub.jms:send service with a valid value for the JMSMessage/header/replyTo parameter, Integration Server creates the javax.jms.Destination and maps it to the JMSReplyTo field within the message header. Integration Server sends the message and returns immediately. The service does not wait for the response message.
  If JMSMessage/header/replyTo is empty, then Integration Server does not set the JMSReplyTo

header for the JMS message. If JMSMessage/header/replyTo is invalid, then Integration Server throws a ServiceException.

- PIE-37261 (IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix8)
  Integration Server logs the message "[B cannot be cast to com.wm.util.Values" in server.log file".
  If the ServiceResults cache has the Copy on Read and Copy on Write check boxes selected and the number of entries in the cache exceeds the value specified for Maximum Elements in Memory, Integration Server logs the following error message to the server log: [B cannot be cast to com.wm.util.Values
  This issue is now resolved.

- PIE-37126 (IS_9.5_SP1_Core_Fix9, IS_9.6_Core_Fix9)
  In the Integration Server Administrator page, the Routing option under Adapters is not available in some cases.
  After a successful installation of the OnRamp for Commerce One Marketplace Adapter, the Routing option under Adapters in the Integration Server Administrator page is not available. The Integration Server console is unable to add this option in the Integration Server Administrators page because an incorrect configuration in the Server Log Facilities page results in a failure in the initialization process for the adapter.
  This issue is resolved. The Routing option is now available under the Adapters section in the Integration Server Administrator page.

- PIE-35036 (IS_9.6_Core_Fix8, IS_9.7_Core_Fix5)
  Integration Server does not reflect the changes made to the Create Temporary Queue option after the JMS connection alias is created.
  If the Create Temporary Queue option is selected while creating a JMS connection alias and if this selection is cleared later, Integration Server does not reflect this change. The setting of the Create Temporary Queue option that was chosen while creating a JMS connection alias is maintained.
  This issue is resolved. Integration Server now reflects the changes made to the Create Temporary Queue option after the JMS connection alias is created.

- PIE-37419 (IS_9.6_Core_Fix9)
  File polling consumed server threads that were not terminated at the end of file polling process.
  The thread leak related to file polling ports is now resolved.

- PIE-37581 (IS_9.6_Core_Fix9)
  Integration Server erroneously logs a message about the connection pool threshold being exceeded.
  Integration Server erroneously writes the following message to the server log:
  [ISS.0096.0008I] JDBC Connection pool threshold exceeded, 0 available for pool CentralUsers:CentralUsersPool
  This issue is now resolved.

- PIE-37893 (IS_9.6_Core_Fix9)
  The FTP command PWD returns incorrect structure in certain situations
  The FTP command PWD sometimes returns an incorrect directory path .
  The PWD command now returns a valid directory path.

- PIE-37268 (IS_9.6_Core_Fix9)
  When JSSE is enabled for outbound HTTP connections from Integration Server, watt.security.ssl.client.ignoreEmptyAuthoritiesList does not work as expected.
  When JSSE is enabled for outbound HTTP connections, Integration Server client accepts empty trusted authorities lists from the SSL server but does not return its CA certificate even if the watt.security.ssl.client.ignoreEmptyAuthoritiesList is set to true.
  This issue is resolved. Now, when JSSE is enabled for outbound HTTP connections, Integration Server client accepts empty trusted authorities lists from the SSL server and returns its CA certificate if the watt.security.ssl.client.ignoreEmptyAuthoritiesList property is set to true.

- PIE-36514
  Integration Server creates an exception when you invoke a flow service exposed as a web service.
  Integration Server creates the following exception when you try to invoke a flow service exposed as a web service:
  ServiceException: Fault returned by invoked service.
  This exception which is created in the stack before the service is executed is not logged in the Integration Server log but, monitoring tools like Dynatrace might interpret the exception as an error even though the service is successfully executed.
  This issue is resolved.

- PIE-37042
  When invoking the pub.client:http service with the useJSSE input parameter set to yes, a java.lang.ClassCastException error occurs if the server configuration parameter, watt.security.ssl.keypurposeverification is set to true.
  This issue is resolved.

- PIE-37056
  When the pub.client:http service sends an HTTPS request, the SSL handshake process fails when the useJSSE parameter is set to yes.
  When the pub.client:http service sends an HTTPS request and the useJSSE input parameter is set to yes, during the SSL handshake process Integration Server sends its certificate even though no matching CA certificate from the other endpoint server is found in the Certificate Authorities list for the SSL handshake.
  This issue is resolved. Integration Server now sends its CA certificate during the SSL handshake process only if a matching CA certificate is available from the other endpoint server.

- PIE-37178
  The watt.server.compile parameter is not set correctly if Integration Server instance is created through migration utility.
  When running the migration utility, selecting the option to create an Integration Server instance during migration results in an incorrect value for the watt.server.compile server configuration parameter, specifically: ${javac.exe} -classpath {0} -d {1} {2}
  Now, the migration utility substitutes the value of ${javac.exe} to point to javac executable located in the JDK shipped with Integration Server.

- PIE-37308

At start up, Integration Server logs a warning message about audit logging even though no audit loggers are configured.
At start up, Integration Server attempts to determine if there are sufficient JDBC connections in the JDBC connection pool even when audit loggers are not configured. If there are no connections, Integration Server logs the following message, which is erroneous if there are no configured audit loggers:
[ISS.0095.0022W] Audit Logging Initialization: Unable to verify the Max Connections setting for the Audit Logging database.
Now, Integration Server does not check for JDBC connections at start up if there are no configured audit loggers.

- PIE-37377
  Update of scheduled tasks to use UTC fails.
  At startup, if Integration Server updates scheduled tasks to use UTC (Coordinated Universal Time) instead of local time, Integration Server writes the following message to the server.log:
  [ISS.0137.0035E] Migration of scheduled tasks to UTC timezone failed
  When Integration Server updates the tasks to UTC, Integration Server also validates the tasks, which, for a task scheduled to run on another Integration Server in a cluster, includes checking that the Integration Server is a member of the cluster. If the Integration Server is not in the cluster, task validation fails and the task timestamp is not updated to UTC. However, Integration Server incorrectly considers that the task that failed validation is now in UTC when, in fact, the task is still in local time.
  Now, the migration utility no longer validates the scheduled task during migration. Consequently, a validation failure does not prevent the migration utility from updating the scheduled tasks.

- PIE-37457
  A webMethods messaging request-reply fails when the encoding type of the reply document type is protocol buffers.
  When encoding the reply message, Integration Server uses the encoding type specified for the publishable document type used as the reply. However, reply messages are sent to a dedicated reply-to channel on Universal Messaging. The channel is generic and is not associated with any specific document type. To receive a protocol buffer encoded message, a channel must be associated with a specific document type. Because the channel is generic, a reply message encoded as a protocol buffers cannot be processed.
  To address this issue, Integration Server now always encodes reply message as IData.

- PIE-37560
  Upgrade does not include layered products if the destination Integration Server does not have an instance with the same name as the source Integration Server.
  When upgrading to an Integration Server instances that does not have the same name as the source Integration Server instance, upgrade does not include layered products such as webMethods Mediator and webMethods Monitor.
  This issue is now resolved. Upgrade always includes layered products, regardless of whether or not the destination Integration Server instance is the same as the source Integration Server instance.

- PIE-37825

When the logging level is set to Trace, upon invoking the pub.client:http service, Integration Server logs an entry for the HTTP Get method in the server log irrespective of the HTTP request type. This issue is resolved. Upon invoking the pub.client:http service, Integration Server now logs entries according to the HTTP request type.\

- PIE-37844
  After a database outage, Integration Server can run out of threads.
  When Integration Server cannot connect to the audit logging database, either due a problem in the database or a problem in the network, all threads requesting database connections can pause indefinitely. If this situation persists, all threads in the server thread pool will be paused. No new requests will be accepted.
  This issue has been fixed. A point of thread contention in the JDBC pool implementation was removed. Threads requesting database connections after a database outage no longer experience lengthy pauses.

- PIE-36702 (IS_9.8_SPM_Fix1)
  Command Central cannot be used to set the all the JDBC connection pool alias properties.
  When using Command Central to create or edit a JDBC connection pool, the Available Connections Warning Threshold and Waiting Thread Threshold Count properties cannot be specified.
  The Available Connections Warning Threshold and Waiting Thread Threshold Count properties can now be set through Command Central.

- PIE-34577
  Integration Server does not validate a certificate that uses the SHA-256 algorithm but should.
  The certificate chain verifier in Integration Server validates the certificates presented by SSL clients. The certificate chain verifier does not validate a certificate chain that uses the SHA-256 algorithm but should. Instead Integration Server throws the following error:
  [ISC.0009.9002] Error in certificate chain: Entrust cannot verify the certificate chain: [ISC.0009.9002] Error in certificate chain
  The certificate chain verifier in Integration Server now validates a certificate chain that uses the SHA-256 algorithm.

- PIE-35422
  The jcode utility fails with a MalformedPatternException on Linux.
  Execution of jcode.sh on Linux fails with the following error: Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/oro/text/regex/MalformedPatternException
  The exception occurs because one of the required jar files in common/lib/ext is missing from the jcode utility classpath.
  The jcode.sh file is now updated to include the required jars in the classpath. This issue is now resolved.

- PIE-35643
  After upgrading to version 9.7, when a webMethods messaging trigger, which has join conditions that are dependent on two or more documents, does not start because Integration Server cannot locate the publishable document types, Integration Server does not reload and start the trigger when the publishable document types are later loaded.

Integration Server does not start a webMethods messaging trigger, which has join conditions that are dependent on two or more publishable document types, if it cannot find the document types specified in the join condition. However, Integration Server issues a PDT_DOES_NOT_EXIST (InvalidDocumentException) exception stating the reason the trigger did not fully load and when Integration Server loads the package containing the publishable document types to which the trigger subscribes, Integration Server reloads the trigger. But, after upgrading to version 9.7, instead of the PDT_DOES_NOT_EXIST exception, Integration Server returns an INVALID (ServiceException) exception incorrectly. As a result, Integration Server fails to reload and start the trigger when the publishable document types are later loaded.
This issue is resolved. Integration Server now returns the PDT_DOES_NOT_EXIST exception correctly if it cannot find the document types specified in the join condition. In addition, when Integration Server loads the package containing the publishable document types specified in the join condition, Integration Server reloads the trigger.

- PIE-35716
  If the database associated with Integration Server is restarted abruptly, service invocations or database calls made through JDBC connection pool alias fail and do not return any records from the database.
  When the database associated with Integration Server is restarted abruptly, Integration Server does not close and release the existing JDBC connections that are stale due to the abrupt database shutdown and fails to return these connections to the JDBC connection pool. As a result, service invocations or database calls made through the JDBC connection pool alias fail and do not return any records from the database.
  The issue is now resolved.

- PIE-35907
  After migrating to Integration Server 9.5 or later, Integration Server displays an exception when calling a web service and using MTOM streaming.
  After migrating 2 to Integration Server 9.5 or later, sending a web service request that uses MTOM streaming for which more than one chunk is sent, the following exception occurs:
  "Exception --> org.apache.axis2.AxisFault: Connection reset by peer: socket write error"
  This issue is now resolved.

- PIE-36224
  When installed on Unix or Linux, scripts located in IntegrationServer/instances/<instanceName>/packages/<packageName> directory, such as IntegrationServer/instances/myInstance/packages/WmDeployer, do not have permission to execute.
  This issue is now resolved.

- PIE-36288
  Executing the jcode.bat/sh utility removes the name of the output template assigned to a service.
  Running the jcode.bat/sh utility removes the value of the Output template Name property for services in the package containing the Java service.
  This issue is now resolved.

- PIE-36399

Using pub.schema:createXSD to create an XML Schema definition for a document type ends with a NullPointerException.
Using the pub.schema:createXSD service to create an XML Schema definition for a document type ends with a NullPointerException if the document type contains a field named *body.
This issue is now resolved.

- PIE-36644
  Memory leak with JMS request-reply.
  Integration Server did not correctly close MessageProducer objects when invoking the pub.jms:reply service, resulting in a memory leak.
  This issue is now fixed.

- PIE-36824
  Users that are members of the Developers group cannot use Designer to build and upload processes.
  When a user that belongs to the Developers group uses Designer to build and upload a process, Designer throws the following exception:
  Error: Build of process <processName> failed.
  com.webmethods.process.connection.is.IntegrationServerConnectionException: [ISC.0064.9314] Authorization Required: [ISS.0084.9004] Access Denied
  at
  com.webmethods.process.generator.util.GeneratorUtils.getJMSProviderData(GeneratorUtils.java:934)
  at
  com.webmethods.process.generator.util.GeneratorUtils.createJNDIDestinations(GeneratorUtils.java:718)
  This issue is now resolved. Members of the Developers group can now use Designer to build and upload processes.

- PIE-36962
  Integration Server fails to enable email ports configured to use transport layer security.
  Upon creating an email port configured to use transport layer security, Integration Server fails to enable the email port and issues the following error message:
  Unsupported ciphersuite SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA
  This issue is resolved.

- PIE-36990
  Duplicate port aliases are assigned during upgrade.
  When upgrading Integration Server 9.5 or earlier to version 9.6 or later, Integration Server might assign the same port alias to multiple Internal Server ports. Upon startup of Integration Server, this may result in only one of the Internal Server ports becoming active and in the following warning: [ISS.0070.30] yyyy-mm-dd hh:mm:ss zone WARN: Duplicate alias protocolListener_portNumber_packageName encountered creating protocol listener on port portNumber.
  This occurs because the naming convention used to assign a port alias to an existing port creates a duplicate alias if more than one Internal Server ports connect to an Enterprise Gateway Server

through the same Registration port.
Now, to ensure that each port alias is unique, Integration Server includes the host name for the port in the port alias for migrated ports. The new naming convention is:
protocolListener_portNumber _hostName_packageName

- PIE-37004
  Integration Server reloads packages more than once in a run-time-based deployment.
  When there are dependencies across different Integration Server packages in a run-time-based deployment, Integration Server reloads these packages more than once.
  Now, Integration Server reloads the packages just once.

- PIE-37041
  While converting an XML node to a document, Integration Server does not issue any validation errors upon receiving empty fields that are not included in the enumeration list.
  When converting an XML node to a document that contains fields with *body attributes, the text specified in *body represents the values of the document fields. These elements can have valid values specified in the enumeration fields. However, while converting, if Integration Server receives an empty value for a field and if the empty value is not included in the enumeration list, Integration Server should issue a validation error, but does not.
  This issue is resolved. Integration Server now issues the following validation error upon receiving an empty field that is not included in the enumeration list:
  No matching enumeration value."

- PIE-37142
  On Windows, the script for creating a new instance of Integration Server does not use the JVM bundled with the product installation.
  The script for creating an Integration Server instance now uses the JVM bundled with the Integration Server product installation.

- PIE-37159
  Users can log in to Integration Server with old and new passwords.
  When central user management is configured for Integration Server and an authentication cache is enabled for Integration Server, users can log in with an old password even after successfully log in with the new password. This occurs because Integration Server does not remove the cached entry with the old password until after the time specified in watt.server.auth.cache.timeout elapses.
  Now, once a user logs in with a new password, Integration Server clears the cached password.

- PIE-37166
  Web browser users are not forced to supply credentials after their session expires.
  When Integration Server receives a request that includes a cookie that identifies a session and valid user credentials, Integration Server will do one of two things:
  - If the session identified by the cookie is valid, Integration Server will use the existing session for the request.
  - If the session identified by the cookie has expired, or is otherwise invalid, Integration Server will use the supplied credentials to authenticate the client and create a new session for the request.
  Modern web browsers cache user credentials and send them to servers with each request. For

Integration Server Administrator users, this means a user's session may expire but because the browser sends the user's credentials every time, a new session is created. The user may continue to use Integration Server Administrator without re-entering their user name and password. The only way to force the re-entry of a user's credentials is to close the browser. This may be considered a security flaw.
Integration Server addresses this situation by adding a new configuration parameter: watt.security.session.forceReauthOnExpiration. When set to true, Integration Server rejects any request that includes a cookie identifying an expired or invalid session, even if the request includes valid user credentials. The rejection response directs the browser to clear its session identifier and to prompt the user for credentials. When set to false, Integration Server creates a new session using the credentials in the cookie. The default value for watt.security.session.forceReauthOnExpiration is false. A value of true offers more secure behavior. Changes to watt.security.session.forceReauthOnExpiration take effect immediately.

- PIE-37211
  The pub.flow:getLastError service does not return a lastError document.
  The pub.flow:getLastError service does not return a lastError document if a parent SEQUENCE step specifies a timeout value but a service called by an INVOKE step within the SEQUENCE causes the SEQUENCE to time out.
  This issue is now resolved.

- PIE-37222
  A signed XML document or a node in an XML document fails verification performed by the pub.security.xml:verifyXML service.
  If the pub.security.xml:signXML service signs an XML document or a node in an XML document and the service input specifies "true" for the includeCertChain parameter or a value other than the default value "X509_CERTIFICATE" for the certData parameter, the resulting signed document fails verification performed by the pub.security.xml:verifyXML service.
  This issue is resolved.

- PIE-37276
  After migrating a consumer web service descriptor that uses Web Services Addressing (WS-Addressing) from Integration Server 8.2 to a later release, invoking a web service connector in the descriptor ends with SocketTimeoutException.
  If a consumer web service descriptor was created from a WSDL document that declared both the Web Services Addressing namespace as defined in the World Wide Web Consortium (W3C) Recommendation and the namespace from the earlier W3C Submission, after migration from Integration Server 8.2, execution of a web service connector fails with the following: java.net.SocketTimeoutException: Read timed out
  The timeout occurs because Integration Server incorrectly uses the WS-Addressing namespace from the earlier W3C Submission of WS-Addressing when executing the web service connector. As a result, the web service provider does not respond in the expected manner which causes the request to time out.
  Now, if a web service descriptor that uses WS-Addressing was created from a WSDL that declared namespace declarations for the WS-Address Recommendation and the Submission, Integration Server uses the namespace declarations defined in the Recommendation.

- PIE-37473
  Package deployment fails when the package contains a document type.
  Using Deployer to deploy a package that contains a publishable document type fails with the following error:
  [DEP.0005.0326] An error occurred while deploying package "packageName". Item "documentTypeName" could not be loaded because of reason "[ISS.0026.9112] Document type documentTypeName has failed to load. Duplicate Broker document type name brokerDocumentTypName is also referenced by documentTypeName". The package may have partially loaded; please check your target server.
  This issue is now resolved.

- PIE-37508
  Integration Server logs an erroneous exception when executing web services without an outbound callback service.
  When executing web service that is part of a provider web service descriptor for which an outbound callback service is not specified, Integration Server sometimes writes the following erroneous message to the error log:
  ISC.0088.9998E Exception --> null.
  This issue is now resolved. Integration Server no longer logs an exception when executing a web service that is part of a web service descriptor that does not specify an outbound callback service.

- PIE-37589
  XSLT services experience intermittent XSLT transformation errors or exhibit slow transformation performance.
  This fix addresses both issues.

- PIE-37790
  Enhancement to use Universal Messaging for audit logging queues.
  In asynchronous audit logging, Integration Server first writes a log entry to a queue and subsequently writes the log entry from the queue to the log destination. Previously, Integration Server provided only an internal queue, sometimes called the light-weight queue, to use as an audit logging queue. With this enhancement, Integration Server provides the option of using a Universal Messaging queue instead of the internal queue as the audit logging queue. Using Universal Messaging with audit logging offers increased performance for asynchronous and synchronous logging.

- PIE-37834
  When using the jcode utility to compile Java services, Integration Server generates wrong jcode utility classpath.
  When using the jcode utility to compile Java services, Integration Server generates wrong jcode utility classpath if the code or classes directory added to the classpath does not include path separator.
  This issue is now resolved.

- PIE-37977
  A web service connector throws a SOAPException about an invalid envelope.

When executing a web service connector for an InOnly or InOnlyRobust operation for which the parent consumer web service descriptor has the Pre-8.2 compatibility mode property set to true, Integration Server may log the following extraneous error message indicating an invalid SOAP Envelope has been received even though the connector executed successfully.
[ISS.0088.9155] this SOAPMessage does not contain a valid Envelope object
This fix eliminates the extraneous error message.

- PIE-38099
  A trigger with an AND join fails when receiving messages concurrently across a cluster of Integration Servers or a non-clustered group of Integration Servers.
  When a JMS trigger or a webMethods messaging trigger with an AND join receives messages at the same time on two different Integration Servers, one of the messages might be lost. The lost message will not be processed which prevents the AND join from being satisfied and causes the trigger to fail. This situation occurs when the trigger resides on Integration Servers in a cluster or on a group of non-clustered Integration Servers.
  This issue is now resolved.

- PIE-38134
  Enhancement to provide the ability to configure the allowed protocols for JSSE on a per port basis.
  In Integration Server, the watt.net.jsse.server.enabledProtocols server configuration parameter specifies the allowed protocols for all JSSE ports. However, there might be times where you wish to allow specific protocols for use with a particular JSSE port only. Integration Server now provides the ability to specify the allowed protocols for JSSE on a per port basis.
  Note: The 'jsseEnabledProtocols' value specified for the port record in the listeners.cnf file overrides the value set by watt.net.jsse.server.enabledProtocols server configuration parameter.
  Note: When the logging facility 0006 Server SSL Interface is set to the Debug logging level, Integration Server writes messages about protocols used for inbound and outbound ports to the server log. At the Trace logging level, Integration Server writes messages about the enabled cipher suites. You can use these server log messages to confirm the enabled protocols for any JSSE port.

- PIE-38158
  The pub.utils.messaging:migrateDocTypesTriggersToUM service incorrectly returns a message indicating a filter was successfully converted.
  When the pub.utils.messaging:migrateDocTypesTriggersToUM service cannot convert a filter for a webMethods messaging trigger because of a filter conversion error, the service returns the error message in the failedTriggers/reason parameter. However, the service also returns a message stating that the filter was successfully converted, which is not correct.
  If a filter cannot be converted successfully, the pub.utils.messaging:migrateDocTypesTriggersToUM no longer returns a message about successful conversion of the filter.

- PIE-38244
  Web service fails with a RampartException while handling a holder-of-key SAML assertion.
  When handling a holder-of-key SAML assertion, a web service fails with the following exception: RampartException: Invalid signature algorithm for Asymmetric binding.
  This issue is now resolved.

- PIE-38300
  A concurrent JMS trigger makes a retry attempt after Integration Server suspends the trigger because of a transient error.
  When a concurrent JMS trigger encounters a transient error when processing a message, the trigger makes an extra retry attempt after the trigger is suspended. For example, if Max retry attempts is 3, Integration Server suspends the trigger after the third retry attempt fails. However, the trigger makes a fourth retry attempt.
  This issue is resolved. When Integration Server suspends a concurrent JMS trigger because of a transient error, the trigger does not make an additional retry attempt.

- PIE-38391
  In a clustered environment, Integration Server sometimes does not create child tasks when a new server is added to the cluster or when an existing server is restarted.
  When a task is scheduled to run on all servers in a clustered environment, Integration Server creates a parent task and a child task for each server in the cluster. When a new server is added to the cluster or when an existing server in the cluster is restarted, Integration Server creates a corresponding child task upon server restart. However, Integration Server sometimes does not create the child task for the newly added server or for the server that was restarted. As a result, the complete information for all servers in the cluster is not available on the Scheduler screen.
  This issue is resolved.

- PIE-38404
  Integration Server resumes document retrieval and/or processing for webMethods messaging triggers after package reload or server restart even when the Apply Change Permanently option was selected.
  When using the Integration Server Administrator or the built-in services for suspending document retrieval and/or processing for a webMethods messaging trigger that receives documents from Universal Messaging, Integration Server does not honor the value of the Apply Change Permanently check box. When selected, the Apply Change Permanently check box indicates that Integration Server persists the change in document retrieval or document processing across package reloads and server restart. However, Integration Server reverts the state and enables retrieval and/or processing of documents when Integration Server restarts or when a package is reloaded.
  This issue is resolved.

- PIE-38473
  While creating an Enterprise Gateway Server port that uses the HTTPS protocol, the 'Use JSSE' option is not available in the Edit Enterprise Gateway Server Configuration screen in Integration Server Administrator.
  This issue is resolved. The 'Use JSSE' option is now available in the Edit Enterprise Gateway Server Configuration screen while creating an Enterprise Gateway Server port that uses the HTTPS protocol.

- PIE-38526
  When creating an Internal Server port that uses the HTTPS protocol, the "Use JSSE" option is not available in the Edit Internal Server Configuration screen in Integration Server Administrator.

This issue is resolved. The Edit Internal Server Configuration screen now includes the “Use JSSE” option when the selected protocol is HTTPS.

- PIE-38530
  An HTTPS port that uses JSSE fails to start when a keystore is specified.
  If Use JSSE is set to Yes for an HTTPS port and the key alias password is different from the password used for the keystore, the port does not start.
  This issue is now resolved.

- PIE-38536
  The pub.client:smtp service completes successfully even though the service finishes with errors.
  Improper exception handling allowed the pub.client:smtp service to execute successfully even though the service encountered errors. This could lead to missing MIME parts in the email sent by the service.
  Now, the pub.client:smtp service includes proper exception handling, which prevents successful completion of the service when the service encounters errors.

- PIE-38557
  When entering quiesce mode, Integration Server writes messages to the client-side queue because Integration Server disables connection aliases before disabling packages.
  When Integration Server enters quiesce mode, Integration Server disables the webMethods messaging connection aliases and JMS connection aliases before disabling packages. As a result, services that publish messages might execute after the needed connection alias is disabled, causing messages to be written to the client-side queue.
  Now, when entering quiesce mode, Integration Server disables packages before disabling connection aliases. This will prevent new messages from being written to the client-side queue as Integration Server enters quiesce mode.

- PIE-38660
  The jcode utility does not scan the jar files present in the static folders of the packages.
  The jar files present in the static folders of the packages (package's code\jars\static folders) were not being scanned and added to the classpath by the jcode utility.
  This issue is resolved.

- PIE-38881
  When migrating from Integration Sever 8.2 or later to a newer version of Integration Server, the migration utility overwrites server configuration parameter values set during an earlier migration.
  If you migrated Integration Server from version 7.x to version 8.2 or later and the later version introduces new behavior for existing functionality, the migration utility prompts you to choose the new behavior or the existing behavior and then sets a server configuration parameter in the server.cnf accordingly. When you migrate Integration Server from 8.2 or later to a newer version of Integration Server, the migration utility prompts to use the new or existing behavior for existing functionality that was changed in the new version. However, the Integration Server migration utility incorrectly overwrites some of the settings that were already set during earlier migrations.
  This issue is now resolved. Integration Server migration utility retains previously selected behavior and sets server configuration parameters for behavior introduced in the new version only.

- PIE-38921
  Setting the watt.server.db.connectionCache server configuration parameter to server increases the number of database connections.
  Setting the watt.server.db.connectionCache server configuration parameter to server tells Integration Server to maintain a pool of connections for each database. However, when a connection became stale, a new connection was getting created without the stale connection getting closed. This increased the number of connections even though the pool reached its maximum number of connections.
  This issue is resolved. Integration Server now closes all the stale connections before creating new connections.

6.0 Documentation Changes

This section describes significant changes to the documentation, such as the addition, relocation, or removal of product guides, online help, chapters, or other major content. A release is listed in this section only if changes occurred in that release.

Release 10.7

- The *webMethods Integration Server Administrator's Guide* contains a new chapter titled "Introduction to the New webMethods Integration Server Administrator".
- The *webMethods Integration Server Built-In Services Reference* contains a new chapter titled "Alert Folder".
- The "Using Command Central to Manage Integration Server" chapter in the *webMethods Integration Server Administrator's Guide* has the following changes.
  - "An Overview of Command Central" topic has been added.
  - Deleted topics:
    - Monitoring JMS Triggers Using the Integration Server Instance
    - Searching for JMS Triggers Using Integration Server Instance
    - Enabling, Disabling and Suspending a JMS Trigger
    - Monitoring Global Controls for JMS Triggers
    - Associating Connection Pools with Functional Aliases
    - Managing Integration Server Groups
    - Managing Scheduled User Tasks
    - Managing ACLs
- The *webMethods Integration Agent Administrator's Guide* is no longer delivered with the webMethods Microservice Runtime and webMethods Integration Server documentation sets as webMethods Integration Agent is no longer available beginning with the 10.7 release. Microservices Runtime can

be used in place of Integration Agent.

- The *REST Developer's Guide* is no longer delivered with the webMethods Integration Server documentation set as the contents of the guide are now available in the *webMethods Service Development Help*.

- A new topic, "Setting Two-Way SSL Communication" is added to the guide *Configuring On-Premise Integration Servers for webMethods Cloud*.

- Descriptions of the server configuration parameters watt.server.jca.connectionPool.threadInterrupt.waitTime and watt.server.jca.connectionPool.threadInterrupter.sleepTime have been removed from the *webMethods Integration Server Administrator's Guide* because they are covered in the *webMethods Adapter Runtime User's Guide.* The description of the watt.server.jca.connectionPool.createConnection.interrupt.waitTime parameter has been removed from the *webMethods Integration Server Administrator's Guide* because the parameter is obsolete, having been replaced by watt.server.jca.connectionPool.threadInterrupt.waitTime and watt.server.jca.connectionPool.threadInterrupter.sleepTime paramters.

Release 10.5

- MQTT feature documentation has been added to the following guides in the Integration Server documentation set: *webMethods Integration Server Administrator's Guide* includes the new chapters "Configuring Integration Server for MQTT Messaging" and "Managing MQTT Triggers"; *webMethods Service Development Help* includes the new chapter "Working with MQTT Triggers". The *webMethods Integration Server Built-In Services Reference* contains the new chapter "MQTT folder".

- The *webMethods Integration Server Administrator's Guide* contains a new chapter titled "Setting Up SSL Session Logging".

- The *webMethods Integration Server Administrator's Guide* contains a new chapter titled "Configuring Integration Server for HTTP Compression".

- The "Configuring Integration Server to Work with Servers Running HTTP 1.0 and Above" section has been removed from the *webMethods Integration Server Administrator's Guide*.

- The pub.xml:getXMLNodeIterator service description now includes information about how the NodeIterator traverses nodes when using search criteria.

Release 10.4

- The *Configuring the VCS Integration Feature* guide is no longer available, and the chapter entitled "Using the VCS Integration Feature to Check Elements in and Out of a VCS" is no longer part of the *webMethods Service Development Help*. The guide and related chapter explained how to use the functionality delivered in the WmVCS package to integrate Integration Server elements with a

version control system. However, the WmVCS package is no longer delivered with Integration Server.

Release 10.3

- Information about configuring the Universal Messaging client log has been moved from the *webMethods Audit Logging Guide* to the *webMethods Integration Server Administrator's Guide*.

Release 10.2

- The *Developing Microservices with webMethods Microservices Container* guide has been renamed *Developing Microservices with webMethods Microservices Runtime* to reflect the product name change from webMethods Microservices Container to webMethods Microservices Runtime.

Release 10.3

- The *webMethods Integration Server Clustering Guide* has been updated to include information about stateless clustering.

Release 10.1

- The *Web Applications Developer's Guide* is no longer available. This guide explained how to use the WmTomcat package to incorporate web applications into the Integration Server environment. As of Integration Server 10.1, the WmTomcat package has been removed from the product.

7.0 Terminology Changes

A release is listed in this section only if changes occurred in that release.

Release 10.3

Old Term	New Term
Named object	Durable subscription or durable. As of Universal Messaging version 10.2, the term *named object* has been replaced with *durable subscription* or its abbreviated form *durable*. A serial webMethods messaging trigger corresponds to a serial durable subscription (or the abbreviated serial durable) on Universal Messaging. A concurrent webMethods messaging trigger corresponds to a shared durable subscription (or the abbreviated shared durable) on Universal Messaging.

Release 10.2

Old Term	New Term
webMethods Microservices Container	webMethods Microservices Runtime

8.0 Added, Removed, Deprecated, or Changed Items

This section lists functionality, controls, portlets, properties, or other items that have been added, removed, deprecated, or changed. A release is listed in this section only if changes occurred in that release.

Release 10.7

Added Item	Description
New Integration Server Administrator	Available through the Try the New Administration Console link at the top of the DSP-based Integration Server Administrator, the new administration console includes: • Simplified, graphically rich, tab-based interface with improved accessibility, simpler debugging tools, and server controls. • An interactive Dashboard that provides visually detailed monitor system status, health, usage patterns and overall performance of Integration Server including JVM, Usage, Services, and API-related metrics. • Alerting and notification framework to notify users about alerts generated in the Integration Server such as password expiry, certificate expiry, and configuration changes that require a server restart.
TLS v1.3	Integration Server supports TLS v1.3 for secure inbound and outbound connections that use JSSE. Integration Server provides support for TLS v1.3 through use of OpenJSSE.
Restart notification	Integration Server Administrator now displays the message “Restart is required for changes to take effect.” if server configuration changes were made that require a restart to take effect
SSL support with MQTT server	Integration Server now supports SSL for connections to the MQTT server.
TLS v1.1 and v1.2 support for email ports and outbound SMTP connections	Integration Server uses the JSSE library to provide TLS v1.1 and v1.2 support for incoming connections on e-mail ports and outbound SMTP connections. If your Integration Server uses OpenJSSE, TLS v1.3 is also supported.

Added Item	Description
Advanced SFTP client in Integration Server	Integration Server includes an advanced SFTP client, which provides new configuration properties and additional Key Exchange Algorithms, Machine Access Code (MAC) algorithms, and Ciphers. This enhanced support is available as a new version (version 2) along with the previous version (version 1) in the SFTP Server Alias settings screen. The preferred key exchange algorithms, ciphers, and MAC algorithms can be configured from the Integration Server Administrator UI.
Support for OAuth 2.0 compliant IMAP e-mail port in Integration Server.	To support OAuth 2.0 for IMAP e-mail ports, the Email Client Listener Configuration includes fields that allow you to enter the OAuth credentials issued by the OAuth Server to Integration Server. With these credentials, you can get the authorization code, which in turn is used by Integration Server to obtain the Access Token that is required to access the mailbox on the configured mail server.
Support for users' email address in LDAP directory configuration for Integration Server.	The "User Email" field is added to the LDAP Directory Settings page, which allows you to specify the name of the attribute that is used to store users' email addresses.
Logging for JNDI	New Trace level logging messages for the JNDI objects returned from the JNDI provider, including messages that provide the returned object, object type, and whether the object was retrieved from the JNDI provider or from cache. The new log messages are visible when the logging level of the facility code 0135 JNDI client configuration is set to Trace.

Added Item	Description
Retention of the protocol buffer definition for a publishable document type that uses protocol buffer encoding.	When you save a publishable document type for which protocol buffers is the encoding type, Integration Server creates a protocol buffer definition. The protocol buffer definition represents the structure and content of the document type as a protocol buffer, including the names, types, and attributes for the fields. Integration Server saves the human-readable protocol definition file named pdt.proto in the same location as the node.ndf for the publishable document type. Previously, Integration Server did not retain the protocol buffer definition.
Default HTTPS port	Integration Server now includes a default HTTPS port with the port alias DefaultSecure. During installation and instance creation, Integration Server automatically creates the diagnostic secure port at 5543. However, you can specify a different value during installation and instance creation.
Truststore alias for JVM truststore	Integration Server includes a default truststore alias named DEFAULT_JVM_TRUSTSTORE for the JVM's default truststore <JVM>/lib/security/cacerts
Secondary truststore alias	Integration Server now allows specifying secondary truststore as part of the truststore alias definition.
JVM Settings on the Server > Certificates page	Keystore alias and truststore alias to use with javax.net.ssl properties can be set on the Server > Certificates page instead of via the watt.server.ssl.keyStoreAlias and watt.server.ssl.trustStoreAlias parameters.
XML Schema Definition Language 1.1 support	Integration Server now provides support for many of the features introduced in XML Schema 1.1. A detailed list of supported features and unsupported features can be found in the *webMethods Service Development Help.*

Added Item	Description
Prefetch cache support for JMS triggers	JMS triggers that receive messages from Universal Messaging can now use a prefetch, or consumer, cache. Each time the JMS trigger requests more messages from Universal Messaging, the JMS trigger can retrieve multiple messages which are then placed in a cache. The JMS trigger processes messages from the cache instead of requesting messages from Universal Messaging.
New flag, recordIdentifierPre105Version, added in the WmFlatFile package	recordIdentifierPre105Version: Determines whether the pub.flatFile:convertToString service exhibits the same behavior as the version before the 10.5 release when recordIdentifierCheck is set to true.
Integration Server provides the configurations to make a REST API descriptor as a deployable asset in a cloud environment.	Integration Server provides the following hidden configurations parameters. You can use these parameters on an Integration Server that is running on a cloud environment, to configure the host address while building Swagger or WSDL URLs. watt.server.host.protocol: Specifies the communication protocol used by the Integration Server. watt.server.host.virtualHost: Specifies the target host machine where the asset is deployed. watt.server.host.port: Specifies the port of the host machine where the asset is deployed. watt.server.host.prefix: Specifies the URL prefix that is used to access the asset.
HTTP HEAD method support	Integration Server now supports HTTP HEAD method for the REST V2 resources.
Support for selection of REST operations for Swagger based consumer REST API descriptors	Integration Server supports to choose REST operations while creating a consumer REST API descriptor using a Swagger document.
Two-way SSL communication support	Integration Server supports two-way SSL communication between the on-premise Integration Server and webMethods Cloud.
Support for OpenAPI based provider REST API descriptors	Integration Server supports to create a REST API descriptor using an OpenAPI document that conforms to the OpenAPI specification 3.0.x.

Added Item	Description
MySQL community edition 8.x version support	Integration Server supports to use MySQL community edition 8.x database driver.
PostgreSQL 10.11 database support	Integration Server supports to use PostgreSQL database 10.11 version.
Tibero database support	Integration Server supports the use of the Tibero JDBC driver to connect to a Tibero database instance.
Caching of well-known DTDs for XHTML	When an XML file references the W3C definition for the XHTML DTD, Integration Server encountered a delay and overhead for accessing a W3C server. To avoid delays encountered while accessing the W3C definition for the XHTML DTD, Integration Server now caches the definitions for the well-known XHTML DTDs.
Custom property named $coderType for a publishable document type	For instances of a publishable document type configured to publish IData-encoded documents to Universal Messaging, when this property is set to idata_xml_bytes, instructs the publishing Integration Server to encode the IData message as XML using IDataXMLCoder. The same property also instructs the receiving Integration Server (that is, the server on which the webMethods messaging trigger resides) to decode the message as XML to IData.
Custom property named $coderType for JMSMessage/properties	When using pub.jms:send to send a message to Universal Messaging, Integration Server includes a custom property to indicate that the message should be encoded as XML and decoded from XML instead of as a byte array. When invoking the pub.jms:send add the following custom property and value to the JMSMessage/properties document in the pipeline: $coderType = idata_xml_bytes. The presence of the custom property $coderType and the value " idata_xml_bytes" instructs the sending Integration Server to encode the IData message as XML using IDataXMLCoder. The same property and value also instruct the receiving Integration Server to decode the JMS message as XML to IData.

Added Item	Description
Environment variables can be used as values of substitution variables during ACDL- based deployment.	When deploying to Cloud Container or any other ACDL-based deployment, the value of an environment variable can be used as the substitution value.
New statistic on Server > Statistics page	The Server > Statistics page in Integration Server Administrator includes Started Requests statistic which displays the number of started services requests during the last polling interval.
Kerberos authentication for outbound connections to a database through JDBC connection pools.	Integration Server supports the use of Kerberos authentication for outbound connections to a database through JDBC connection pools. Integration Server provides support for using the driver's Kerberos re-authentication feature with any supported database. This feature introduces the new database URL parameter to the JDBC connection pool: ReAuthenticateUser=<username>.
Sending of pub.publish.notification:error messages can be suppressed.	Integration Server generates pub.publish.notification:error messages (documents) when a webMethods messaging trigger encounters an error or exception condition during processing. In turn the pub.publish.notification:error messages generate Adapter::Error events. These messages and events may be unwanted and can now be suppressed using the watt.server.messaging.deliverNotificationErrors server configuration parameter.
JSON primitive type support	JSON can represent four primitive types (strings, numbers, Booleans, and null) at the root level. Integration Server now supports these types a valid JSON when converting to and from JSON.
Parquet support	Integration Server provides the WmParquet package which contains built-in services for reading Apache Parquet files and converting them to IData as well as writing IData to a Parquet file.
pub.mime* services support large payloads	The pub.mime services have been enhanced to support large payloads and to ensure that large payloads work. This enables the services to process data that exceeds the heap size.

Changed Item	Description
A web service connector for a web service descriptor that uses the current web services stack now supports the use of a custom HTTP Host header.	Previously, a web service connector that is part of a consumer web service descriptor that used the current web services stack (that is, the Pre8.2 compatibility mode property is set to false) could not use a custom Host header. Integration Server would overwrite any value supplied for Host in the transportHeaders input parameter for the web service connector. Now, a custom Host header value can be supplied to the transportHeaders input parameter for the web service connector and Integration Server will not overwrite it.
Suspending a webMethods messaging trigger that receives documents from Universal Messaging is now a pause in the subscription.	Prior to Integration Server version 10.7, suspending a webMethods messaging trigger that received messages from Universal Messaging resulted in Integration Server stopping and closing the subscription. Integration Server rolled back all of the documents that had been retrieved but not acknowledged to Universal Messaging, including unprocessed documents in the trigger queue and documents currently being processed. This led to duplicates. Pausing the subscription will not cause any documents to be rolled back, thus introducing duplicates. Any messages already received by the trigger will be processed as long as the trigger is enabled and processing is not suspended.

Changed Item	Description
How Integration Server chooses a media type for a response when the request Accept header with multiple media types and none of the media types has a "q" parameter.	Previously, when a request included an Accept header with multiple media types and none of the media types had a factor weighting ("q" parameter), Integration Server attempted to return the response using the last media type in the Accept header. The HTTP standard does not specify how to handle multiple media types with no "q" parameter, so this not incorrect. However, it is intuitive to assume that in this scenario, the first media type in the Accept header is the preferred one. Now, when a request includes an Accept header with multiple media types and none of the media types has a "q" parameter, Integration Server now attempts to respond with the first media type in the Accept header. If there is no content handler defined for the first media type, it attempts to respond with the second media type from the Accept header. This repeats until Integration Server finds a media type with which it can respond in the Accept header. If none are found, that is, there is not a content handler registered for any of the media types in the Accept header, Integration Server will respond with the text/html media type.
Exception handling during an implicit or explicit transaction.	If an error occurs during a transaction, Integration Server propagates an exception for this local transaction and does not proceed with service execution. Previously, Integration Server ignored the exception and executed the next step in the flow service. While the new behavior is the correct behavior, in some circumstances the proper handling of the exception for the transaction service may be undesirable. Integration Serve provides the watt.server.transaction.ignore.exception parameter to control this behavior.

Changed Item	Description
The addition of the '$httpMethod' variable to the input pipeline of a service signature is configurable	Previously, by default, Integration Server adds the '$httpMethod' variable to the input pipeline of a service signature while processing REST requests. Now, you can configure the addition of this property to the input pipeline using the "watt.server.rest.addHTTPMethodToInputPipeline" server configuration parameter. If you set this parameter to 'true', Integration Server adds the '$httpMethod' input variable with the requested HTTP method in the input pipeline while processing a REST request. The default value is 'false'.
Statistics log format	The default format of the statistics log is now csv (comma-separated values). To change the format to hexadecimal, the previous default format, set watt.server.stats.logfile.csv to false.
Changed statistic names on Server > Statistics page	Names of statistics on the Server > Statistics page in Integration Server Administrator have been changed as follows: Completed Req's is now Completed Requests Req's per minute is now Requests in last polling interval The Ended column is now named Completed.
Debug logging for 0088 SOAP facility	In the server log, SOAP request, SOAP response, and SOAP exception related log entries include the Thread Id and Session Age. This can help with debugging.
webMethods messaging trigger thread usage	The thread used for running the webMethods messaging trigger and listening for messages from Universal Messaging is not considered to be a document processing thread. That is, the thread that runs the trigger does not count towards the usage limit set by the Maximum Threads for document processing. Previously, the thread used to run the trigger and listen for messages counted toward the thread usage limit set by the Maximum Threads for document processing.

Changed Item	Description
Improved debug logging for OAuth authorization problems.	When an access token from an external authorization server is rejected, information about the rejection is written to the server log. Set the OAuth logging facility (0010) to Debug to see the messages in the server log.
Docker images created for Microservices Runtime or Integration Sever can be deployed to OpenShift.	A Docker image for Microservices Runtime or Integration Server created with the is_container.bat/sh script can be used with OpenShift by setting the -Dtarget.configuration parameter to OpenShift when executing the createDockerfile, createPackageDockerfile or createLeanDockerfile command.
Server log messages include thread ID	To improve troubleshooting, especially under heavy load, server log messages include the ID of the thread performing the activity that generated the log message.
Security log enhancements	Integration Server now includes the following changes for the security log: - Logging anonymous authentication requests. - Logging the IP address of the client that initiated the request instead of the proxy server address.
The -Dimage.name parameter now required when creating a Dockerfile for an Integration Server that runs on Windows	When the is_container.bat script is being used to create a Dockerfile for an Integration Server that runs on Windows the createDockerfile command must specify the -Dimage.name parameter. This is because Microsoft now provides Windows image tags per OS version instead of a "latest" base image.
TLSv1.0	TLSv1.0 is no longer a default enabled protocol for inbound or outbound connections.
JSON Schema enhancements	The enhanced JSON schema feature provides lucid and easy to understand message while creating and validating JSON documents using the built-in json schema services. With the enhanced support, JSON document types can be validated when used in the flow service signature at runtime.

Release 10.5

Added Item	Description

Added Item	Description
Administrator API	Integration Server provides an administrative API that can be used to perform administrative actions such as restarting Integration Server as well as creating, retrieving, updating, and deleting assets on Integration Server. The Administrator API uses the REST architectural style.
GraphQL Dataloader support	Dataloader is a utility that improves the performance of your GraphQL query. Dataloader supports batching and caching functional capabilities. When you create a Dataloader, Integration Server generates a loader service and a document type for keys. You can specify the field(s) in the key document for which you want to fetch the data from the data source. A loader service loads the data for the list of keys and returns a list of values.
MQTT Support	Integration Server can be used to publish MQTT messages to and receive MQTT messages from an MQTT server. Use the pub.mqtt:publish service to publish an MQTT message to an MQTT server. Use MQTT triggers to create topic subscriptions and then receive and process MQTT messages. Integration Server supports MQTT version 3.1.1
SSL session logging	Integration Server SSL session log contains SSL session information for inbound connections in JSON format. It contains information related to the cipher suite used, version of protocol, and client details along with server and session creation details. Using this information, you can analyze the details of a successful SSL handshake.
HTTP request or response compression	Integration Server supports data compression for both HTTP requests and HTTP responses. Integration Server as HTTP client supports data compression before sending the HTTP request and can also instruct the HTTP server to compress the data before responding back to the client.
SSL support for third party JNDI provider	Integration Server provides options to configure SSL communication between Integration Server and the JNDI provider.

Added Item	Description
HTTP/S port can be stateless	Integration Server provides the ability to make Integration Server HTTP and HTTPS ports stateless. A stateless port will not maintain any sessions or provide session IDs for requests received by the port.
SAML2 support at transport level	Integration Server adds support for sending SAML2 tokens in a custom HTTP header "wmIS-SAML2-Assertion", making it possible to use SAML2 tokens with all types of services and enabling integration with other security providers.
Service blacklist	Integration Server provides the ability to block the invocation of services through use of a service blacklist. An attempt to invoke a blacklisted service results in an Access Denied error.
Support for Outbound interceptor	The support for HTTP interceptors is enhanced with the addition of Outbound interceptor that is invoked when Integration Server is an HTTP client.
Additional support for JSON schemas	Integration Server now supports additional features when generating JSON document types, such as support for readOnly and writeOnly from the JSON schema draft 7 specification.
Support for Kerberos authentication at the HTTP/S transport level for consumer web services.	Integration Server now supports Kerberos authentication at the HTTP/S transport level for consumer web services.
Support for content-security policy	Two new server configuration properties: watt.server.http.Content-Security-Policy and watt.server.http.X-Permitted-Cross-Domain-Policies, enable you to secure Integration Server against attacks such as Cross Site Scripting (XSS) and data injection.
Server log archive file name	The server.log archive file name format for a size-based rotation changed from server.log_yyyyMMdd_HHmmssSSSZ, (where yyyyMMdd_HHmmssSSSZ is the date and time the log file was created) to server.log.yyyyMMdd.n.

Changed Item	Description
JVM version requirement for package release and the subscribing Integration Server	Previously, Integration Server installed but did not activate a package when the version of the JVM used by Integration Server was less than the minimum JVM version required for the package. This included cases where the JVM used by Integration Server and the JVM version required by the package were the same major version. For example, if the minimum JVM version required for a package is 1.8.0_23 and the Integration Server JVM version is 1.8.0_12, Integration Server installed the package but did not activate it. Now, the Integration Server on which the package is installed must run in a JVM with a major version that is the same or higher than the JVM version required by the package.
Support for accommodating clock skew between a Java Web Token (JWT) issuer and Integration Server	A variation between the JWT issuer server clock and the Integration Server clock can cause valid JWT tokens to be rejected. A new setting in the IS Administrator enables you to accommodate unavoidable and authentic variations. You can use this setting to define the permissible limits for variations between JWT issuer server clocks and the Integration Server clock.
Support for configuring and using Time To Live (TTL) settings for the UsernameToken used in the UsernameToken security policy	Two new properties, Username Token TTL and Username Token Future TTL, and their corresponding server configuration properties: watt.server.ws.security.usernameTokenTTL and watt.server.ws.security.usernameTokenFutureTTL, enable you to accommodate time differences between when a SOAP username token is created and when it reaches Integration Server.

Release 10.4

Added Item	Description
GraphQL support	GraphQL is a query language designed to build client applications by providing a flexible syntax and system for describing their data requirements and interactions. Using GraphQL service, you can query a specific data to the server and get the response in a predictable way. Integration Server acts as a GraphQL service provider and supports GraphQL version 9.X.
JSON Schema support	Integration Server provides native support for JSON Schema in which a developer can create a JSON document type from a JSON schema and then use the JSON document type to validate a JSOON payload.
Enhanced support for Swagger 2.0 standard while creating REST APIs	Integration Server supports Swagger constructs like tags, external docs and basic security definitions while creating APIs using resource first or swagger first approaches. Integration Server supports Swagger files that reference other files.
Support for Universal Messaging horizontal scalability	Universal Messaging horizontal scalability feature can be used with webMethods messaging connection alias that connects to Universal Messaging.
Enhancements for Enterprise Gateway Server while communicating with ICAP server for virus scanning or content filtering.	Enterprise Gateway supports both REQMOD and RESPMOD methods based on the ICAP server response to the OPTIONS method. Enterprise Gateway Server sends a header named, X-wMUUID with every outbound ICAP request to differentiate one scan from another. The Enterprise Gateway Server returns success if ICAP server status code is 200 and HTTP status code is within the range of 200-300. Enterprise Gateway Server returns failure if ICAP server status code is greater than or equal to 300.
Support for AMQP protocol using QPID JMS client libraries	Integration Server is certified to communicate with JMS providers that support AMQP protocol using QPID JMS client libraries. Integration Server supports the "Qpid JMS AMQP 0-x 6.3.3" JMS client libraries.

Added Item	Description
Lifecycle management settings for sockets.	Integration Server now includes server configuration parameters to control how long a socket is kept in the pool (watt.net.clientKeepaliveAgingLimit) and how many times it can be used (watt.net.clientKeepaliveUsageLimit). If either limit is exceeded, Integration Server will not return a socket to the pool. The new parameters are tuning parameters that will need to be adjusted based on use case and usage patterns.

Removed Item	Replacement, if any
WmVCS package and VCS Integration feature	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system directly from Designer.

Deprecated Item	Replacement, if any
Web services implementation introduced in Integration Server 7.1.	The current web services stack. Web service descriptors can be changed to work with the web services stack by changing the Pre-8.2 compatibility mode property to false. A bulk change of web service descriptors can be accomplished using the built-in service pub.utils.ws:setCompatibilityModeFalse. Deprecation of the web services implementation introduced in Integration Server 7.1 results in the deprecation of the following: • The ability to run in pre-8.2 compatibility mode. • The Pre-8.2 compatibility mode property. • The Integration Server WS-Security facility which provides support for WS-Security by associating built-in WS-Security handlers to web service descriptors. Note: Securing web services using WS-SecurityPolicy is not deprecated. For information about how WS-SecurityPolicy compares to the Integration Server WS-Security facility, including a comparison of policies provided for each, see the *Web Services Developer's Guide*. • Any behavior specific to the web services implementation introduced in 7.1. For more information about the differences between the 7.1 web services implementation and the web services stack, see the "About Pre-8.2 Compatibility Mode" section in the "Working with Web Services" chapter available in the *Web Services Developer's Guide* or *webMethods Service Development Help*.

Changed Item	Description
The Fail-Fast Mode Enabled and Currently In Fail-Fast options do not appear when editing the CentralUsers functional alias.	Fail-fast cannot be used with the CentralUsers functional alias definition. Consequently, the Fail-Fast Mode Enabled and Currently In Fail-Fast options do not appear when editing the CentralUsers functional alias.
is_container script contains a parameter for specifying when the default truststore store should be overwritten.	The is_container script now includes the -Dtarget.configuration parameter which must be set to wmcloud when creating a Docker image for an Integration Server that will be lifted and shifted into Integration Cloud. When this parameter is set to wmcloud, the is_container script overwrites the default truststore with the truststore required by Integration Cloud.
Allow and Deny Lists for ports	The port access control feature can now specify service URIs that are not part of the Integration Server namespace. Though any service URI can now be included in port access control, Integration Server only validates and enforces web service URIs as part of this change.
File-based audit logs can use character sequence delimiters.	The fields for an entry in a file-based audit log can now be fixed length or character delimited. Using character-delimited entries may reduce the size of the log.
JMSMessageID	When sending a JMS message to Universal Messaging, Integration Server sets the JMSMessageID.
Namespace preservation when decoding an xsd:any element	Integration Server now preserves all namespace declarations when decoding an xsd:any element. Previously, when decoding a SOAP request or response that includes an xsd:any element, Integration Server preserved only the namespace declarations (xmlns attributes) in the top level element of an xsd:any element. An xsd:any element may have any number of nested elements. Integration Server should not remove he namespace declarations in nested elements.
Resource Owner Password Credentials (ROPC) grant type can be used with public clients.	Integration Server now supports use of the ROPC grant type with public clients.

Release 10.3

Added Item	Description
Support for using the --time option (-t) with docker stop command.	The --time option specifies how long Docker waits before killing a container. Use the --time option to specify a waiting time long enough to allow Integration Server to shut down gracefully.
Custom security area for security logging.	The new method com.wm.app.b2b.server.ServerAPI.logSecurity makes it possible for applications to write entries to the security log as logging for the Custom security area.
SAG_IS_AUDIT_STDOUT_LOGGERS environment variable	Specifies the audit loggers to write to the console (STDOUT). When an audit logger writes to STDOUT it is considered an auxiliary output which means the logger will still write to the specified log destination of file or database.
EXTERNALIZE_PACKAGES environment variable	Instructs the Integration Server running in the Docker container to load the packages located in one of the specified directories at startup.
SAG_IS_LICENSE_FILE environment variable	Specifies the license key to be used with an Integration Server running in a Docker container which allows the license key to be changed without rebuilding the Docker image for the Integration Sever.
Support for ICAP preview headers	Integration Server now supports an ICAP preview header that allows the ICAP server to scan the preview content instead of scanning the entire request from Enterprise Gateway.
Support for OAuth grant types Resource Owner Password Credentials and Client Credentials	Integration Server now supports the Resource Owner Password Credentials and Client Credentials grant types.
JMS trigger restart task	If a JMS trigger does not start when a JMS connection alias starts, Integration Server schedules a task to retry starting the JMS triggers and then continues with activities associated with starting the JMS connection alias. The trigger restart task, which runs in its own thread, attempts to restart the failed JMS triggers at a set interval.

Changed Item	Description
Updated messageNumber key for reliableMessagingProperties.	Specifies the message number in the message sequence. In cases where there are several message numbers in the message sequence, you can specify a custom message number.
Serial webMethods messaging triggers that receive documents from Universal Messaging now process documents from all publishers in publication order in a load-balanced fashion.	Previously, a webMethods messaging trigger with serial processing corresponded to a priority named object on a channel on Universal Messaging server. Now, a serial webMethods messaging trigger corresponds to a serial durable subscription (named objects in Universal Messaging are now called durable subscriptions). This provides processing in publishing order for a serial trigger in a cluster or a non-clustered group of Integration Servers. Over time, it allows for the Integration Server in a cluster or non-clustered group to process messages in a load balanced fashion.
Server log entries written to the console now include the identifier “ISSERVER”	To help differentiate server log messages from other messages written to the console, server log entries include the identifier “ISSERVER”. Previously, there was no identifier.
Allowed Grants property for registering clients	When registering clients for use with an Integration Server acting as an authorization server for OAuth, you can specify the grant types a client can use. Previously, all registered clients could use the authorization code grant type and the implicit grant type. All registered clients migrated from an earlier version of Integration Server will have the Authorization Code Grant type and Implicit Grant type selected.
Token endpoint authorization property for global OAuth settings	Specifies whether the token endpoint accepts an existing session or requires credentials for authentication. Previously, the token endpoint service always accepted requests from clients that had an active session on Integration Server and did not provide an option to require authentication every time the client requests a new access token or refreshes an existing access token

Changed Item	Description
Improved visibility into sending messages from client side queue to Broker.	To improve visibility and control of sending messages from the CSQ to the Broker for webMethods messaging, Integration Server introduces the following: The maximum number of retry attempts that Integration Server makes when publishing a message from the CSQ to Broker is now configurable. Previously, the maximum number of retries was 5 and could not be changed. For details about the server configuration parameter watt.server.publish.maxCSQRedeliveryCount, see below. At the debug level for facility 0098 Dispatcher, the Integration Server now logs additional messages that indicate why an attempt to send a message from the CSQ to Broker failed. Existing log entries as well as the newly introduce log entries for facility 0098 Dispatcher now include the message ID and the name of the publishable document type for which the message is an instance document.
Ownership of files in Docker image is changed to non-root user	When running the is_container.sh script to create Docker images, ownership of the files is changed to non-root (sagadmin) user. The optional –Dimage.createUser parameter can be passed to the createDockerfile, createLeanDockerfile, or createPackageDockerfile commands to control the ownership of the files in the Docker image.

Changed Item	Description
Client ID for webMethods messaging triggers that receive message from Universal Messaging	For webMethods messaging triggers created in Integration Server 10.3, the client ID of the trigger is now the trigger's shared durable name which is as follows: *clientPrefix##triggerName* Where *clientPrefix* is the Universal Messaging connection alias used by the trigger, *triggerName* is the fully qualified name of the trigger where periods and colons are replaced by double underscores. Previously, the client ID naming convention for a trigger that receives message from Broker was the same as the client ID naming convention for a trigger that receives messages from Universal Messaging.
JSON payload for invoking a REST v2 resource	Prior to Integration Server 10.3, the JSON payload for a REST request of type POST, PUT, or PATCH needed to include the root element of a document in the service signature. This was necessary even though the Swagger document generated for the REST API descriptor did not include the root document. Beginning with Integration Server 10.3, the request payload should match the Swagger definition – not the service signature. If there is a root element in the service signature input, only the contents of the root element need to be provided in the request payload.

Release 10.2

Added Item	Description
Enhanced messaging logging	Integration Server writes detailed log entries when sending or receiving and processing messages. The detailed logging makes it possible to track an individual message across your Integration Servers from the time the message is published through the time a trigger receives, processes, and acknowledges the message. Enhanced logging includes the introduction of the messaging audit log.

Added Item	Description
WebSocket support	Integration Server enables development of applications using the WebSocket protocol. The WebSocket protocol provides simultaneous two-way communication between a client endpoint and server endpoint over a single TCP connection.
Whitelist filtering	To prevent Integration Server from deserializing untrusted Java objects, Integration Server now performs whitelist filtering and includes a whitelist of classes that can be loaded and deserialized in Integration Server.
Support for default scope for Integration Server acting as the OAuth resource server	When configuring an external authorization server, you can specify a default scope in the Default Scope field.
SSL field on the About page of Integration Server Administrator	To help troubleshoot connectivity issues, the About page of Integration Server Administrator, the Software section now contains an SSL field that displays one of the following: - The JCE Unlimited Strength Jurisdiction Policy File was not found. Please install it. - The JCE Unlimited Strength Jurisdiction Policy File was found
Support for security response headers	To enhance the security of response headers, Integration Server now supports the following HTTP response security headers: − X-Content-Type-Options − X-XSS-Protection − Strict-Transport-Security
Improved support of REST multipart messages	Integration Server provides Java API that enables you to register your own services along with the associated coders for multipart content handler.
Support for strong password polices	To enhance the security, Integration Server now supports stronger password policies for user accounts. To achieve this, Integration Server now covers the following aspects of password security: - Stronger password requirements - Password expiration requirements - Account locking

Added Item	Description
Circuit Breaker	The circuit breaker feature provides the ability to configure a circuit breaker for any service residing on a Microservices Runtime or an appropriately licensed Integration Server.
Any field type support for Digital Event Services	An Object or Object List with a Java wrapper type of UNKNOWN is included in the Digital Event Servers event type definition as an Any or Any[] field. Previously, an Object or Object List field with an UNKNOWN Java wrapper type were not included in the corresponding Digital Event Services type definition. Additionally, IS document types can be used to define the structure and content for an Any field in a document published to Digital Event Services.
Map field type support for Digital Event Services	An empty inline document variable configured to allow unspecified fields in a publishable document type is now represented as Map field in the corresponding Digital Event Services type definition.
Synchronizing document types with Digital Event Services automatically	Document types can be set to automatically synchronize with a corresponding Digital Event Services event type definition in the Digital Event Services repository upon save. Set this property to true for a document type that may be used to define the structure and content for the value of an Any field in a document that is published to Digital Event Services.

Added Item	Description
Migration Utility handles updates for hostname	Prior to release 10.2, if you upgraded your Integration Server by installing the new release on a different machine, you had to update the database components below to reflect the host name of the new machine, as follows: - If you had scheduled tasks that executed on Integration Server on specific machines in your old installation, you had to update database component ISInternal, table IS_USER_TASKS, column TARGET. - If you wanted to be able to resubmit services that ran before you upgraded, you had to update the following: - Database component ProcessAudit, table WMRULEDIST, column SERVID. - Database components IS CoreAudit, ProcessAudit; tables WMDOCUMENT, WMSERVICE, WMSERVICE4X, MSERVICEACTIVITYLOG, WMSERVICEASSOC, WMSERVICECUSTOMFLDS; column SERVERID. - If you wanted to be able to resubmit processes and process steps that ran before you upgraded, you had to update database component ProcessAudit, table PRA_PROCESS_STEP, SERVERID column. The Integration Server migration utility now updates all of these tables for you automatically.

Removed Item	Replacement, if any
No support for host-based name format.	Integration Server does not support the host-based name format to specify the Service Principal Name Format when adding or editing an HTTP(s) port.

Changed Item	Description
updateLanguagePack command for is_instance script.	Provides an option to specify the package and/or feature list to which to apply the language pack installation.
create command for is_instance script.	Provides an option for specifying a default IP address to which to bind ports on the new Integration Sever instance.
Server log entries can be written to the console, a file, or both.	Previously, Integration Server could write server log entries to a file or the console (STDOUT). Now, the -log switch includes the both option to write server log information to the computer screen (STDOUT) and to the destination specified by the watt.debug.logfile parameter.
Docker container for an Integration Server Docker image writes the server log to the console (STDOUT).	When running a Docker image of Integration Server in a Docker container, Integration Server writes the server log to the console as well as to the server.log file. Many container deployment solutions provide the ability to view the console log STDOUT for a container.
Repeating tasks scheduled to execute in the Daylight Savings Time (DST) overlap.	If a task includes an hours mask (e.g., "1:15" for running every date at 0115), then the task will run only once, at the pre-overlap ("first") matching time. However, if a task does not have an hours mask (e.g., "hh:15", meaning run at 15 minutes past each hour), then the task will run at the pre-overlap time (the "first" 1:15), and also at the overlap time (the "second" 1:15). Previously, the absence of an hours mask did not result in the task executing twice.
Diagnostic collector	To help with debugging, the Integration Server diagnostic collector returns information regarding whether or not the JCE Unlimited Strength Jurisdiction Policy File is applied to the JVM in the config\ServerAbout.txt file. Integration Server also adds the following file to the list of security providers returned with the diagnostic data: config\JCESecurityInfo.txt

Changed Item	Description
Digital Event Service type definition	With the introduction of support or Any fields and Map fields for Digital Event Services, the Digital Event Type definition generated for a publishable document type changes in the following ways: ▪ An Object or Object List with a Java wrapper type of UNKNOWN is included in the Digital Event Servers event type definition as an Any or Any[] field. Previously, an Object or Object List field with an UNKNOWN Java wrapper type were not included in the corresponding Digital Event Services type definition ▪ An empty inline document variable in a publishable document type is now represented as Map field in the corresponding Digital Event Services type definition. Previously, an inline document variable became an empty nested event.

Release 10.1

Added Item	Description
SFTP Host Key Checking field in the SFTP User Alias Advanced Settings screen	Allows Integration Server to verify the host key of SFTP server during connection.
Create JMS administered objects on demand.	Integration Server can create administered objects in the JNDI namespace automatically when the lookup for the object fails. This functionality, which is controlled by the Create Administered Objects On Demand option for a JMS connection alias, is available only when Universal Messaging is the JMS provider and the JNDI provider.
Support for using volumes when running an Integration Server image in a Docker container.	Docker volumes can be used for externalizing the contents of the logs directory for the Integration Server instance and the Integration Server profile as well as the config directory for the Integration Server instance running in the Docker container.
The is_container.sh script, which facilitates interaction with Docker, now provides an option for specifying a base image.	When using the createDockerFile or createLeanDockerfile command to create a Docker image for Integration Sever, specify a base image using the argument -Dimage.name.

Added Item	Description
Support for using an Integration Server running in a Docker container as the local development server for the local service development feature in Designer.	When using the createDockerFile or createLeanDockerfile command to create a Docker image for Integration Sever, specify that the Integration Server will be used as the local development server using the argument -Dtarget.configuration=localdev
New flag, useAlternateNameForSegment, added in the WmFlatFile package.	useAlternateNameForSegment: Determines whether the flat file parser uses an alternate name for segments if an alternate name is provided.
Support for RFC 7662, OAuth 2.0 Token Introspection	Access tokens generated by an external authorization server can be used to access resources in Integration Server. Access tokens generated by an Integration Server authorization server can be used to access resources in a resource sever that is not an Integration Server.
Support for RFC 7009, OAuth 2.0 Token Revocation	Access tokens generated by Integration Server can be revoked.
Support for REST V2 resources	Integration Server provides a flexible way of defining resources for REST requests. In the Service Development perspective, a developer can define a REST V2 resource with operations that include the following: • The format of the URL that REST clients must follow when sending requests to Integration Server acting as the REST server. • The HTTP methods supported by the resource operation. • The flow service associated with a resource operation.
Support for JWT authentication	Integration Server provides support for JSON Web Token (JWT) authentication. JWT provides a secured means of exchanging claims between two parties.

Added Item	Description
HTTP Interceptor	Integration Server provides a framework for creating and registering an HTTP interceptor. An HTTP interceptor intercepts all received HTTP requests and outbound HTTP responses. Integration Server makes the raw HTTP request and response, including the HTTP header information, accessible to the HTTP interceptor.
Serial JMS triggers supported with Universal Messaging.	When using a connection factory configured for a durable subscriber, Integration Server now supports the use of serial JMS triggers with Universal Messaging.
Additional client side queue (CSQ) logging.	When facility code 134 is set to debug or trace levels, Integration Server provides additional logging when CSQ errors occur.
Integration Server now supports migration when source and target version are same.	The migration utility can now be used for migration where the source and target versions of Integration Server are the same. This can be useful for data center or machine moves

Removed Item	Replacement, if any
WmTomcat package	Existing web applications based on the WmTomcat package can be moved to webMethods Application Platform.

Changed Item	Description
Archived stats.log file name format.	Integration Server now uses the following naming format for the archived stats.log files: stats.yyymmdd_hhmmss.log where the timestamp indicates the time at which Integration Server archived the log file. Previously, the archived stats.log name did not include the time portion which caused a naming conflict that prevented successful log rotation if Integration Server attempted to rotate the stats.log more than once on the same day.
Manage Destinations check box	The Manage Destinations check box for a JMS connection alias is now named Enable Destination Management with Designer.

Changed Item	Description
Share Client Prefix check box	The Share Client Prefix check box for webMethods messaging connection aliases is now named Client Prefix Is Shared. The field name now includes the text "prevents removal of shared messaging provider objects" in parentheses to clarify the purpose of the option.

Release 10.0

Added Item	Description
Allowed HTTP method per service	For each service, you can identify the HTTP methods that can be used to invoke the service.
Default web service provider endpoint alias	For HTTP and HTTPS, you can identify a provider web service endpoint alias as the default for each protocol. If a provider web service descriptor contains a binder that specifies the default endpoint alias or no alias for the Port alias property, Integration Server uses the information in the default alias when constructing the WSDL for the descriptor and during run-time processing.
Follow the master support for JMS connection aliases	When Universal Messaging is the JMS provider, you can configure the JMS connection alias to follow the master for consumer and producer connections.
JDBC pool configuration when Integration Server is offline	In conjunction with Command Central, Integration Server makes it possible for administrators to change the configuration of JDBC pools when Integration Server is offline.
JMS Trigger Groups	A JMS trigger group is a collection of two or more identical JMS triggers. Using the methods in the com.wm.app.b2b.server.jms.consumer. JMSTriggerGroupFacade class, users can quickly create JMS triggers for a trigger group. This can be especially helpful when needing to create several identical triggers that point different Universal Messaging realms, such as when using the new Universal Messaging round robin cluster feature for JMS.

Added Item	Description
PATCH support	Integration Server offers support for HTTP PATCH method, which is used for partial updates. In Service Development, a developer can specify PATCH support when defining REST resources, Integration Server also supports the PATCH methods in the pub.client:http service as well as with CORS (Cross Origin Resource Sharing) requests.
REST Resource URLs	Integration Server provides a flexible way to define resource URLs for REST APIs. The new capability allows a developer to define a URL format for the service that does not adhere to the folder structure within the package. Additionally, you can define multiple REST resources for a single service with the HTTP methods of your choice.
Retrieving and Setting HTTP Headers	Integration Server provides the ability to get and set HTTP request and response headers easily. To get the details of an HTTP request for a service, you can add a document reference variable named $httpRequest that references the pub.flow:HTTPRequest document type in the input signature of a service. While invoking a service that includes a $httpRequest document reference variable, Integration Server populates the $httpRequest variable with information from the from the HTTP request. To get the details of an HTTP request for a service, you can add a document reference variable named $httpRequest that references the pub.flow:HTTPRequest document type in the input signature of a service. While invoking a service that includes a $httpRequest document reference variable, Integration Server populates the $httpRequest variable with information from the from the HTTP request.
Searching for JMS Triggers	You can search for of filter the displayed triggers using the Search Triggers link on the Settings > Messaging > JMS Trigger Management page. You can search by trigger name, JMS connection alias or destination to which the trigger subscribes.

Added Item	Description
Session ID Masking in Logs	Session IDs in the session and server logs can now be secured by masking the ID. When you enable the option for securing Session IDs, Integration Server uses the asterisk (*) character to mask the Session ID strings in the logs which prevents user who access the log entries from viewing the actual Session IDs.

Changed Item	Description
Format of the <soap:address location> element in the WSDL of a provider web service descriptor.	The <soap:address location> element in the WSDL document generated for a service first provider web service descriptor will be in the following format regardless of specifying either a provider web service endpoint alias or a host and port as the endpoint: http://host_name.port/ws/namespace:descriptorname
Follow the master is configurable per Universal Messaging connection alias	For a Universal Messaging connection alias, you can now enable follow the master on a per connection alias basis. Additionally, you can decide whether message producers and/or consumers follow the master realm server using the “Enable Follow the Master for Producers “and “Enable Follow the Master for Consumers” options respectively. Previously, following the master was either enabled or disabled for all connections.

Changed Item	Description
JMS Trigger Management page	The Settings > Messaging > JMS Trigger Management page includes the following changes: Clicking Search Triggers displays a simple search feature. In the JSM trigger list, columns are re-ordered to put the State column right next to the trigger name. The State column values can now be “Enabled”, “Disabled”, or “Suspended”. Previously, the possible State values were “Yes”, “No”, “Suspended”. The Status column is new. Possible values are “Running” or “Not Running” followed up by an explanation, such as “Not Running (trigger disabled)”. The SOAP-JMS triggers list is displayed only if Integration Server contains a loaded package with a SOAP-JMS trigger.
Maximum Entries in Cache field	For a distributed cache, the Maximum Entries in Cache value must be set and must be set to a value greater than 0.
Universal Messaging connection alias Client Authentication	The Client Authentication option named SSL is now named Certificate Based, which is more accurate.

Release 9.12

Added Item	Description
IS_DES_CONNECTION messaging connection alias	A predefined messaging connection alias used to establish a connection with a Universal Messaging server for purposes of sending and receiving events with Digital Event Services.
Kill All Sessions Except Your Session option in the Server > Statistics > Sessions screen	Use the Kill All Sessions Except Your Session option in the Current Sessions Screen to kill all the sessions except the session you are currently running.
is_container scripts to create Integration Server images for use in Docker containers	Use the is_container scripts, created by the Integration Server installation, to build Integration Server images for use in Docker containers. You can access the scripts from the following location: Integration Server_directory/docker.

Added Item	Description
Caches and cache managers can be edited while in quiesce mode	Public and system cache managers and caches can be edited when Integration Server runs in quiesce mode.
Public cache can be used for service results caching.	A local or distributed public cache can now be used for service results caching. Previously, Integration Server stored cached service results in the ServiceResults system cache which is part of the SoftwareAG.IS.Services system cache manager.

Changed Item	Description
FTPS ports that use the HTTPS protocol can now be configured using the “Use JSSE” option.	FTPS ports that support TLS 1.1 or TLS 1.2 protocol can now be configured using the “Use JSSE” option in the Edit FTPS Port Configuration screen of Integration Server Administrator. Previously, this option was not available.
Allowed cipher suites can be specified in a file.	To make it easier to specify a long list of cipher suites for use with inbound and outbound SSL connections, Integration Server now allows specifying a file as the value for the cipher suite server configuration properties watt.net.ssl.client.cipherSuiteList, watt.net.ssl.server.cipherSuiteList, watt.net.jsse.client.enabledCipherSuiteList, watt.net.jsse.server.enabledCipherSuiteList.

Changed Item	Description
Some screens in Integration Server Administrator are disabled if webMethods API Gateway is installed on Integration Server.	The following screens under Security screens are disabled: • Enterprise Gateway Rules • Enterprise Gateway Rules > Default Alert Options • Enterprise Gateway Rules > Default Alert Options > Edit • Enterprise Gateway Rules > Denial of Service Options • Enterprise Gateway Rules > Denial of Service Options > Configure Global Denial of Service • Enterprise Gateway Rules > Denial of Service Options > Configure Denial of Service by IP Address • Enterprise Gateway Rules > Mobile Application Protection Options • Enterprise Gateway Rules > Mobile Application Protection Options > Device Types > Edit • Enterprise Gateway Rules > Mobile Application Protection Options > Mobile Applications > Edit • Enterprise Gateway Rules > Rules > Create The following screens under Security > Ports > Add Port are disabled: • Edit Enterprise Gateway Server Configuration • Edit Internal Server Configuration

Release 9.10

Added Item	Description

Added Item	Description
Configure the allowed protocols for JSSE on a per port basis	Integration Server now provides the ability to specify the allowed protocols for JSSE on a per port basis.
Out of the box support for Content-Type application/xml.	Integration Server content handler named ContentHandler_XML now includes support for the content type application/xml.
Kerberos authentication for outbound service requests.	The pub.client:http service now includes Kerberos-related parameters that you can use to acquire a Kerberos ticket for inclusion in the outbound service request.
Java API Classes for working with XMLData in Java services	A set of classes in com.wm.data for manipulating XMLData in Java services. For a list of classes, see the Added APIs for the Release 9.10 in section 11.0 Added, Removed, Deprecated, or Changed APIs
Universal Messaging connection alias can now be configured to connect to an SSL-enabled port on a Universal Messaging realm server	When configuring a Universal Messaging connection alias, you can specify SSL as the client authentication and provide the truststore and keystore information needed for Integration Server to establish an SSL connection with the Universal Messaging realm server. Previously, to establish a secure connection to the Universal Messaging realm server, you needed to configure the javax.net.ssl properties in the JVM used by Integration Server.
A queue on Universal Messaging can be used as the audit logging queue	Integration Server provides the option of using a Universal Messaging queue instead of the internal queue as the audit logging queue on a per logger basis.
Follow the master for webMethods messaging connections to Universal Messaging server	Integration Server now uses the Universal Messaging client setting “follow the master” which indicates that the client session always connects to the master realm server. The follow the master behavior takes precedence over the behavior indicated through the use of a comma-separated or semicolon-separated list of realm servers in the Realm URL field for a Universal Messaging connection alias.

Removed Item	Replacement, if any
EventBus predefined JMS connection alias	The EventBus alias is no longer available as a predefined JMS connection alias. That is, new Integration Server installations will not include this predefined JMS connection alias. However, if you migrated to Integration Server version 9.10 or later from an earlier version, then you might still have the EventBus alias.
EventBusJndiProvider predefined JNDI provider alias	The EventBusJndiProvider alias is no longer available as a predefined JNDI provider alias. That is, new Integration Server installations will not include this predefined JNDI provider alias. However, if you migrated to Integration Server version 9.10 or later from an earlier version, then you might still have the EventBusJndiProvider alias.

Deprecated Item	Replacement, if any
WmTomcat Package	Existing web applications based on the WmTomcat package can be moved to webMethods Application Platform.

Changed Item	Description
Client queue window size	The Universal Messaging window size for the client queue that corresponds to webMethods messaging trigger is now set to the sum of the Capacity and Max execution threads trigger properties. Previously, Universal Messaging determined the window size.

Changed Item	Description
Filters in a webMethods messaging trigger must be identical if the conditions subscribe to the same publishable document type.	If more than one condition in the webMethods messaging trigger specifies the same publishable document type and the trigger receives messages from Universal Messaging, the provider filters must be identical in each condition but the local filters can be different. Specifically, the contents of the Provider Filter (UM) column must be identical for each condition that subscribes to the publishable document type. The contents of the Filter column can be different. When using pub.trigger:createTrigger to create a trigger, the contents of *conditions/messageTypeFilterPairs/providerFilter* must be the same for each condition. The contents of *conditions/messageTypeFilterPairs/filter* can be different for each condition. The filter requirement is unchanged for a webMethods messaging trigger that receives messages from Broker. That is, if more than one condition in the webMethods messaging trigger specifies the same publishable document type and the trigger receives messages from the Broker, the filters in the conditions must be the same.
Integration Server Administrator	The HTML-based utility used to administer the webMethods Integration Server has a new look and feel. Menu and screen locations have not changed.
Internal Server ports that use the HTTPS protocol can now be configured using the “Use JSSE” option.	Internal Server ports that support TLS 1.1 or TLS 1.2 protocol can now be configured using the “Use JSSE” option in the Edit Internal Server Configuration screen of Integration Server Administrator. Previously, this option was not available.
OAuth Errors now returned in the content type specified in the client request	Previously, Integration Server always returned OAuth errors as HTML. Now, Integration Server returns OAuth errors in the content type specified in the client request.
Reply messages for webMethods messaging routed through Universal Messaging.	Integration Server now always encodes reply message as IData even when the encoding type of the reply document type is protocol buffers.

9.0 Added, Removed, Deprecated, or Changed Built-In Services

A release is listed in this section only if changes occurred in that release.

Release 10.7

Added Service	Description
pub.alert.notifier:create	Creates a notifier.
pub.alert.notifier:delete	Deletes the specified notifier.
pub.alert.notifier:disable	Disables the specified notifier.
pub.alert.notifier:enable	Enables the specified notifier.
pub.alert.notifier:list	Lists all notifiers in the system.
pub.alert.notifier:update	Updates the settings of the specified notifier.
pub.alert:channels	Gets all alert channels in the system.
pub.alert:severities	Lists all the severities in the system.
pub.alert:countAll	Gets the count of all alerts in the system.
pub.alert:countUnread	Gets the count of all unread alerts in the system.
pub.alert:emit	Generates an alert for the configured alert channel.
pub.alert:fetch	Gets unread alerts in the system based on the page number and page size.
pub.alert:fetchAll	Gets all alerts in the system based on the page number and page size.
pub.alert:markAllAsRead	Marks all alerts as read.
pub.alert:markAsRead	Marks a specific alert as read.
pub.alert:markAsUnread	Marks a specific alert as unread.
pub.alert:getSettings	Gets the settings of the system notifier.
pub.alert:setSettings	Updates the settings of the system notifier.
pub.client.oauth:getExternalAccessToken	Gets the access and refresh tokens that the OAuth server generates for the Integration Server. This service can be used when Integration Server uses the OAuth authentication mechanism for services such as “pub.client:smtp”.
pub.client.sftp.admin:getDefaultAlgorithms	Gets the default list of algorithms supported by the Integration Server SFTP client.
pub.client.sftp.admin:getHostKey	Gets the host key of the SFTP server.

Added Service	Description
pub.json:getArrayIterator	Returns a batch iterator object.
pub.json:getNextBatch	Gets the next batch of array elements by parsing the array paths in the iterator object returned by the “pub.json:getArrayIterator” service.
pub.json:closeArrayIterator	Closes the iteration. The iterator object used in an iteration cannot be reused after this service runs.
pub.mime:releaseBodyPartReferenceInTSpace -	Utility service that releases the reference to the file in the Tspace to free up disk space
pub.parquet:closeBatchIterator	Closes the Parquet iterator.
pub.parquet:documentTypeToSchema	Converts an Integration Server document type to the Parquet schema format.
pub.parquet:getBatchIterator	Reads a Parquet file and returns a batch iterator object that can be used to iterate over the records in the Parquet file.
pub.parquet:getNextBatch	Gets the next batch of Parquet records from the iterator object returned by the pub.parquet:getBatchIterator service.
pub.parquet:read	Reads a Parquet file and converts it to an IData array (Document list).
pub.parquet:write	Writes a document list (an array of IData objects) to a Parquet file
pub.security:clearAuthenticationCache	Clears the authentication cache in Integration Server.

Changed Service	Description
pub.client.sftp:login	The SFTP login service includes additional input parameters that enable you to log in without configuring the SFTP server and user aliases.

Changed Service	Description
pub.client:smtp	The service includes the “useJSSE” field in the “secure” input parameter to provide TLS 1.1 and 1.2 support for outbound SMTP connections. The service adds support for OAuth 2.0 as an alternative authorization mechanism. For this purpose, the service now includes the “bearer” type in the “auth” field and the “token” field to accept the access token issued by the OAuth Server to Integration Server.
pub.client:restClient	The service now includes “OAUTH” and “APIKEY” fields as “type” for the “auth” input parameter. For this purpose, the service now includes the support of “token” and “apiKey” fields as “type” while using “OAUTH” and “APIKEY”. The service also includes the support of the “secure” input parameter and the “trustStoreAlias” field so that users can specify the truststore information for certificate validation.
pub.flatFile:convertToString	The service now includes the support of the “lineSeparator” filed for the “delimiters” input parameter which specifies the character to be used to override the input record delimiter.
pub.json:documentToJSONString	The service now includes the encodeDateAs input parameter to specify how java.util.Date instances in the document are encoded in the returned JSON.
pub.json:jsonStreamToDocument	The service now includes the unescapeSpecialChars input parameter to control whether Integration Server unescapes the special characters '\n', '\r', '\t', '\b', '\f', '\\', '\"' while parsing JSON documents. The service now includes the decodeNullRootAsEmpty input parameter which controls whether the service converts a null value that Integration Server retrieves from JSON content to either IData or empty IData.

Changed Service	Description
pub.json:jsonStringToDocument	The service now includes the unescapeSpecialChars input parameter to control whether Integration Server unescapes the special characters '\n', '\r', '\t', '\b', '\f', '\\', '\"' while parsing JSON documents. The service now includes the decodeNullRootAsEmpty input parameter which controls whether the service converts a null value that Integration Server retrieves from JSON content to either IData or empty IData.
pub.mime:addBodyPart	If the data passed to this service is greater than the threshold set by the server configuration property watt.server.mime.largeDataThreshold, this service internally stores the content as a temporary file in the Tspace.
pub.mime:createMimeData	The input parameter of this service now also accepts a javax.mail.internet.MimeMessage object.
pub.mqtt:triggerSpec	The specification for MQTT trigger input now includes the *topicName* field. MQTT triggers can use wildcards to receive messages from multiple topics.
pub.string:tokenize	The service now includes the support of 'useRegex' parameter which indicates whether Integration Server must support recognizing delimiter character set for 'delim' parameter as regular expressions.

Release 10.5

Added Service	Description
pub.client:restClient	Creates and sends REST API requests over HTTP or HTTPS. Integration Server generates the REST connector services while creating a consumer REST API descriptor and calls this service while executing any REST connector service.
pub.compress:compressData	Compresses the data before sending the HTTP request using any of the specified compression schemes.

Added Service	Description
pub.compress:decompressData	Decompresses the data based on the response header of the HTTP response.
pub.datetime:build	Builds a date/time string using the specified pattern and the supplied date/time elements.
pub.datetime:increment	Increments or decrements a date and time by a specified amount of time.
pub.graphql:load	Loads data for a single key using Dataloader.
pub.graphql:loadMany	Loads data for multiple keys using Dataloader.
pub.mqtt:publish	Publishes an MQTT message to an MQTT server.
pub.mqtt:triggerSpec	Specification for the signature of an MQTT trigger service.

Deprecated Service	Replacement, if any
pub.date:dateBuild	pub.datetime:build
pub.date:dateTimeBuild	pub.datetime:build
pub.date:incrementDate	pub.datetime:increment

Changed Service	Description
pub.client:http	The default value of the Accept header is now controlled by the watt.net.default.accept server configuration parameter. Previously, the default value of the Accept header was: "image/gif, */*"
pub.xml:documentToXMLString	Adds the input parameter preserveRefs which indicates whether the leading & (ampersand) of a well-formed entity or character reference is left as & or further encoded as & when the data is to be HTML-encoded.
pub.xml:xmlStringToXMLNode	Adds input parameter validateXML which indicates whether the service validates the incoming XML document to ensure it is well-formed XML before converting it to a node.

Changed Service	Description
pub.client:restClient	The service now includes “OAUTH” and “APIKEY” fields as “type” for the “auth” input parameter. For this purpose, the service now includes the support of “token” and “apiKey” fields as “type” while using “OAUTH” and “APIKEY”. The service also includes the support of the “secure” input parameter and the “trustStoreAlias” field so that users can specify the truststore information for certificate validation.

Release 10.4

Added Service	Description
pub.json.schema:validate	Validates JSON content against a JSON document type.
pub.string:isNullEmptyOrWhitespace	Determines if a string is null, empty, or only whitespace.
pub.websocket:getCookies	Retrieves the cookies that are part of the HTTP request to establish a WebSocket connection.
pub.websocket:getExtensions	Retrieves the extensions used in the request URL to establish a WebSocket connection.
pub.websocket:getHeaders	Retrieves the headers used in the request URL to establish a WebSocket connection.
pub.websocket:getPathParameter	Retrieves the path parameter used in the request URL to establish a WebSocket connection.
pub.websocket:getRequest	Retrieves the HTTP request information used to establish the WebSocket connection.
pub.websocket:getQueryParameters	Retrieves the query parameters used in the request URL used to establish a WebSocket connection.
pub.websocket:getQueryParameter	Retrieves the value of specific query parameter information in the request URL used to establish a WebSocket connection.
pub.websocket:getSubProtocols	Retrieves the sub protocols used in the request URL to establish a WebSocket connection.

Added Service	Description
pub.websocket:ping	Sends a ping message containing the given application data to the remote endpoint after establishing the connection.
pub.websocket:pong	Sends back an unsolicited pong message with the exact application data as the ping for the WebSocket session.

Removed Service	Replacement, if any
pub.vcs* services	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer.

Deprecated Service	Replacement, if any
pub.json:validate	Use pub.json.schema:validate to validate JSON content against a JSON document type.
pub.soap.utils:resetWSDEffectivePolicy	None. The service is deprecated because the service applies to web services that run in pre-8.2 compatibility mode only. The ability to run in pre-8.2 compatibility mode is deprecated as of Integration Server 10.4 due to the deprecation of the web services implementation introduced in version 7.1
pub.soap.utils:setWSDEffectivePolicy	None. The service is deprecated because the service applies to web services that run in pre-8.2 compatibility mode only. The ability to run in pre-8.2 compatibility mode is deprecated as of Integration Server 10.4 due to the deprecation of the web services implementation introduced in version 7.1
pub.string:isNullOrBlank	pub.string:isNullEmptyOrWhitespace

Changed Service	Description
pub.client:websocket	Adds input parameters cookies, extensions, and subProtocols which are to be parts of the HTTP request used to establish a WebSocket connection.

Changed Service	Description
pub.jms:reply	When sending a JMS message to Universal Messaging, Integration Server sets the JMSMessageID.
pub.jms:send	When sending a JMS message to Universal Messaging, Integration Server sets the JMSMessageID.
pub.jms:sendAndWait	When sending a JMS message to Universal Messaging, Integration Server sets the JMSMessageID.
pub:oauth:getAccessToken	Supports requests for an access token for the Resource Owner Password Credentials (ROPC) grant type for a public client. Previously, only confidential clients could obtain an access token for the ROPC grant type.
pub:oauth:getToken	Supports requests for an access token for the Resource Owner Password Credentials (ROPC) grant type for a public client. Previously, only confidential clients could obtain an access token for the ROPC grant type.

Release 10.3

Added Service	Description
pub.flow:getLastFailureCaught	Returns information about the last failure that was caught by a CATCH step.

Changed Service	Description
pub.client:http	A new input parameter *trustStore* has been added which specifies the alias for the truststore that contains the list of certificates that Integration Server uses to validate the trust relationship.
pub.mime*	The multipart MIME messages created by the pub.mime:* services now strip line folding from the header fields.
pub.oauth:getAccesssToken	The service now supports the client credentials grant and the resource owner password credentials grant types.

Changed Service	Description
pub.oauth:getToken	The service now supports the client credentials grant and the resource owner password credentials grant types.

Release 10.2

Added Service	Description
pub.client:websocket	Establishes a WebSocket connection to the URL captured in the identified WebSocket client endpoint.
pub.oauth:getToken	Requests an access token from the Integration Server acting as the authorization server. This service replaces the pub.oauth:getAccessToken and pub.oauth:refeshAccessToken services. Providing a single OAuth token endpoint for requesting an access token conforms to the OAuth standard, RFC 6749.
pub.oauth:removeExpiredAccessTokens	Removes expired access tokens from the database.
pub.websocket:close	Closes the WebSocket connection.
pub.websocket:disconnect	Closes the WebSocket connection abruptly.
pub.websocket:onBinary	Specification to use as the signature for a callback service related to WebSocket server or client endpoint that handles a binary payload.
pub.websocket:onClose	Specification to use as the signature for a callback service that gets invoked when a WebSocket session is closed gracefully or disconnected.
pub.websocket:onConnect	Specification to use as the signature for a callback service that gets invoked when a WebSocket session is successfully established.
pub.websocket:onError	Specification to use as the signature for a callback service that gets invoked when an error is encountered, resulting in WebSocket session termination.
pub.websocket:onText	Specification to use as the signature for a callback service that handles a text payload.
pub.websocket:send	Sends a message on a connected WebSocket session.

Deprecated Service	Replacement, if any
pub.jms.wmjms:receiveStream	None. webMethods Broker is deprecated, resulting in the deprecation of this service.
pub.jms.wmjms:sendStream	None. webMethods Broker is deprecated, resulting in the deprecation of this service.
pub.oauth:getAccessToken	pub.oauth:getToken
pub.oauth:refreshAccessToken	pub.oauth:getToken

Changed Service	Description
pub.client:http handling of a 307 or 308 status code	When the pub.client:http service sends a POST, PUT, DELETE, or PATCH request and receives a 307 or 308 response status code from an HTTP server, the service no longer changes the request method to GET when redirecting.
pub.publish:envelope	The *uuid* field can now be set in the document envelope when publishing a document to Universal Messaging using the pub.publish:publish service. Previously, the *uuid* field was read-only.

Release 10.1

Added Service	Description
pub.json:validate	Validates JSON content against a JSON schema.
pub.oauth:introspectToken	Checks if an access token or refresh token generated by an Integration Server acting as an authorization server is active.
pub.oauth:revokeToken	Revokes a token issued by an Integration Server acting as an authorization server
pub.xmldata:getGroupObjects	Returns information about the objects and their tags in a group value. This service is useful primarily when the structure of data is not known in advanced and must be discovered. Examples include any wildcard content, anyType content, substitution groups, and XmlData that was created without using an XML document type.

Added Service	Description
pub.xmldata.getGroupValues	Returns information about the values and their tags in a group value. This service is useful primarily when the structure of data is not known in advanced and must be discovered. Examples include any wildcard content, anyType content, substitution groups, and XmlData that was created without using n XML document type.
pub.xmldata:queryXMLNode	Queries an XML node and returns the results as XMLData.

Removed Service	Replacement, if any
pub.restV2:listAllRESTResources	None. REST V2 resources are visible in the Package Navigator view of Designer.

Changed Service	Description
pub.jms:sendBatch	The pub.jms:sendBatch service can now be used with Universal Messaging as the JMS provider.
pub.report*	Integration Server now executes the template file and any services therein, using the credentials of the user who invoked the pub.report* service.
pub.xml:loadEnhancedXMLNode	The inputProcessing\supportDTD input parameter now has a default value of false. The inputProcessing\isSupportingExternalEntities input parameter now has a default value of false.
pub.xml:XMLStringToEnhancedXMLNode	The inputProcessing\supportDTD input parameter now has a default value of false. The inputProcessing\isSupportingExternalEntities input parameter now has a default value of false.

Release 10.0

Added Service	Description
pub.flow:HTTPRequest	Document type that represents information about the HTTP request received by Integration Server.

Added Service	Description
pub.flow:HTTPResponse	Document type that specifies the HTTP response information to be returned by Integration Server to the client.
pub.restV2:listAllRESTResources	Lists the REST resources configured using the URL template-based approach for a specified Integration Server service or for all the Integration Server services in a specified package. These REST resources are invoked using the restv2 directive.

Removed Service	Replacement, if any
pub.event.nerv:eventToDocument	No replacement service because the pub.event.routing:subscribe service, which replaces the deprecated pub.event.nerv:subscribe service, handles the conversion of the incoming JMS message into an IS document (IData).
pub.event.nerv:send	pub.event.routing:send
pub.event.nerv:subscribe	pub.event.routing:subscribe
pub.event.nerv:unsubscribe	pub.event.routing:unsubscribe

Changed Service	Description
pub.client:ftp	Updated the input parameter *secure* to include *useJSSE* as a child parameter.
pub.client.ftp:login	Updated the input parameter *secure* to include *useJSSE* as a child parameter.
pub.client:http	You can now supply a value in the data/bytes input variable when the method input variable is DELETE.

Changed Service	Description
pub.security.xml:signXML	Added input parameter isDetached to indicate whether the signature should be detached or not. Added input parameter keyName which is the name that is used by the signer to communicate a key identifier to the recipient. Added input parameter keyNameValue to indicate whether to include the key value (RSAKeyValue or DSAKeyValue), based on the certificate used for signing. The KeyValue element contains a single public key that can be used in validating the signature. The certData input parameter now includes X509_CRL as an option.

Release 9.12

Added Service	Description
pub.cache.serviceResults:addCacheEntry	Adds cached entry into service results for the service without executing the actual service. This service can be useful to perform bulk load of service results at Integration Server startup.
pub.cache.serviceResults:listServiceCache	Returns a list of the cached service results for a particular service.
pub.cache.serviceResults:resetServerCache	Resets the cache for all services in the service results cache, resulting in the removal of all cached service results for all services from the service results cache.
pub.cache.serviceResults:resetServiceCache	Resets the cache for specific service, resulting in the removal of cached service results for the service.

Changed Service	Description
pub.client:http	The default value of the *useJSSE* parameter is now null, which allows the watt.net.ssl.client.useJSSE server configuration property to determine the default behavior for outbound HTTPS connections. Added input parameter *delegation* to allow a user to execute a service on behalf of other user. Added input parameter *requestDelegatableToken* to specify if you want to request for a forwardable ticket granting ticket to send to the intermediary. The intermediary can use this forwardable ticket granting ticket for Kerberos delegation.Integration Server, which the server can use for delegation
pub.client.ldap:search	Added input parameter *pageSize* to set the number of entries to return in a page. Added input and output parameter *ldapCookie* which contains the index of the page count for the search.
pub.client.ftp:login	Added input parameter *useJSSE* to allow use of JSSE for creating outbound FTPS connections.
pub.client:soapClient	Added input parameter *useJSSE* to allow use of JSSE for outbound web service invocation.
pub.date*	Previously, if a value was not specified for the locale input parameter, Integration Server used the locale from the session used by the client that invoked the service. Now, Integration Server uses the value of the watt.server.session.locale.ignore server configuration parameter to determine whether the locale is obtained from the session used by the client that invoked the service or if the locale is the locale of Integration Server.
pub.flow:setResponse2	Added input variable responseStreamTransferEncoding to use chunked transfer encoding for the response and include the "Transfer-Encoding: chunked" response header.
pub.json:jsonStreamToDocument	Added input parameter decodeRealAsString to converts real numbers in the jsonStream to String.
pub.json:jsonStringToDocument	Added input parameter decodeRealAsString to converts real numbers in the jsonStream to String.

Changed Service	Description
pub.security:userInfoSpec	The error input parameter is now named userInfoError.

Release 9.10

Added Service	Description
pub.date:compareDates	Compares two dates.
pub.date:incrementDate	Increments a date by a specified period.
pub.event.routing:eventAcknowledgement	Defines the input signature for a callback service that processes acknowledgments sent by the Event Routing framework.
pub.security:userInfoSpec	Specification for UserInfo service that performs custom processing based on the personally identifiable information in the OpenID Connect UserInfo token.
pub.string:compareStrings	Performs a case-sensitive comparison of two strings and indicates whether the strings are identical.
pub.string:isAlphanumeric	Determines whether a string consists entirely of alphanumeric characters (in the ranges A–Z, a–z, or 0–9).
pub.string:isDate	Determines whether a string follows a specified date pattern.
pub.string:isNullOrBlank	Checks a string for a null or a blank value.
pub.string:isNumber	Determines whether the contents of a string can be converted to a float value.
pub.string:substitutePipelineVariables	Replaces a pipeline variable with its corresponding value.

Changed Service	Description
pub.client:http	The *method* input parameter now supports the patch method. The pub.client:http service now includes fields for supplying Kerberos information that Integration Server uses to acquire a Kerberos ticket to include in the outbound service request. Specifically, the *auth* input parameter now contains a *kerberos* document in which you can specify *jaasContext*, *clientPrincipal*, *clientPassword*, *servicePrincipal*, and *servicePrincipalForm*. If the remote server redirected pub.client:http to a different location, *encodedURL* contain the URL that pub.client:http submitted to the server to which it was redirected.
pub.event.routing:send	New input parameters *isAsync*, *service Name*, and *runAsUser.*
pub.jms:send	Added the *JMSMessage/header/replyTo* input parameter which can be used to specify a destination for replies without actually needing to wait for replies.
pub.security.keystore.pkcs7:sign	New input parameters *dataAsStream* and *signatureAsStream*.
pub.security.pkcs7:verify	New input parameters *signatureAsStream* and *dataAsStream*. New output parameter *contentAsStream*.
pub.security.util:createMessageDigest	New input parameter *inputAsStream*. New output parameter *outputAsStream*

10.0 Added, Removed, Deprecated, or Changed Parameters

A release is listed in this section only if changes occurred in that release.

Release 10.7

Added Parameter	Description
watt.core.nodeIterator.retainFailures	Specifies whether the NodeIterator retains a skipped XML node in moving-window mode.
watt.net.jsse.client.disabledProtocols	Specifies the list of disabled SSL and TLS protocol versions when Integration Server acts as a client making outbound requests.
watt.net.jsse.server.disabledProtocols	Specifies the list of disabled SSL and TLS protocol versions when Integration server acts as the server handling inbound requests.
watt.net.ssl.email.client.useJSSE	Controls the use of JSSE for all the outbound SMTP connections from Integration Server.
watt.security.min.bytesPerSecond	Specifies the minimum rate at which data may arrive at Integration Server. If data arrives more slowly than the watt.security.min.bytesPerSecond value, Integration Server rejects the request with a 408 Request Timeout error and then closes the connection. Note: The watt.sercurity.min.bytesPerSecond parameter was introduced in PIE-53144 which was included in IS_10.3_Core_Fix2, IS_10.1_Core_Fix9, IS_9.12_Core_Fix19, IS_9.10_Core_Fix16, and IS_9.10_Core_Fix16 and had a slightly different behavior. See the watt.security.min.bytesPerSecond description in the Changed Parameter table, below.
watt.server.audit.journal.truncated	The logging level at which Integration Server writes the full value of a truncated field from an audit record to the server log. Integration Server only write the full field to the server log when the server logging facility 0095 Audit Log Manager is set to the same level as watt.server.audit.journal.truncated or a more verbose level.
watt.server.clientTimeoutRedirect	Specifies how Integration Server Administrator behaves when the browser session times out.
watt.server.http.forwardableHeaders	The headers that Integration Server collects from inbound requests and includes in outbound requests.

Added Parameter	Description
watt.server.http.forwardHeaders	Specifies whether Integration Server supports distributed tracing by collecting the headers used by Istio Service Mesh and opentracing.
watt.server.jdbc.datadirect.useJaasSubjectForKerberos	Specifies that Integration Server uses the Subject in the Integration Server JAAS configuration when using Kerberos with JDBC pool connections.
watt.server.json.decodeNullRootAsEmpty	Converts a null data that Integration Server retrieves from JSON content to either IData or empty IData.
watt.server.json.encodeDateAs	Specifies how java.util.Date objects are formatted by Integration Server, including IDataJSONCoder.encode() and the pub.json:documentToJSONString service.
watt.server.json.iterator.maxBatchSize	Specifies the maximum number of array elements that the pub.json:getNextBatch service can retrieve.
watt.server.json.iterator.maxConcurrentRequests	Specifies the maximum number of concurrent requests for JSON iterator that Integration Server should process.
watt.server.jms.trigger.caching.pollingInterval	Specifies the length of time, measured in milliseconds, between polling requests to Universal Messaging to retrieve messages for a JMS trigger that uses prefetch caching (which is also called consumer caching).
watt.server.json.iterator.minBatchSize	Specifies the minimum number of array elements that the pub.json:getNextBatch service can retrieve.
watt.server.messaging.deliverNotificationErrors	Specifies whether Integration Server generates and delivers pub.publish.notifcation.error messages.
watt.server.messaging.trigger.startup.useServerThread	Specifies whether the thread used to start and run a webMethods messaging trigger that receives messages from Universal Messaging counts toward the thread usage limit established by the Maximum Threads value for document processing.

Added Parameter	Description
watt.server.mime.largeDataThreshold	The payload size at which a payload greater than the parameter value results in an execution of the pub.mime:getEnvelopeStream service returning a javax.mail.internet.MimeMessage object instead of an InputStream.
watt.server.oauth.log.authErrors	Specifies whether Integration Server writes OAuth authorization failures to the error log.
watt.server.ping	Specifies the number of milliseconds that the Pinger thread created for the pub.remote:invoke service waits between sending wm.sever.ping requests.
watt.server.ports.ipaccess. ignoreXForwardedForHeader	Specifies whether Integration Server uses or ignores the IP address in the X-Forwarded-For (XFF) header of a request when determining whether to allow or deny access to the port. This parameter applies to requests received on an Enterprise Gateway external port and API Gateway external port only.
watt.server.rest. addHTTPMethodToInputPipeline	Specifies whether Integration Server must add the '$httpMethod' input variable with the requested HTTP method in the input pipeline while processing a REST request.
watt.server.securePort	Specifies the port number of the default secure (HTTPS) port on Integration Server.
watt.server.securityLog. ignoreXForwardedForHeader	Specifies whether Integration Server ignores the X-Forwarded-For request header and uses the IP address of the proxy server as the client ID.
watt.server.securityLog.logAnonymousRequests	Controls whether anonymous requests are written to the security log.
watt.server.smtp.userName	Specifies the username to be used by Integration Server to log in to the SMTP server.
watt.server.stats.logfile.csv	Specifies the format of the statistics log.
watt.server.stats.packages.exclude	Specifies the list of packages whose services are excluded from the request statistics.

Added Parameter	Description
watt.server.transaction.ignore.exception	Specifies whether Integration Server ignores exceptions that occur when committing an implicit or explicit transaction.
watt.server.trigger.shutdown.timeout	Specifies the number of seconds that Integration Server wait for trigger services to finish executing after disabling a concurrent webMethods messaging trigger that retrieves messages from Broker.
watt.server.xmlCoder. getUndefinedDataTypeClassName	Determines whether the class name of an unsupported data type is included in the XML document produced by com.wm.util.coder.XMLCoder.encode, the pub.document:documentToXMLValues service, and the deprecated service pub.record:XMLValuesToRecord, the IData object returned by the pub.document:XMLValuesToDocument service and the deprecated pub.record:XMLValuesToRecord service, and the Values object returned by com.wm.util.coder.XMLCoder.decode.
watt.serverlog.includeThreadID	Specifies whether the log messages written to the server log include the thread ID of the thread performing the action.
watt.wmcloud.listeners.maxIdleTime	Determines the length of time an on-premise listener for a webMethods Cloud account can be idle before it gets recreated.
watt.wmcloud.listeners.monitoringInterval.	Determines the interval at which monitoring thread executes to ensure that each enabled webMethods Cloud account has an active listener on the on-premise Integration Server.

Removed Parameter	Replacement, if any
watt.net.jsse.client.enabledProtocols	watt.net.jsse.client.disabledProtocols
watt.net.jsse.server.enabledProtocols	watt.net.jsse.server.disabledProtocols

Deprecated Parameter	Replacement, if any
watt.ssh.jsch.kex	The Key Exchange Algorithms can be included and reordered in the “Preferred Key Exchange Algorithms” list and excluded by moving them to the “Excluded Key Exchange Algorithms” list.
watt.ssh.jsch.ciphers	All the ciphers can be included and reordered in either the “Preferred Ciphers S2C” or “Preferred Ciphers C2S” lists. Similarly, they can be excluded from either the “Excluded Ciphers S2C” or “Excluded Ciphers C2S” list.
watt.ssh.jsch.mac_s2c	The Server to Client MAC algorithms can be included and reordered in the “Preferred MAC Algorithms S2C” list and can be excluded by moving them to the “Excluded MAC Algorithms S2C” list.
watt.ssh.jsch.mac_c2s	The Client to Server MAC algorithms can be included and reordered in the “Preferred MAC Algorithms C2S” list and can be excluded by moving them to the “Excluded MAC Algorithms C2S” list.

Changed Parameter	Description
watt.adminapi.returnExceptions	The default value of this parameter is now false.

Changed Parameter	Description
watt.security.min.bytesPerSecond	In cases where the reading of the bytes is controlled by a customer's application or by a third-party library, Integration Server is not reading the incoming bytes and cannot monitor the rate at which they arrive. Integration Server does not enforce the watt.security.min.bytesPerSecond value in these scenarios: - When the request payload is JSON and watt.server.http.jsonFormat is stream - When the request payload is XML and watt.server.http.xmlFormat is stream - When the request contains no Content-type header - When enhanced XML parsing is being used - When using the pub.xml NodeIterator services - For SOAP requests
watt.server.commonmessaging.trigger.stopRequestTimeout	The default value is now 3 seconds instead of 1.
watt.server.package.maxSizeMB	The value of this parameter is now a long. Previously, it was an int.
watt.server.threadPool.cloudRequests	The default value is now 5 instead of 75.

Release 10.5

Added Parameter	Description
watt.adminapi.group.readOnly	Specifies the name of the user group whose members have read-only access to the Integration Server Administrator API.
watt.adminapi.log.clientErrors	Specifies whether client errors that occur during execution of REST resources in the Administrator API are written to the error log.
watt.adminapi.returnExceptions	Specifies whether the HTTP response sent when the Integration Server Administrator API encounters an exception includes a stack trace in the response body.

Added Parameter	Description
watt.core.schema.anonymousCyclic ExtensionDepth	Specifies the nesting level when creating an IS schema from an XML schema definition that contains an anonymous complex type definition that references its own parent complex type extension.
watt.net.default.accept	Specifies the default value of the Accept header when an Accept header is not present in the headers input parameter to the pub.client:http service.
watt.net.ssl.server.sessionlog	Specifies whether Integration Server logs the SSL session information.
watt.net.ssl.server.sessionlog.maxFileSize	Specifies the maximum size of the inboundSSLSessions.log file in megabytes (MB).
watt.net.ssl.server.sessionlog.cacheLogEntries	Specifies whether Integration Server tracks the SSL session log entries in cache.
watt.net.ssl.server.sessionlog.cachedLogEntries. expiryTime	Specifies, in seconds, how often Integration Server checks for and removes the expired SSL session log entries from its cache.
watt.net.ssl.server.sessionlog.file	Specifies either a fully qualified or relative path to the file to which Integration Server writes the SSL session information.
watt.net.ssl.server.sessionlog.prettyPrint	Specifies whether the SSL session log entry is formatted with carriage returns and indentation to make the SSL session log easier to read.
watt.net.ssl.server.sessionlog.includeTimestamp	Specifies whether the SSL session log entries are added along with the timestamp.
watt.server.commonmessaging.connection. retryPeriod	Specifies the length of time, in seconds, that Integration Server waits between connection attempts when a connection to the MQTT server fails.
watt.server.commonmessaging.trigger. monitoringInterval	Specifies the interval, measured in seconds, at which Integration Server executes resource monitoring services for MQTT triggers
watt.server.commonmessaging.trigger. restartTaskRetryCount	Specifies the maximum number of retry attempts the trigger restart task makes to start MQTT triggers automatically.

Added Parameter	Description
watt.server.commonmessaging.trigger.restartTaskRetryInterval	Specifies the number of seconds that the trigger restart task waits between attempts to restart MQTT triggers.
watt.server.commonmessaging.trigger.reuseSession	Indicates whether instances of an MQTT trigger use the same session on Integration Server.
watt.server.commonmessaging.trigger.stopRequestTimeout	Specifies the maximum amount of time, measured in seconds, that Integration Server waits after an MQTT trigger is disabled before forcing the MQTT trigger to stop processing messages.
watt.server.http.interceptor.outbound.enabled	Enables the use of an outbound HTTP interceptor.
watt.server.http.interceptor.outbound.impl	Fully qualified name of the class that implements the outbound interceptor interface.
watt.server.http.request.supportCompression	Specifies whether Integration Server needs to support HTTP request compression.
watt.server.http.Content-Security-Policy a	Sets the HTTP security header Content-Security-Policy in the responses to requests for Accessing the Integration Server Administrator. Use this property to detect and mitigate attacks such as Cross Site Scripting (XSS) and data injection.
watt.server.http.X-Permitted-Cross-Domain-Policies	Sets the HTTP security header X-Permitted-Cross-Domain-Policies, which informs clients what cross-domain policies they can use for accessing the Integration Server Administrator.
watt.server.http.response.supportCompression	Specifies whether Integration Server needs to support HTTP response compression.
watt.server.oauth.token.endpoint.internal.requireSecret	Specifies whether Integration Server ensures that authorization code provided by a confidential client invoking the OAuth token endpoint service was issued to the confidential client when the OAuth token endpoint service is invoked directly because it is on the same Integration Server acting as the OAuth authorization server.
watt.server.portAccess.axis2	Specifies whether Integration Server verifies that an Axis2-based web services can be accessed through a port.

Added Parameter	Description
watt.server.portStateless	Specifies a comma-separated list of the port numbers for the ports on Integration Server that are stateless.
watt.server.service.blacklist	Specifies, using a comma-separated list or a file, the services on the service blacklist and/or the interfaces whose services are on the service blacklist.
watt.server.soap.validateInput	Specifies whether a validation error occurs when an inbound SOAP request includes fields that are not declared in the service input signature.
watt.server.systemtasks.debug	Enables debug logging related to system tasks. The logs are written to server.log under the logging facility User Task Scheduler and include information such as when a task got created, when it got terminated, and for recurring tasks, when it will run again.
watt.server.threadPool.cloudRequests	Specifies the maximum percentage of the server thread pool that can be used for processing Integration Cloud requests concurrently.
watt.server.ws.security. usernameTokenTTL	Specifies, at the global level, the permitted time difference between the time when the UsernameToken was created and the time when it reaches the server.
watt.server.ws.security. usernameTokenFutureTTL	Specifies, at the global level, the permitted time difference for wsu:Created elements that have a timestamp in the future with respect to the Integration Server clock.
watt.ssh.jsch.kex	Specifies a comma-separated list of the key exchange algorithms that will appear in the Preferred Key Exchange Algorithms list when creating or editing an SFTP server alias.

Changed Parameter	Description
watt.ssh.jsch.ciphers	Previously, the parameter only supported adding more ciphers. Now, to remove a cipher, delete the cipher from the parameter value.

Changed Parameter	Description
watt.ssh.jsch.mac_c2s	Previously, the parameter only controlled the order of MAC algorithms for client to server transmission. Now, to disable use of a MAC algorithm, delete it from the parameter value.
watt.ssh.jsch.mac_s2c	Previously, the parameter only controlled the order of MAC algorithms for server to client transmission, respectively. Now, to disable use of a MAC algorithm, delete it from the parameter value.

Release 10.4

Added Parameter	Description
watt.core.schema.generate ObsoleteDocumentTypeNames	Specifies whether Integration Server uses the naming convention that existed prior to version 8.2 when generating document types for complex types referenced from a global element declaration.
watt.core.transientStore.logExceptions	Specifies whether Integration Server logs exceptions thrown by the transient store that might cause the consumer thread to terminate. This can aid in diagnosing problems with an internal store used by the resubmission process.
watt.frag.keep.original.servicename	Specifies whether Integration Server retains the original service name for input fields while fragging or compiling a Java service.
watt.net.clientKeepaliveAgingLimit	Specifies how long a socket in a client connection pool is kept alive, measured in seconds.
watt.net.clientKeepaliveUsageLimit	Specifies the maximum number of usages for a socket in a client connection pool.
watt.server.audit.file.fieldDelimiter	Specifies the character sequence to use to delimit fields within a log record in a file-based audit log
watt.server.audit.file.recordDelimiter	Specifies the character sequence to use to delimit records in a file-based audit log.
watt.server.audit.stdout.fieldDelimiter	Specifies the character string to delimit audit log fields within a log entry when the audit log is written to the console (STDOUT).

Added Parameter	Description
watt.server.audit.stdout.recordDelimiter	Specifies the character string to delimit the audit log records when the audit logger is written to the console (STDOUT).
watt.server.package.maxSizeMB	Specifies the maximum expanded size of a package that can be installed, measured in megabytes.
watt.server.serviceMonitor.queryOnServerId	Specifies monitoring capabilities for the entire stateful cluster and not just for a single instance of the cluster.
watt.server.SOAP.retainUndeclaredNamespace	Specifies whether Integration Server retains namespaces from an xsd:any element when decoding a SOAP request or SOAP response.
watt.server.search.fast	Specifies whether Integration Server must process the value assigned to the pipeline variable while loading a flow service thereby enhancing the search operation and displaying the search results much faster.

Deprecated Parameter	Replacement, if any
watt.server.package.pre82WSD. loadExternalResources	None. The web services implementation that handles web services that run in pre-8.2 compatibility mode is deprecated, specifically the implementation introduced in Integration Server version 7.1. As a result, this parameter is deprecated.
watt.server.setResponse.pre82Mode	None. The web services implementation that handles web services that run in pre-8.2 compatibility mode is deprecated, specifically the implementation introduced in Integration Server version 7.1. As a result, this parameter is deprecated.
watt.server.soap.convertPlainText HTTPResponseIntoSOAPFault	None. The web services implementation that handles web services that run in pre-8.2 compatibility mode is deprecated, specifically the implementation introduced in Integration Server version 7.1. As a result, this parameter is deprecated.

Deprecated Parameter	Replacement, if any
watt.server.SOAP.pre82WSD. ignoreVersionMismatch	None. The web services implementation that handles web services that run in pre-8.2 compatibility mode is deprecated, specifically the implementation introduced in Integration Server version 7.1. As a result, this parameter is deprecated.
watt.server.ws.responseTNS.from.request	None. The web services implementation that handles web services that run in pre-8.2 compatibility mode is deprecated, specifically the implementation introduced in Integration Server version 7.1. As a result, this parameter is deprecated.

Changed Parameter	Description
watt.server.cors.allowedOrigins	To make it easier to specify a long list of allowed URIs, Integration Server now allows specifying a file as the value for this parameter.
watt.server.cors.exposedHeaders	This parameter is no longer case-sensitive.
watt.server.netEncoding	When the pub.client:http service submits a password digest for authentication (that is, the auth/type field is set to Digest) and the HTTP server response includes the header field "Content-Type" but does not contain the charset parameter, uses the value of the watt.server.netEncoding server configuration parameter as the default character set.
watt.server.SOAP.ignoreMissingResponseHeader	This parameter now ignores missing required headers for returned SOAP faults.

Release 10.3

Added Parameter	Description
watt.net.jsse.server.SSLSessionTimeout	Specifies the amount of time in seconds for which Integration Server waits before timing out and removing an SSL session from the SSL cache.

Added Parameter	Description
watt.net.jsse.server.useCipherSuitesOrder	Specifies whether the local cipher suites preference should be honored by Integration Server during the SSL/TLS handshake when Integration Server acts as the SSL/TLS server and requires the use of JSSE
watt.security.kerberos.client.useSPNEGO	Specifies whether Integration Server generates a SPNEGO-based Kerberos ticket for all outbound requests that use Kerberos authentication.
watt.security.max.contentLength	Specifies the maximum size limit of the payload for an HTTP request.
watt.security.max.headerLength	Specifies the maximum length of the header for an HTTP request.
watt.security.max.urlLength	Specifies the maximum length of the URL for an HTTP request.
watt.server.autodeploy.enabled	Specifies whether automatic deployment of packages is enabled or disabled for Microservices Runtime. Note: The automatic package deployment feature is available by default for Microservices Runtime. To use the automatic package deployment feature with Integration Server, your Integration Server must have additional licensing.
watt.server.autodeploy.interval	Specifies the interval, measured in minutes, at which Microservices Runtime executes the autodeploy system task. Note: The automatic package deployment feature is available by default for Microservices Runtime. To use the automatic package deployment feature with Integration Server, your Integration Server must have additional licensing.
watt.server.autodeploy.alwaysUseHotDeployment	Specifies whether Microservices Runtime always uses hot deployment for automatic deployment of packages. Note: The automatic package deployment feature is available by default for Microservices Runtime. To use the automatic package deployment feature with Integration Server, your Integration Server must have additional licensing.

Added Parameter	Description
watt.server.checkPath.restorePipelineFromFile	Specifies whether Integration Server verifies that the value of the filename input parameter supplied to pub.flow:restorePipelineFromFile service is in the allowedReadPaths parameter of the file access control configuration file (fileAccessControl.cnf).
watt.server.checkPath.savePipelineToFile	Specifies whether Integration Server verifies that the value of filename input parameter supplied for the pub.flow:savePipelineToFile service is in the allowedWritePaths parameter of the file access control configuration file (fileAccessControl.cnf).
watt.server.coder.responseAsXML	Specifies how Integration Server receives the XML response for any REST API request -- in proper XML format or in converted XML format containing IData objects.
watt.server.jms.trigger.startupFailure.restartTaskRetryCount	Specifies the maximum number of retry attempts the trigger restart task makes to start the JMS triggers that fail to start when the JMS connection alias starts up
watt.server.jms.trigger.startupFailure.restartTaskRetryInterval	Specifies the number of seconds that the trigger restart task waits between attempts to restart JMS triggers that failed to start when the JMS connection alias started.
watt.server.oauth.token.endpoint.auth	Specifies whether the token endpoint accepts an existing session or requires credentials for authentication.
watt.server.publish.maxCSQRedeliveryCount	Specifies the maximum redelivery attempts Integration Server makes when publishing a message from the client side queue (CSQ) to the Broker.
watt.server.SOAP.inbound.CDATA.removeTags	Specifies whether Integration Server removes or preserves the CDATA delimiter tags found in an inbound SOAP request.
watt.server.search.fast	Specifies whether Integration Server must process the value assigned to the pipeline variable while loading a flow service thereby enhancing the search operation and displaying the search results much faster.

Added Parameter	Description
watt.server.enterprisegateway. ignoreXForwardedForHeader	Specifies whether Integration Server must ignore the 'X-Forwarded-For' request header while processing the rules in Enterprise Gateway. If this property is set to 'true' then Integration Server ignores the 'X-Forwarded-For' request header and considers the proxy server's IP address as the host IP address. If the property is set to 'false' then Integration Server obtains the actual host IP address from the 'X-Forwarded-For' request header. Default value is 'true'.

Removed Parameter	Replacement, if any
watt.ssl.iaik.clientAllowUnboundRenegotiate	None. This parameter is obsolete. Integration Server uses the Entrust security provider library which has been updated for RFC 5746 to support secure TLS/SSL renegotiation. Secure renegotiation cannot be disabled.
watt.ssl.iaik.serverAllowUnboundRenegotiate	None. This parameter is obsolete. Integration Server uses the Entrust security provider library which has been updated for RFC 5746 to support secure TLS/SSL renegotiation. Secure renegotiation cannot be disabled.

Deprecated Parameter	Replacement, if any
watt.server.publish.maxCSQRedeliveryCount	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.

Changed Parameter	Description
watt.server.coder.responseAsXML	The default value of this parameter is changed to ‘true’.

Release 10.2

Added Parameter	Description
watt.adapters.withOnlineHelp	Specifies a comma-separated list of the adapter names that have online help content that needs to be accessible through the Help link on Integration Server Administrator.

Added Parameter	Description
watt.core.xsd.useKnownSchemaLocation	Specifies whether Integration Server includes a "known schema location" and the schemaLocation attribute for an import statement in the WSDL document generated for a provider web service descriptor. When the schemaLocation attribute is not present in the WSDL document, an Integration Server that consumes the WSDL document will not dereference the schemaLocation attribute value during web service processing. If the known schema locations are the only import statements that required namespace resolution, the absence of the schemaLocation attribute allows web service development to occur while not connected to the Internet.
watt.server.checkWhitelist	Specifies whether Integration Server uses a whitelist to filter the list of classes than can be used for deserialization.
watt.server.circuitBreaker.threadPoolMax	Specifies the maximum number of threads that the server maintains in the circuit breaker thread pool. Note: The circuit breaker feature is available by default for a service that resides in a Microservices Runtime. To use the circuit breaker feature with Integration Server, your Integration Server must have additional licensing.
watt.server.circuitBreaker.threadPoolMin	Specifies the minimum number of threads that the server maintains in the circuit breaker thread pool. Note: The circuit breaker feature is available by default for a service that resides in a Microservices Runtime. To use the circuit breaker feature with Integration Server, your Integration Server must have additional licensing.
watt.server.dumpWhitelist	When whitelist class filtering is enabled, indicates whether or not Integration Server generates a log file that lists any classes for which deserialization was attempted but are not part of a whiteslistclasses.xml file.
watt.server.http.Strict-Transport-Security	Specifies whether Integration Server includes Strict-Transport-Security header in response headers

Added Parameter	Description
watt.server.http.X-Content-Type-Options	Specifies whether Integration Server includes the X-Content-Type-Options header in response headers.
watt.server.http.X-XSS-Protection	Specifies whether Integration Server includes X-XSS-Protection in the response headers.
watt.server.jms.csq.publishDelayWhileDraining	Specifies the number of seconds that Integration Server waits before publishing a JMS message to the client side queue (CSQ) while the CSQ drains.
watt.server.messaging.csq. publishDelayWhileDraining	Specifies the number of seconds that Integration Server waits before publishing a message to the client side queue (CSQ) while the CSQ drains.
watt.server.oauth.disableClient.disableTokens	Specifies whether a client access token issued to a client account that is disabled can be used to access resources
watt.server.password.historyLength	Specifies the maximum number of previously set passwords that Integration Server saves in the memory for a user.
watt.server.password.maxIdenticalCharsInARow	Specifies the maximum number of identical characters in a row a password can contain for non-Administrator users.
watt.server.password.maxLength	Specifies the maximum number of characters (alphabetic characters, digits, and special characters combined) the password must contain for non-Administrator users.
watt.server.serverlogMaskBearerToken	Specifies whether Integration Server masks the value of a bearer token when writing to the server log.
watt.server.SOAP.preserveCDATA	Specifies whether Integration Server encodes CDATA blocks in outbound messages.
watt.ssh.jsch.mac_c2s	Specifies the order of message authentication code (MAC) algorithms for client to server transmission.
watt.ssh.jsch.mac_s2c	Specifies the order of message authentication code (MAC) algorithms for server to client transmission.

Changed Parameter	Description
watt.server.illegalUserChars	This parameter applies to user names only now. Previously, it applied to usernames and passwords.
watt.server.password.minDigits	The default value of this parameter is now 0. Integration Server uses the new default value for new Integration Server instances only.
watt.server.password.minLowerChars	The default value of this parameter is now 0. Integration Server uses the new default value for new Integration Server instances only.
watt.server.password.minSpecialChars	The default value of this parameter is now 0. Integration Server uses the new default value for new Integration Server instances only.
watt.server.password.minUpperChars	The default value of this parameter is now 0. Integration Server uses the new default value for new Integration Server instances only.

Removed Parameter	Replacement, if any
watt.server.jms.debugTrace	Use enhanced logging for the JMS connection aliases used by the JMS triggers for which you want Integration Server to generate additional logging.
watt.server.SOAP.HTTP.useMailWriter	None. This parameter was for use with webMethods Mediator which reached end-of-life in 10.1.
watt.xslt.debug.facList	This parameter is obsolete. Use Integration Server Administrator to set the Server Logger configuration for the facilities in the WmXSLT package.
watt.xslt.debug.level	None. Use Integration Server Administrator to set the server logging level for the facilities in the WmXSLT package.
watt.xslt.debug.logfile	None. Integration Server writes log messages for the WmXSLT package to the server.log.

Deprecated Parameter	Replacement, if any
watt.broker.sync.enableBrokerSync	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.broker.sync.forceDispatcherInit	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.brokerCoder.verbose	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.core.brokerTypeCoder.verbose	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.core.brokerCoder.wireFormat	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.auditDocIdField	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.broker.producer.multiclient	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.broker.replyConsumer.fetchSize	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.broker.replyConsumer.multiclient	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.broker.replyConsumer.sweeperInterval	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.brokerTransport.dur	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.brokerTransport.max	None. WebMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.brokerTransport.ret	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.control.maxPersist	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.control.triggerInputControl. delayIncrementInterval	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.control.triggerInputControl.delays	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.dispatcher.comms.brokerPing	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.

Deprecated Parameter	Replacement, if any
watt.server.dispatcher.comms.connectionShareLimit	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.jms.trigger.maxPrefetchSize	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.jms.trigger.wmjms.clientIDSharing	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.jms.wmjms.lms.readTimeout	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.publish.useCSQ	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.publish.drainCSQInOrder	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.publish.usePipelineBrokerEvent	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.
watt.server.trigger.keepAsBrokerEvent	None. webMethods Broker is deprecated, resulting in the deprecation of this parameter.

Changed Parameter	Description
watt.server.content.type.mappings	Specifies a one-to-one mapping of content types for inbound and outbound requests. Previously, this parameter only mapped wildcards found in the Accept header field to a specific content type for the outbound content handler.
watt.server.http.interceptor.preprocess.sizeLimit	Integration Server now includes support for size units with this server configuration parameter. Set this property to N [KB|MB|GB], where N is any valid integer. Do not include any spaces between the integer and the unit of measure. If you do not specify a unit of measure, Integration Server treats the supplied N value as bytes.
watt.server.http.preserveUriReservedChars	The server configuration property watt.server.http.preserveUriReservedChars has been modified to include "space" as one of the reserved characters as well.

Release 10.1

Added Parameter	Description
watt.art.notification.publishToJMS.useCSQ	Specifies whether the asynchronous notification of Adapter Runtime can use CSQ (Client side Queueing) when publishing to JMS provider.
watt.server.http.interceptor.enabled	Specifies whether an HTTP interceptor is enabled for Integration Server.
watt.server.http.interceptor.impl	Specifies the fully qualified name of the java class for the HTTP interceptor implementation.
watt.server.http.interceptor.preprocess.sizeLimit	Specifies the maximum number of bytes that an HTTP interceptor reads during preprocessing.
watt.server.http.uriPath.decodePlus	Specifies whether or not the '+' character is interpreted as a ' ' when it appears in the path component of the request URI .
watt.server.jms.trigger.startupFailure.retryCount	Determines the maximum number of retry attempts that Integration Server makes to start a JMS trigger after the trigger fails to start.
watt.server.oauth.requirePost	Indicates whether Integration Server requires that client invoke the pub.oauth services via an HTTP POST request.

Added Parameter	Description
watt.server.rg.internalsocket.timeout	Specifies the length of time, in milliseconds, that Enterprise Gateway Server allows a client request to wait for a connection to the Internal Server before terminating the request with an HTTP 500 Internal Server Error.
watt.server.service.list.treatEmptyAsNull	Specifies whether Integration Server assigns a null value to all list data types like Document List, String List, Document Reference List, and Object list during the execution of a flow service without an input value.
watt.server.stats.logFilesToKeep	Specifies the number of stats.log files that Integration Server keeps on the file system, including the current log file. When Integration Server reaches the limit for the number of stats.log files, each time Integration Server rotates the stats.log, Integration Server deletes the oldest archived stats.log file.
watt.server.stats.logRotateSize	Specifies the file size at which Integration Server rolls over the stats.log.
watt.server.uri.decodePath	Specifies whether or not the '+' character is interpreted as a ' ' when it appears in the path component of the request URI.
watt.server.serviceErrorsAsLogged	Specifies whether the Integration Server administrator should display the service errors under "Statistics" page according to the actual number of errors logged in the WMERROR table.
watt.server.search.fast	Specifies whether Integration Server must process the value assigned to the pipeline variable while loading a flow service thereby enhancing the search operation and displaying the search results much faster.

Added Parameter	Description
watt.server.enterprisegateway. ignoreXForwardedForHeader	Specifies whether Integration Server must ignore the 'X-Forwarded-For' request header while processing the rules in Enterprise Gateway. If this property is set to 'true' then Integration Server ignores the 'X-Forwarded-For' request header and considers the proxy server's IP address as the host IP address. If the property is set to 'false' then Integration Server obtains the actual host IP address from the 'X-Forwarded-For' request header. Default value is 'true'.

Removed Parameter	Replacement, if any
watt.net.defaultBufferSize	None. This parameter specifies the maximum content length of an HTTP request that the WmTomcat package will process. The WmTomcat package has been removed from the product.
watt.net.webapp.cookies.useRelevantPath	None. This parameter specifies how WmTomcat can create fewer cookies to prevent the web application from logging out because of exceeding the browser cookie limit. The WmTomcat package has been removed from the product.

Changed Parameter	Description
watt.net.maxClientKeepaliveConns	The default value of this parameter is now 0. Integration Server uses the new default value for new Integration Server instances only.
watt.server.http.preserveUriReservedChars	Changes to this property no longer require a restart of Integration Server to take effect. The changes take effect immediately.
watt.server.oauth.requireHTTPS	If you set the value of this property to true and the client application accesses any of the pub.oauth services over HTTP, Integration Server issues an HTTP 400 error response to the client and writes a service exception to the error log. An HTTP 400 error indicates a client error. Previously, Integration Server issued an HTTP 500 error, which indicates a server error. According to the OAuth Framework , it is a client error when the client does not use a secure channel.

Release 10.0

Added Parameter	Description
watt.core.schema.useUnboundedForMaxOccurs	Specifies the number at which a maxOccurs value greater than this number is treated as unbounded.
watt.net.ssl.client.ftps.useJSSE	Controls the use of JSSE for all of the outbound FTPS connections from Integration Server. Set this parameter to true to use JSSE for all of the outbound FTPS connections. Set this property to false to indicate that JSSE is not used for outbound FTPS connections. The default is false. When executing the pub.client:ftp service or the pub.client.ftp:login service, the value of the useJSSE input parameter overrides the value of the watt.net.ssl.client.ftps.useJSSE server configuration parameter.

Added Parameter	Description
watt.server.audit.logFilesToKeep	Specifies the number of audit log files, including the current log file for the audit logger, that Integration Server keeps on the file system for an audit logger that writes to a file. When Integration Server reaches the limit for the number of log files for the audit logger, each time Integration Server rotates the audit log, Integration Server deletes the oldest archived audit log file.
watt.server.audit.logRotateSize	Specifies the file size at which Integration Server rolls over the audit log for a logger that writes to a file.
watt.server.cache.maxEntriesInCache	Specifies the default value for the Maximum Entries in Cache property for a distributed system cache used in an Integration Server cluster.
watt.server.diagnostic.logFiles.maxMB	Specifies the maximum number of megabytes of data that Integration Server reads from the file system for an audit log while collecting diagnostic data. This parameter affects the collection of audit log files only. It does not affect audit log data read from the database, nor does it affect other log files such as the server log, stats log or terracotta-client logs. The default is 250 megabytes. You do not need to restart Integration Server for changes to this parameter to take effect.
watt.server.jms.trigger.groupTag	Specifies the group tag used in the names of JMS triggers that belong to a trigger group. Integration Server treats JMS triggers with the specified group tag in the name as members of a trigger group.

Added Parameter	Description
watt.server.remoteInvoke.queryCSRFToken	Indicates if, during remote service invocation, Integration Server queries the remote server for the CSRF token for the current session and then includes the token in the service request. If an Integration Server uses Cross-Site Request Forgery (CSRF) guard, requests sent to the server must include a CSRF token. If the request does not include a CSRF token, the server rejects the request. When an Integration Server performs a remote invoke to execute a service on another Integration Server that uses CSRF guard, the request needs to include the CSRF token. To ensure that the request includes a CSRF token, the requesting Integration Server obtains the CSRF token for the current session from the remote Integration Server. The requesting Integration Server then modifies the request to include the CSRF token. However, this is only necessary if the remote Integration Server uses CSRF guard.
watt.server.response.displayISErrorCode	Indicates whether the response from Integration Server to a client application in the case of an error situation includes the Integration Server error code in the header and the body of the response. If you set the value of this parameter to false, the response during error situations includes the error message text without the Integration Server error code in both the header and the body of the response. If you set the value to true, the response during error situations includes the Integration Server error code and the corresponding error message text in the header and body of the response. The default value of the parameter is true.
watt.server.SOAP.default.endpointHTTP	Specifies the default provider web service endpoint alias for the HTTP protocol.
watt.server.SOAP.default.endpointHTTPS	Specifies the default provider web service endpoint alias for the HTTPS protocol.
watt.server.SOAP.HTTP.useMailWriter	Configures Axis stack to use an alternate multipart writer implementation based on the JavaMail AP

Added Parameter	Description
watt.server.SOAP.identifyISGeneratedWSDL	When creating a web service descriptor from a WSDL document as well as the refresh, Integration Server detects whether the WSDL document was originated by Integration Server.

Removed Parameter	Replacement, if any
watt.server.event.nerv.subscribeService.user	None. The pub.event.nerv:subscribe service has been removed from the product as of Integration Server version 10.0.

Changed Parameter	Description
watt.net.ssl.client.useJSSE	Controls the use of JSSE only for the outbound HTTPS connections from Integration Server. Note: This change does not impact any FTP connections created using the pub.client:ftp and the pub.client.ftp:login services.

Release 9.12

Added Parameter	Description
watt.core.validation.skipAbsentStarBody	Specifies whether to skip a validation that Integration Server performs when decoding mixed content elements that have an enumeration restriction.
watt.server.apiportal.url	Specifies the URL for establishing a connection to API Portal and publishing a REST API descriptor. This parameter derives the API Portal URL from the following: • Host name and port number for the API Portal connection. • Tenant for which the REST API descriptor is to be published. • Specifications of the REST API descriptor.

Added Parameter	Description
watt.server.jca.connectionPool. threadInterrupt.waitTime	Specifies the maximum number of milliseconds that a thread can take while creating or closing a connection before the pool interrupter thread interrupts the thread. After the specified time elapses, the pool interrupter thread considers the thread to be blocked and interrupts it.
watt.server.jca.connectionPool. threadInterrupter.sleepTime	Specifies the number of milliseconds the pool interrupter thread sleeps between sweeps for server threads that became blocked while creating or closing a connection.
watt.server.json.decodeRealAsString	Converts a real number that Integration Server retrieves from JSON content to String.
watt.server.log.maxEntries	Specifies the default number of log entries to be displayed in the log viewing utility.
watt.server.mediator.directives	Specifies a comma-separated list of Mediator directives.
watt.server.messaging.debugTrace	Enables an extra level of verbose logging for webMethods messaging triggers that receive messages from Universal Messaging or through Digital Event Services. You can configure the additional logging globally or at the individual webMethods messaging trigger level.
watt.server.ns.decodeJavaService	Enables Integration Server to display non-ASCII Unicode characters in the body of a Java service.

Added Parameter	Description
watt.server.ns.logDuplicateDocTypeRegistrationAsError	Indicates whether to suppress or continue logging of error messages related to registration of duplicate universal names for a document type. Set the value of the parameter to false to suppress the logging of the error messages and true to resume the logging. The default value of this parameter is true. Note: Setting the parameter value to false suppresses the error messages about duplicate universal names only. It does not resolve the duplicate names.
watt.server.returnCurrentDateTimeString	Returns the current date and time for an Object field that is not of type java.util.Date and has the format "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'".
watt.server.serverlogFilesToKeep	Specifies the number of server log files that Integration Server keeps on the file system, including the current server log file.
watt.server.session.locale.ignore	Specifies whether the default locale for the pub.date* services is the server locale or the locale from the session used by the client that invoked the service.
watt.server.SOAP.pre82WSD.ignoreVersionMismatch	For a web service provider that was created in an Integration Server release earlier than 8.2.2 and for which the Pre-8.2 compatibility mode property is set to true, specifies whether to emulate pre-8.2 behavior for process SOAP requests.

Added Parameter	Description
watt.server.sftp.dateStampFmt	Specifies the format of date to be used in the SFTP client public services, specifically the pub.client.sftp* services in the WmPublic package. To specify the date format to use, you can use any format that is supported by the Java class java.text.SimpleDateFormat . For example, to display the date with the format 08-12-02 14:44:33:1235, specify dd-MM-yy HH:mm:ss:SSSS. If the watt.server.sftp.dateStampFmt property is not set, Integration Server uses the default format, which is yyyy-MM-dd HH:mm:ss z.
watt.server.transaction.xastore.maxTxnPerFile	Specifies the maximum number of unique XA transactions in an XA recovery log file. When the XA recovery log file reaches the maximum number of transactions, Integration Server creates a new file.
watt.server.transaction.xastore.performXALogging	Specifies whether or not Integration Server writes transaction information to the XA recovery store. Set to true to instruct Integration Server to log information about the state and progress of each XA transaction. Set to false to instruct Integration Server to skip logging XA transaction information. The default is true.
watt.server.um.producer.transaction.commitRetryCount	Specifies the number of attempts made by Integration Server in publishing a guaranteed document to a Universal Messaging server, after the initial attempt at publishing fails because of a transaction failure.
watt.um.clientLog.level	Determines the information that is written to the Universal Messaging client log file, called umClient.log. Each level outputs log entries with that level or higher. Valid values are trace, debug, info, warn, error, fatal, and off. The default is error.

Added Parameter	Description
watt.um.clientLog.size	Maximum size, in MB, of the Universal Messaging client log file. When this size is reached, Integration Server rolls the file over to a backup called umClient (number).log and creates a new file.
watt.um.clientLog.fileDepth	Number of backup log files to keep on disk when using log rolling for the Universal Messaging client log file.

Changed Parameter	Description
watt.net.ftpDataConn	Now, when this parameter is set to true, Integration Server allows parallel downloads from multiple FTP sessions. When this parameter is set to false, Integration Server does not allow parallel downloads and reuses the same FTP session. The default is false.
watt.net.jsse.client.enabledCipherSuiteList	To make it easier to specify a long list of allowed cipher suites, Integration Server now allows specifying a file as the value for this parameter.
watt.net.jsse.server.enabledCipherSuiteList	To make it easier to specify a long list of allowed cipher suites, Integration Server now allows specifying a file as the value for this parameter.
watt.net.ssl.client.cipherSuiteList	To make it easier to specify a long list of allowed cipher suites, Integration Server now allows specifying a file as the value for this parameter.
watt.net.ssl.client.useJSSE	The default value of this parameter is now true, indicating that global default for outbound HTTP requests is to use JSSE. Individual outbound requests can override this setting.
watt.net.ssl.server.cipherSuiteList	To make it easier to specify a long list of allowed cipher suites, Integration Server now allows specifying a file as the value for this parameter.
watt.server.cors.allowedOrigins	Now supports the use of regular expressions in the comma-separated list of allowed origin servers. Integration Server treats any value in the comma-separated list that begins with "r:" as a regular expression.

Removed Parameter	Replacement, if any
watt.ssh.jsch.kex	

Release 9.10

Added Parameter	Description
watt.net.http401.throwException	Specifies whether the pub.client:http service throws a NetException when receiving a 401 error response or, instead, places the HTTP response header and body in the pipeline.
watt.net.http501-599.throwException	Specifies whether the pub.client:http service throws a ServiceException or returns response headers and response body when receiving a 501 to 599 level response from a remote HTTP server.
watt.security.openid.logExceptions	Specifies whether Integration Server writes OpenID errors to the error log.
watt.security.session.forceReauthOnExpiration	Specifies whether Integration Server accepts or rejects a request that includes an expired or invalid session.
watt.server.audit.um.sessionPool.max	Specifies the maximum number of sessions in the Universal Messaging session pool for each Universal Messaging connection alias used by an audit logger.
watt.server.audit.um.sessionPool.min	Specifies the minimum number of sessions in the Universal Messaging session pool for each Universal Messaging connection alias used by an audit logger.
watt.server.audit.um.sessionPool.retryInterval	Specifies the number of seconds Integration Server waits between attempts to re-establish a session on the Universal Messaging server after an audit logger enters queue fail-fast mode.
watt.server.jca.connectionPool. createConnection.interrupt.waitTime	Specifies the wait time, measured in milliseconds, that elapses before Integration Server interrupts a connection creation thread that is in a wait state

Added Parameter	Description
watt.server.package.pre82WSD.loadExternalResources	Specifies whether, at package load time, Integration Server loads external resources for a consumer web service descriptor or a WSDL first web descriptor created on a version of Integration Server prior to version 8.2 and for which the Pre-8.2 compatibility mode property is set to true.

Changed Parameter	Description
watt.server.oauth.custom.responseHeader	Specifies whether the OpenID redirection endpoint for Integration Server include a brief description of the error in the response header when an OpenID error or Integration Server exception occurs during authentication. Previously, this parameter applied to OAuth authorization server only.
watt.server.stats.pollTime	Restrictions for the value for the parameter. For a stand-alone Integration Server, the watt.server.stats.pollTime must be an integer greater than or equal to 0 (zero). For an Integration Server in a cluster, the watt.server.stats.pollTime must be an integer greater than 0 (zero) but less than or equal to 60.

11.0 Added, Removed, Deprecated, or Changed Java APIs

A release is listed in this section only if changes occurred in that release.

Release 10.7

Added API	Description
com.wm.app.b2b.server.cache.config.SearchResult	Cache search results.
com.wm.app.b2b.server.ServerAPI.registerCoderForMultipartServices	Registers the services and associated coder with Multipart content handler
com.wm.app.b2b.server.ServerAPI.removeCoderForMultipartServices	Unregisters the coder for the specified services from Multipart content handler.

Added API	Description
com.wm.util.coder.IDataJSONCoder. decodeNullRootAsEmpty	Indicates to the IDataJSONCoder whether null scalar value at root in the JSON being parsed should be converted to empty IData or to {"$rootValue":null}.
com.wm.util.coder.IDataJSONCoder. setDateEncoding	Indicates whether java.util.Date instances in the com.wm.data.IData being encoded should become long, timestamps, or Strings in the JSON text returned by encode(OutputStream, IData)
com.wm.util.coder.IDataJSONCoder. unescapeSpecialChars	Indicates to the IDataJSONCoder whether the escape characters will be un-escaped while decoding the JSON string value.

Deprecated API	Replacement, if any
com.wm.app.b2b.client.cache.config. SearchResult	com.wm.app.b2b.server.cache.config.SearchResult

Release 10.5

Added API	Description
com.softwareag.is.interceptor. HTTPInterceptorOutboundIFC	Defines a set of contracts that any outbound HTTP interceptor must implement.

Deprecated API	Replacement, if any
com.wm.data.IDataFactory.create(int size)	com.wm.data.IDataFactory.create()

Release 10.4

Removed API	Replacement, if any
com.webmethods.vcs.AbstractClient	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer
com.webmethods.vcs.VCSClient	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer

Removed API	Replacement, if any
com.webmethods.vcs.VCSException	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer
com.webmethods.vcs.VCSLog	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer

Release 10.3

Added API	Description
com.wm.app.b2b.server.ServerAPI. logSecurity	Logs a message to the security log. When security logging is enabled and Custom is selected as a security area to audit, use of this method allows users' applications to write entries to the security log.

Deprecated API	Replacement, if any
com.webmethods.vcs.AbstractClient	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer
com.webmethods.vcs.VCSClient	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer
com.webmethods.vcs.VCSException	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer

Deprecated API	Replacement, if any
com.webmethods.vcs.VCSLog	Use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer

Release 10.2

Added API	Description
com.wm.app.b2b.server.ServerAPI. registerCoderForMultipart	Registers a service and associated coder for a multipart content handler (ContentHandler_Multipart). The specified coder is the content handler for REST requests for that service.
com.wm.app.b2b.server.ServerAPI. removeCoderForMultipart	Unregisters the coder for the specified service from multipart content handler.

Deprecated API	Replacement, if any
com.wm.app.b2b.server.jms.producer. ProducerFacace.sendLargeMessageStream	None. webMethods Broker is deprecated, resulting in the deprecation of this method.
com.wm.app.b2b.server.jms.consumer. JMSTriggerFacade.getPrefetchSize	None. webMethods Broker is deprecated, resulting in the deprecation of this method.

Release 10.1

Added API	Description
com.softwareag.is.interceptor.HttpInterceptor Exception	Exception that can be thrown from an HttpInterceptorIFC implementation of the preProcess() method to interrupt normal processing of the HTTP request.
com.softwareag.is.interceptor.HttpInterceptorIFC	Defines a set of contracts that any inbound HTTP interceptor must implement.
com.wm.xmldata.XmlData	Class for creating, getting, and setting XmlData content.
com.wm.xmldata.XmlDataAnyTag	Class for creating keys that are used as XmlData *any element wildcards.
com.wm.xmldata.XmlDataAttributeTag	Class for creating XmlDataTags for use as attribute names.

Added API	Description
com.wm.xmldata.XmlDataConstants	Class implements static methods that server as shortcuts for various XmlData method names. By using a static import of this class, a program can significantly reduce the syntax clutter resulting from long class names and long method names.
com.wm.xmldata.XmlDataContentTag	Class that is a common superclass for the XmlDataTags that may appear in content models.
com.wm.xmldata.XmlDataCopy	Class that implements a structured copying utility for XmlData infosets.
com.wm.xmldata.XmlDataCursor	Class for navigating and manipulating the XMLData element content.
com.wm.xmldata.XmlDataElementTag	Class for creating keys that are used as XmlData element names
com.wm.xmldata.XmlDataException	Class that serves as the base class for all exceptions thrown by the XMLData feature.
com.wm.xmldata.XmlDataGroupTag	Class for constructing an XmlData key for group models.
com.wm.xmldata.XmlDataIteratorTag	Class that provides a standard implementation for creating XmlData iterator-tags.
com.wm.xmldata.XmlDataMap	Class that provides the direct manipulation of fields in an XmlData object without the use of cursors.
com.wm.xmldata.XmlDataPath	Class to facilitate the encoding or decoding of an XmlData path as a String literal.
com.wm.xmldata.XmlDataSubstitutionTag	Class for creating specialized element tags that are the heads of substitution groups.
com.wm.xmldata.XmlDataTreeCursor	Class that provides additional capabilities beyond those of the XmlDataCursor. In particular, repeating values can be presented as if flattened so that nextItem() and previousItem() position to individual items rather than arrays of items.
com.wm.xmldata.XmlDataTypeTag	Class for creating an XmlDataTypeTag and returning the value of an XMLDataTypeTag.
com.wm.xmldata.XmlNamespaceMap	Class that maintains a mapping between prefixes and namespace URIs.

Added API	Description
com.wm.xmldata.xmldocumenttype.SchemaConverterException	Class for throwing a SchemaConverterException.
com.wm.xmldata.xmldocumenttype.XmlDataSchemaConverter	Class that converts the scripted form of an XmlData document type specification to actual XmlData Document Type namespace nodes.
com.wm.xmldata.xmldocumenttype.XmlDataWorkspace	Class that provides a 'local' workspace in which XmlData document types and fields can be created on an ad hoc basis. This class also allows read-only access to the XmlData document types in the Integration Server namespace.

Removed API	Replacement, if any
com.wm.data.XmlData	com.wm.xmldata.XmlData
com.wm.data.XmlDataAnyTag	com.wm.xmldata.XmlDataAnyTag
com.wm.data.XmlDataAttributeTag	com.wm.xmldata.XmlDataAttributeTag
com.wm.data.XmlDataContentTag	com.wm.xmldata.XmlDataContentTag
com.wm.data.XmlDataCursor	com.wm.xmldata.XmlDataCursor
com.wm.data.XmlDataElementTag	com.wm.xmldata.XmlDataElementTag
com.wm.data.XmlDataException	com.wm.xmldata.XmlDataException
com.wm.data.XmlDataGroupTag	com.wm.xmldata.XmlDataGroupTag
com.wm.data.XmlDataSubstitutionTag	com.wm.xmldata.XmlDataSubstitutionTag
com.wm.data.obsolete.XmlData	com.wm.xmldata.XmlData
com.wm.data.obsolete.XmlDataCursor	com.wm.xmldata.XmlDataCursor

Release 10.0

Added API	Description
com.wm.app.b2b.server.jms.consumer. JMSTriggerGroupFacade	Class containing methods for creating, viewing, and deleting JMS trigger groups.

Release 9.12

Added API	Description
com.wm.data.obsolete.XmlData	Contains the 9.10 version of the com.wm.data.XmlData class. The com.wm.data.XmlData has been revised and simplified in 9.12.
com.wm.data.obsolete.XmlDataCursor	Contains the 9.10 version of the com.wm.data.XmlDataCursor class. The com.wm.data.XmlDataCursor has been revised and simplified in 9.12.

Changed API	Description
com.wm.data.XmlData	Revised to present a more consistent programming interface in which there are clearer distinctions between the underlying IData implementation of XmlData and the logical XmlData model as viewed by a typical application. Additionally, the API formalizes aspects of the IData implementation of XmlData to reduce the knowledge of the underlying Xm;Data implementation that is required to use the methods in the class. For a detailed list of changes to this class, see the webMethods Integration Server Java API Reference. Note: If you have any existing Java classes that use this class, you must modify your Java classes to either use new APIs introduced in com.wm.data.XmlData or change the Java imports in to reference the 9.10 version of the class which is now contained in com.wm.data.obsolete.XmlData.
com.wm.data.XmlDataCursor	Revised to present a more consistent programming interface in which there are clearer distinctions between the underlying IData implementation of XmlData and the logical XmlData model as viewed by a typical application. Additionally, the API formalizes aspects of the IData implementation of XmlData to reduce the knowledge of the underlying XmlData implementation that is required to use the methods in the class. For a detailed list of changes to this class, see the webMethods Integration Server Java API Reference. Note: If you have any existing Java classes that use this class, you must modify your Java classes to either use new APIs introduced in com.wm.data.XmlDataCursor or change the Java imports in to reference the 9.10 version of the class which is now contained in com.wm.data.obsolete.XmlDataCursor.

Release 9.10

Added API	Description

Added API	Description
com.wm.data.XmlData	Class for creating, getting, and setting XmlData content.
com.wm.data.XmlDataAnyTag	Class for creating keys that are used as XmlData *any element wildcards.
com.wm.data.XmlDataAttributeTag	Class for creating XmlDataTags for use as attribute names.
com.wm.data.XmlDataContentTag	Class that is a common superclass for the XmlDataTags that may appear in content models.
com.wm.data.XmlDataCursor	Class for navigating and manipulating the XmlData element content.
com.wm.data.XmlDataElementTag	Class for creating keys that are used as XmlData element names
com.wm.data.XmlDataException	Class that serves as the base class for all exceptions thrown by the XmlData feature.
com.wm.data.XmlDataGroupTag	Class for constructing an XmlData key for group models.
com.wm.data.XmlDataSubstitutionTag	Class for creating specialized element tags that are the heads of substitution groups.

12.0 Added, Removed, Deprecated, or Changed Administrator APIs

A release is listed in this section only if changes occurred in that release.

Release 10.7

Added API	Description
GET /admin/account/locking	Retrieves account locking settings.
PATCH /admin/account/locking	Updates account locking settings.
GET /admin/account/password/expiration	Retrieves password expiration settings.
PATCH /admin/account/password/expiration	Updates password expiration settings.
GET /admin/account/password/restriction	Retrieves password restriction settings.
PATCH /admin/account/password/restriction	Updates password restriction settings.
GET /admin/cachemanager/	Retrieves all cache managers.

Added API	Description
GET /admin/cachemanager/{cacheManagerName}	Retrieves a cache manager.
POST /admin/cachemanager/	Creates a cache manager.
POST /admin/cachemanager/{cacheManagerName}	Stops, starts, or restarts a cache manager.
PATCH /admin/cachemanager/{cacheManagerName}	Updates a cache manager.
PUT /admin/cachemanager/{cacheManagerName}	Replaces a cache manager.
DELETE /admin/cachemanager/{cacheManagerName}	Deletes a cache manager and all caches contained in the cache manager.
GET /admin/cachemanager/{cacheManagerName}/cache	Retrieves all caches for the requested cache manager.
GET /admin/cachemanager/{cacheManagerName}/cache/{cacheName}	Retrieves the requested cache.
POST /admin/cachemanager/{cacheManagerName}/cache	Creates a cache.
PATCH /admin/cachemanager/{cacheManagerName}/cache/{cacheName}	Updates a cache.
PUT /admin/cachemanager/{cacheManagerName}/cache/{cacheName}	Replaces a cache.
DELETE /admin/cachemanager/{cacheManagerName}/cache/{cacheName}	Deletes a cache and discards all data in the cache.
GET /admin/certificate/client	Retrieves all client certificate details.
GET /admin/certificate/client/issuer/{issuer}/usage/{usage}/serialNum/{serialNum}	Retrieves details for the requested client certificate.
POST /admin/certificate/client	Imports a client certificate.
PATCH /admin/certificate/client/issuer/{issuer}/oldUsage/{oldUsage}/serialNum/{serialNum}	Updates a client certificate.
PUT /admin/certificate/client/issuer/{issuer}/oldUsage/{oldUsage}/serialNum/{serialNum}	Replaces a client certificate.
DELETE /admin/certificate/client/issuer/{issuer}/oldUsage/{oldUsage}/serialNum/{serialNum}	Deletes a client certificate.
GET /admin/certificate/server	Retrieves server certificate settings.

Added API	Description
POST /admin/certificate/server/cache	Clears the cache of current SSL session information.
PATCH /admin/certificate/server	Updates a server certificate.
PUT /admin/certificate/server	Replaces a server certificate.
GET /admin/csrfguard	Retrieves CSRF guard settings
GET /admin/csrfguard/token	Retrieves CSRF secure token.
PATCH /admin/csrfguard	Updates CSRF guard settings
PUT /admin/csrfguard	Replaces CSRF guard settings.
GET /admin/fileaccesscontrol/	Retrieves file access control configuration from fileAccessControl.cnf.
PATCH /admin/fileaccesscontrol/	Updates the file access control configuration in fileAccessControl.cnf.
GET /admin/globalvariable	Retrieves all global variables.
GET /admin/globalvariable/{key}	Retrieves a specific global variable.
POST /admin/globalvariable	Creates a global variable.
PATCH /admin/globalvariable/{key}	Updates a global variable.
PUT /admin/globalvariable/{key}	Updates a global variable.
DELETE /admin/globalvariable/{key}	Deletes a global variable.
GET /admin/healthgauge/	Retrieves all health indicators.
GET /admin/healthgauge/{indicatorName}	Retrieves information about a particular health indicator.
PATCH /admin/healthgauge/{indicatorName}	Updates a health indicator.
POST /admin/healthgauge/{indicatorName}	Performs an administrative action on a health indicator.
GET /admin/hotdeployment/	Retrieves the hot deployment configuration
PATCH /admin/hotdeployment/	Updates the hot deployment configuration
GET /admin/jdbc/driver/	Retrieves all JDBC driver aliases.
POST /admin/jdbc/driver/	Creates a JDBC driver alias.
GET /admin/jdbc/driver/{driverAliasName}	Retrieves a JDBC driver alias.
PATCH /admin/jdbc/driver/{driverAliasName}	Updates a JDBC driver alias.
PUT /admin/jdbc/driver/{driverAliasName}	Replaces a JDBC driver alias.

Added API	Description
DELETE /admin/jdbc/driver/{driverAliasName}	Deletes a JDBC driver alias.
GET /admin/jdbc/function/	Retrieves all JDBC functions.
GET /admin/jdbc/function/{functionAliasName}	Retrieves a JDBC function.
PATCH /admin/jdbc/function/{functionAliasName}	Updates a JDBC function.
POST /admin/jdbc/function/{functionAliasName}	Performs administrative actions on a JDBC function.
GET /admin/jdbc/pool/	Retrieves all JDBC pool aliases.
GET /admin/jdbc/pool/{poolAliasName}	Retrieves a JDBC pool alias.
POST /admin/jdbc/pool/	Creates a JDBC pool alias.
PUT /admin/jdbc/pool/{poolAliasName}	Replaces a JDBC pool alias.
POST /admin/jdbc/pool/{poolAliasName}	Performs an administrative action on a JDBC pool alias
PATCH /admin/jdbc/pool/{poolAliasName}	Updates a JDBC pool alias.
DELETE /admin/jdbc/pool/{poolAliasName}	Deletes a JDBC pool alias.
GET /admin/jms	Retrieves all JMS connection aliases.
GET /admin/jms/{jmsAliasName}	Retrieves a specific JMS connection alias.
POST /admin/jms	Creates a JMS connection alias.
POST /admin/jms/{jmsAliasName}	Enables or disables a JMS connection alias.
PATCH /admin/jms/{jmsAliasName}	Updates a JMS connection alias.
PUT /admin/jms/{jmsAliasName}	Replaces a JMS connection alias.
DELETE /admin/jms/{jmsAliasName}	Deletes a JMS connection alias.
GET /admin/jmstrigger	Retrieves all JMS triggers.
GET /admin/jmstrigger/{triggerName}	Retrieves a specific JMS trigger.
POST /admin/jmstrigger/{triggerName}	Enables, disables, or suspends all JMS triggers, all JMS triggers of a particular type (standard or SOAP-JMS), or an individual JMS trigger.
GET /admin/jndi	Retrieves all JNDI connection aliases.
GET /admin/jndi/{jndiAliasName}	Retrieves information for a specific JNDI connection alias.
POST /admin/jndi	Creates a JNDI connection alias.

Added API	Description
POST /admin/jndi/{jndiAliasName}	Performs an administrative action on a JNDI connection alias.
PATCH /admin/jndi/{jndiAliasName}	Updates a JNDI connection alias.
PUT /admin/jndi/{jndiAliasName}	Replaces a JNDI connection alias.
DELETE /admin/jndi/{jndiAliasName}	Deletes a JNDI connection alias.
GET /admin/jwt/globalsettings	Retrieves the JWT global settings.
PATCH /admin/jwt/globalsettings	Updates the JWT global settings.
GET /admin/jwt/issuer/	Retrieves a list of all trusted issuers for JWT authentication
GET /admin/jwt/issuer/{issuer}	Retrieves a specific trusted issuer for JWT authentication.
POST /admin/jwt/issuer/	Creates a trusted issuer.
PATCH /admin/jwt/issuer/{issuer}	Updates a trusted issuer.
PUT /admin/jwt/issuer/{issuer}	Replaces a trusted issuer.
DELETE /admin/jwt/issuer/{issuer}	Deletes a trusted issuer.
DELETE /admin/jwt/issuer/{issuer}/issuerSkew	Deletes an issuer-skew mapping.
DELETE admin/jwt/issuer/{issuer}/truststoreAlias/{truststoreAlias}/certificateAlias/{certificateAlias}	Deletes an issuer-certificate mapping.
GET /admin/kerberos/	Retrieves the Kerberos configuration.
PATCH /admin/kerberos/	Updates the Kerberos configuration.
GET /admin/ldap/	Retrieves LDAP settings.
PATCH /admin/ldap	Updates LDAP settings.
GET /admin/ldap/dir/	Retrieves all LDAP directory configurations.
GET /admin/ldap/dir/{index}	Retrieves an LDAP directory configuration.
POST /admin/ldap/dir/	Creates an LDAP directory configuration.
PATCH /admin/ldap/dir/{index}	Updates an LDAP directory configuration.
PUT /admin/ldap/dir/{index}	Replaces an LDAP directory configuration.
DELETE /admin/ldap/dir/{index}	Deletes an LDAP directory configuration.
GET /admin/log/{logName}	Retrieves audit logger data.

Added API	Description
GET /admin/logger/	Returns a list of all audit loggers.
GET /admin/logger/{loggerName}	Retrieves an audit logger.
POST /admin/logger/{loggerName}	Performs an enable or disable action on an audit logger.
PATCH /admin/logger/{loggerName}	Updates an audit logger configuration.
PUT /admin/logger/{loggerName}	Replaces an audit logger configuration.
GET /admin/logger/server	Retrieves all the top-level components for which the server logger can be configured.
GET /admin/logger/server/{component}	Retrieves the server logger facility codes and logging levels for a particular component.
GET /admin/logger/server/{component}/{facilityCode}	Retrieves the logging level for a top-level component and the logging levels for all the facilities in the component.
PATCH /admin/logger/server	Updates Default logging level for Integration Server.
PATCH /admin/logger/server/{component}	Updates the default logging level for a top-level component.
PATCH /admin/logger/server/{component}/{facilityCode}	Updates the logging level for a facility code in a component.
GET /admin/messaging	Retrieves all webMethods messaging connection aliases.
GET /admin/messaging/{aliasName}	Retrieves a specific webMethods messaging connection alias.
POST /admin/messaging/	Creates a webMethods messaging connection alias.
POST /admin/messaging/{aliasName}	Enables or disables a webMethods messaging connection alias or sets an alias as the default webMethods messaging connection alias.
PATCH /admin/messaging/{aliasName}	Updates a webMethods messaging connection alias.
PUT /admin/messaging/{aliasName}	Replaces a webMethods messaging connection alias.
DELETE /admin/messaging/{aliasName}	Deletes a webMethods messaging connection alias.
GET /admin/messagingtrigger	Retrieves all webMethods messaging triggers.

Added API	Description
GET /admin/messagingtrigger/{triggerName}	Retrieves a particular webMethods messaging trigger.
POST /admin/messagingtrigger/{triggerName}	Activates or suspends message retrieval and/or message processing for a webMethods messaging trigger. Set triggerName to "all" to affect all webMethods messaging triggers
GET /admin/mqtt	Retrieves all MQTT connection aliases.
GET /admin/mqtt/{connectionAliasName}	Retrieves an MQTT connection alias.
POST /admin/mqtt/	Creates an MQTT connection alias.
POST /admin/mqtt/{connectionAliasName}	Performs administrative actions (enable or disable) on an MQTT connection alias.
PATCH /admin/mqtt/{connectionAliasName}	Updates an MQTT connection alias.
PUT /admin/mqtt/{connectionAliasName}	Replaces an MQTT connection alias.
DELETE /admin/mqtt/{connectionAliasName}	Deletes an MQTT connection alias.
GET /admin/mqtttrigger	Retrieves all MQTT triggers.
GET /admin/mqtttrigger/{triggerName}	Retrieves a specific MQTT trigger.
POST /admin/mqtttrigger/{triggerName}	Enables, disables, or suspends all or a specific MQTT trigger.
DELETE /admin/package/{packageName}	Deletes a package.
GET /admin/port	Retrieves all ports for Integration Server or for a package. This operation does not support the expand parameter.
GET /admin/port/diagnostic	Retrieves a link to the diagnostic port for Integration Server.
GET /admin/port/primary	Retrieves a link to the primary port for Integration Server.
POST /admin/port	Sets the primary port for Integration Server.
GET /admin/port/ipaccess	Retrieves the global IP access settings.
PUT /admin/port/ipaccess	Replaces the global IP access settings.
GET /admin/port/email	Retrieves all email ports.
GET /admin/port/email/{alias}	Retrieves an email port.
POST /admin/port/email	Creates an email port.
POST /admin/port/email/{alias}	Enables or disables an email port.

Added API	Description
PATCH /admin/port/email/{alias}	Updates an email port.
PUT /admin/port/email/{alias}	Replaces an email port.
DELETE /admin/port/email/{alias}	Deletes an email port.
GET /admin/port/email/{alias}/ipaccess	Retrieves the IP access settings for an email port.
PUT /admin/port/email/{alias}/ipaccess	Replaces the IP access settings for an email port.
DELETE /admin/port/email/{alias}/ipaccess	Deletes the IP access settings for an email port. Once deleted, the port will use the global IP access settings.
GET /admin/port/email/{alias}/resourceaccess	Retrieves the resource access settings for an email port.
PUT /admin/port/email/{alias}/resourceaccess	Replaces the resource access settings for an email port.
GET /admin/port/enterprisegateway/	Retrieves all Enterprise Gateway ports.
GET /admin/port/enterprisegateway/{alias}	Retrieves an Enterprise Gateway port.
POST /admin/port/enterprisegateway/	Creates an Enterprise Gateway port.
POST /admin/port/enterprisegateway/ {alias}	Enables or disables an Enterprise Gateway port.
PATCH /admin/port/enterprisegateway/ {alias}	Updates an Enterprise Gateway port.
PUT /admin/port/enterprisegateway/{alias}	Replaces an Enterprise Gateway port.
DELETE /admin/port/enterprisegateway/ {alias}	Deletes an Enterprise Gateway port.
GET /admin/port/enterprisegateway/ {alias}/ipaccess	Retrieves the IP access settings for an Enterprise Gateway port.
PUT /admin/port/enterprisegateway/ {alias}/ipaccess	Replaces the IP access settings for an Enterprise Gateway port.
GET /admin/port/internalserver/	Retrieves all Internal Server ports.
GET /admin/port/internalserver/{alias}	Retrieves an Internal Server port.
POST /admin/port/internalserver	Creates an Internal Server port.
POST /admin/port/internalserver/{alias}	Enables or disables an Internal Server port.
PATCH /admin/port/internalserver/{alias}	Updates an Internal Server port.
PUT /admin/port/internalserver/{alias}	Replaces an Internal Server port.

Added API	Description
DELETE /admin/port/internalserver/{alias}	Deletes an Internal Server port.
GET /admin/port/internalserver/{alias}/resourceaccess	Retrieves the resource access settings for an Internal Server port.
PUT /admin/port/internalserver/{alias}/resourceaccess	Replaces the resource access settings for an Internal Server port.
GET /admin/port/http/	Retrieves all HTTP ports.
GET /admin/port/http/{alias}	Retrieves an HTTP port.
POST /admin/port/http/	Creates an HTTP port.
PATCH /admin/port/http/{alias}	Updates an HTTP port.
PUT /admin/port/http/{alias}	Replaces an HTTP port.
DELETE /admin/port/http/{alias}	Deletes an HTTP port.
POST /admin/port/http/{alias}	Enables, disables, suspends, or resumes an HTTP port.
GET /admin/port/http/{alias}/ipaccess	Retrieves the IP access settings for an HTTP port.
PUT /admin/port/http/{alias}/ipaccess	Replaces the IP access settings for an HTTP port.
DELETE /admin/port/http/{alias}/ipaccess	Deletes the IP access settings for an HTTP port. Once deleted, the port will use the global IP access settings.
GET /admin/port/http/{alias}/resourceaccess	Retrieves the resource access settings for an HTTP port.
PUT /admin/port/http/{alias}/resourceaccess	Replaces the resource access settings for an HTTP port.
GET /admin/port/https/	Retrieves all HTTPS ports.
GET /admin/port/https/{alias}	Retrieves an HTTPS port.
POST /admin/port/https/	Creates an HTTPS port.
PATCH /admin/port/https/{alias}	Updates an HTTPS port.
PUT /admin/port/https/{alias}	Replaces an HTTPS port.
DELETE /admin/port/https/{alias}	Deletes an HTTPS port.
POST /admin/port/https/{alias}	Enables, disables, suspends, or resumes an HTTPS port.
GET /admin/port/https/{alias}/ipaccess	Retrieves the IP access settings for an HTTPS port.
PUT /admin/port/https/{alias}/ipaccess	Replaces the IP access settings for an HTTPS port.

Added API	Description
DELETE /admin/port/https/{alias}/ipaccess	Deletes the IP access settings for an HTTPS port. Once deleted, the port will use the global IP access settings.
GET /admin/port/https/{alias}/resourceaccess	Retrieves the resource access settings for an HTTPS port.
PUT admin/port/https/{alias}/resourceaccess	Replaces the resource access settings for an HTTPS port.
GET /admin/port/websocket	Retrieves all WebSocket and WebSocketSecure ports.
GET /admin/port/websocket/{alias}	Retrieves a WebSocket or WebSocketSecure port.
POST /admin/port/websocket	Creates a WebSocket or WebSocketSecure port.
POST /admin/port/websocket/{alias}	Enables or disables a WebSocket or WebSocketSecure port.
PATCH /admin/port/websocket/{alias}	Updates a WebSocket or WebSocketSecure port.
PUT /admin/port/websocket/{alias}	Replaces a WebSocket or WebSocketSecure port.
DELETE /admin/port/websocket/{alias}	Deletes a WebSocket or WebSocketSecure port.
GET /admin/port/websocket/{alias}/ipaccess	Retrieves the IP access settings for a WebSocket or WebSocketSecure port.
PUT /admin/port/websocket/{alias}/ipaccess	Replaces the IP access settings for a WebSocket or WebSocketSecure port.
DELETE /admin/port/websocket/{alias}/ipaccess	Deletes the IP access settings for a WebSocket or WebSocketSecure port. Once deleted, the port will use the global IP access settings.
GET /admin/port/websocket/{alias}/resourceaccess	Retrieves the resource access settings for a WebSocket or WebSocketSecure port
PUT /admin/port/websocket/{alias}/resourceaccess	Replaces the resource access settings for a WebSocket or WebSocketSecure port.
GET /admin/proxy/	Retrieves all proxy server aliases.
GET /admin/proxy/{proxyAlias}	Retrieves a proxy server alias.
POST /admin/proxy/	Creates a proxy server alias.
POST /admin/proxy/{proxyAlias}	Performs administrative actions on a proxy server alias.
PATCH /admin/proxy/{proxyAlias}	Updates a proxy server alias.
PUT /admin/proxy/{proxyAlias}	Replaces a proxy server alias.

Added API	Description
DELETE /admin/proxy/{proxyAlias}	Deletes a proxy server alias
GET /admin/quiesce/	Retrieves the current server mode (active or quiesce) and the quiesce port.
POST /admin/quiesce/	Enters the Integration Server into quiesce mode from active mode or exits it from quiesce mode to active mode.
POST /admin/quiesce/port	Sets the specified port as the quiesce port.
GET /admin/scheduledtask/	Retrieves all scheduled user tasks.
GET /admin/scheduledtask/{taskId}	Retrieves a scheduled user task.
POST /admin/scheduledtask/	Creates a scheduled task.
POST /admin/scheduledtask/{taskId}	Suspends, wakes up, or cancels a scheduled task.
PATCH /admin/scheduledtask/{taskId}	Updates a scheduled task
PUT /admin/scheduledtask/{taskId}	Replaces a scheduled task.
DELETE admin/scheduledtask/{taskId}	Deletes a scheduled task.
GET /admin/scheduler	Retrieves the current scheduler state (running or paused) and a list of scheduled tasks.
POST /admin/scheduler	Resumes or pauses scheduler.
GET /admin/server/httpout	Retrieves outbound HTTP settings.
PATCH /admin/server/httpout	Updates outbound HTTP settings.
GET /admin/server/memory	Retrieves available memory information for Integration Server
GET /admin/server/notification	Retrieves email notification settings.
PATCH /admin/server/notification	Updates email notification settings.
GET /admin/server/session	Retrieves the session settings.
PATCH /admin/server/session	Updates the session settings.
GET /admin/server/setting	Retrieves values for all extended settings in Integration Server.
GET /admin/server/setting/{settingName}	Retrieves the value of a specific extended setting.
POST /admin/server/setting	Resets specified extended settings to their default values.
PATCH /admin/server/setting/{settingName}	Updates an extended setting.
PATCH /admin/server/setting/	Updates multiple extended settings

Added API	Description
PUT /admin/server/setting/	Updates multiple extended settings.
GET /admin/server/sso	Retrieves the MWS single sign-on setting
PATCH /admin/server/sso	Updates the MWS single sign-on setting
GET /admin/server/threadpool	Retrieves server threadpool settings.
PATCH /admin/server/threadpool	Updates server threadpool settings.
GET /admin/sftpserver	Retrieves all SFTP server aliases.
GET /admin/sftpserver/{alias}	Retrieves a specific SFTP server alias.
POST /admin/sftpserver	Creates an SFTP server alias.
PATCH /admin/sftpserver/{alias}	Updates an SFTP server alias.
PUT /admin/sftpserver/{alias}	Replaces an SFTP server alias.
DELETE /admin/sftpserver/{alias}	Deletes an SFTP server alias.
GET /admin/sftpuser	Retrieves all SFTP user aliases.
GET /admin/sftpuser/{alias}	Retrieves a specific SFTP user alias.
POST /admin/sftpuser	Creates an SFTP user alias.
PATCH /admin/sftpuser/{alias}	Updates an SFTP user alias.
PUT /admin/sftpuser/{alias}	Replaces an SFTP user alias.
DELETE /admin/sftpuser/{alias}	Deletes an SFTP user alias.
GET /admin/user/	Retrieves all user accounts.
GET /admin/user/{username}	Retrieves a user account.
POST /admin/user/	Creates a user account.
POST /admin/user/{username}	Enables or disables a user account.
PATCH /admin/user/{username}	Updates a user account.
PUT /admin/user/{username}	Replaces a user account.
DELETE /admin/user/{username}	Deletes a user account.

Changed API	Description
GET /admin/server	Now returns information indicating if and why a server restart is needed for configuration changes to take effect.

Changed API	Description
POST /admin/package/{packageName}	Now supports the administrative actions archive and recover.

Release 10.5

Added API	Description
GET /admin/license/	Retrieves license information.
POST /admin/license/	Updates Integration Server license key file or Terracotta license file.
PUT /admin/license/	Updates Integration Server license key file or Terracotta license file.
PATCH /admin/license/	Updates Integration Server license key file or Terracotta license file.
GET /admin/package	Retrieves all packages.
GET /admin/package/{packageName}	Gets package information for a package.
POST /admin/package/{packageName}	Performs an administrative action on a package where the action can be: disable, enable, reload, or activate.
GET /admin/server	Retrieves information about Integration Server.
POST /admin/server	Performs administrative actions on Integration Server where the action can be: stop, restart, restartQuiesce, quiesce, exitQuiesce.
GET /admin/server/diagnostics	Returns a diagnostic archive as a ZIP attachment.
GET /admin/server/updates	Retrieves information about fixes installed on Integration Server.
GET /admin/swagger/{productName}	Retrieves the Swagger document for a product's API.
GET /admin/swagger	Retrieves the Swagger documents for all administrative APIs in Integration Server.

13.0 Copyright Information

and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses. http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

14.0 Support

Visit the Empower website to learn about support policies and critical alerts, read technical articles and papers, download products and fixes, submit feature/enhancement requests, and more.

Visit the TECHcommunity website to access additional articles, demos, and tutorials, technical information, samples, useful resources, online discussion forums, and more.

IS-RM-107-20210307