Contents

[IS-Classloading 6](#_Toc7950594)

[Server Configuration 16](#_Toc7950595)

[Licensing Info: 16](#_Toc7950596)

[Managing Server Thread Pool: 16](#_Toc7950597)

[Setting Remote IS: 16](#_Toc7950598)

[Managing Server Sessions: 18](#_Toc7950599)

[Configuring Outbound HTTP Settings: 19](#_Toc7950600)

[Extended Configuration Settings : 19](#_Toc7950601)

[Specifying the JDK or JRE for Integration Server 20](#_Toc7950602)

[JDBC Pool 20](#_Toc7950603)

[User Management 26](#_Toc7950604)

[Securing Communications with the Server 30](#_Toc7950605)

Every Integration Server is installed with a predefined user account called "Administrator." and predefined password "manage."

Administrative Responsibilities

* **Installing and upgrading the server**
  + Equipping the server computer with appropriate hardware and software,
  + Downloading and installing the server program, and
  + Implementing upgrades as needed
* **Starting and stopping the server**,
  + Shutting down the server for routine maintenance or reconfiguration and restarting it afterwards.
  + Performing your site's standard recovery procedures following a hardware or software failure of the server computer.
* **Configuring server settings** : setting basic operating parameters such as
  + the maximum session limits,
  + log file options, and
  + Port assignments.
* **Administering users and groups**,
  + Defining user names and passwords for authorized users and assigning them to groups.
  + Configuring the server to acquire user and group information from an external system (e.g., LDAP)
* **Administering server security**,
  + Identifying other administrators,
  + Assigning access controls to individual services, and
  + Configuring the server's use of digital certificates.
* **Managing packages and services,** 
  + activating/deactivating/copying packages and
  + updating services and/or packages as necessary.
* **Administering multiple instances of the server**, which includes performing all or some of the activities listed above to manage two or more Integration Servers running on the same machine

**The Integration Server Administrator**

The Integration Server Administrator is the utility used to accomplish administrative tasks. It is used

* to monitor server activity,
* examine log information,
* add users,
* enable/disable services, and
* Adjust the server's performance features.

The Integration Server issues e-mail messages for a variety of failure conditions (internal errors, binding errors, and transaction manager errors).

Administrator, should receive these messages and take appropriate action when errors occur.

you must set the E-mail Notification parameters.

Adding Backup Administrators : To add a backup administrator to your server,

* Create a regular user account for the user (if he or she does not already have one);
* Then add that user account to the "Administrators" group.

**Integration Server Instances**

* The Software AG installer creates the instance under the SoftwareAG\_directory\**IntegrationServer\instances** directory
  + C:\SAG\SAG912\IntegrationServer\instances\**default**
* When you install webMethods Integration Server, you specify a name for the initial instance.
  + The default name of the initial instance is “default”.
* You can create and run multiple instances of the server on one machine.
  + Using the scripts provided by Integration Server or
  + Through Software AG Command Central.
* Each instance has a home directory under Software AG\_directory\IntegrationServer\instances
  + That contains its own packages, configuration files, log files, and updates.
  + You administer and apply packages and updates to each Integration Server instance separately.
  + You can apply the latest fixes using the Software AG Update Manager.
* The **Software AG\_directory\IntegrationServer** directory is the parent directory for all server instances you create.
  + It contains common and shared files that all server instances use, such as common jar files and fixes.

**IS-Package overview:**

The webMethods Integration Server hosts packages that contain services and related files. These packages are of two types:

* g : These are provided as part of Integration Server.
  + packages contain built-in services that
  + your developers(or client applications) invoke them from their services and
  + Services demonstrate some of the features of the Integration Server.
* Created by users : You can create additional packages(through Designer) to hold the services that your developers create

WebMethods-IS provides an environment for the orderly, efficient, and secure execution of services.

Each package has its own class loader.

**How the Server Executes Services**

* The server authenticates the client.
* If a session already exists for the client, the server uses the existing session.
  + If one does not exist, the server creates a session.
* The server determines the content-type of the service request so it can prepare data for the requested service.
* The server uses the supplied service name to look up the service.
* Requested service is being controlled based on the port on which the request came in.
  + If there is no restriction, the server continues with the execution of the service.
* Checks if requested HTTP method is allowed for the service.
  + If it is not, the server sends an back an error message,
  + Else the server continues with the execution of the service.
* Server looks up the Access Control List (ACL) for the service **to check access of client**.
  + If the ACL indicates that the client is allowed to access the service, the server continues with the execution of the service.
* If auditing is enabled, the server adds an entry to the Audit Log to mark the start of the request.
* The server starts gathering service statistics for the service.
* The server checks the service-results cache.
  + If service results are cached and the inputs for the cached results match the inputs for this request, the server returns the cached results.
  + If matching results are not cached, the server invokes the service.
  + If the service is a flow service, which can consist of several services, it invokes each service in the flow.
* The server ends the gathering of server statistics for the service.
* If auditing is enabled, the server adds an entry to the Audit Log to mark the end of the request.
* The server encodes the service results as specified by the content type.
* The server returns the results to the client.

# IS-Classloading

Integration Server employs two kinds of class loaders to locate the executable bytecode of the class and brings it into the JVM:

* OSGi bundle class loader
  + The OSGi Bundle classloader is the parent classloader for Integration Server.
  + This class loader is provided by the OSGi framework (Eclipse Equinox) and
  + Shipped with webMethods Integration Server.
* Integration Server class loaders
* IS Server Class Loader loads the classes that comprise the core of Integration Server.
  + This loader loads from the Server Classpath.
* Integration Server Package Class Loaders load Integration Server packages
  + These packages include those created by users through Designer and
  + Predefined packages that are provided as part of Integration Server.
  + Each package has its own class loader.

**Note**:

The OSGi Bundle classloader cannot access any jars specified in the CLASSPATH environment variable

to access jars specified in either the CLASSPATH, you must add the following entry

osgi.parentClassloader=app

In the Software AG\_directory\profiles\IS\_instance\_name\config.ini file and restart Integration Server

A service from PackageA calls a class that resides in PackageX (and known to Integration).

* The manifest.v3 files for both PackageA and PackageX specify "server" as the class loader
  + So package class loader defer the search to its parent Integration Server class loader.
* A service in PackageA calls a class that has not yet loaded into memory.
* PackageA's class loader passes the request to its parent class loader, i.e. Integration Server class loader.
* The Integration Server class loader passes the request up to the OSGi bundle class loader.
* The OSGi bundle class loader does not find the class. and request comes back to the Integration Server class loader, which searches for the class,
  + First in cache, and then
  + In the Integration Server classpath.
* The Integration Server class loader does not find the class. and The request comes back to PackageA's class loader
* PackageA's class loader searches for the class,
  + First in cache, and then
  + In the directory IntegrationServer\instances\instance\_name\packages\package\_name in this order:
    - PackageA\code\jars
    - PackageA\code\classes
    - PackageA\lib
    - PackageA\resource folders
* PackageA's class loader does not find the class and delegates the search to PackageX's class loader.
* PackageX's class loader searches for the class
  + in cache and
  + then in:
    - PackageX\code\jars
    - PackageX\code\classes
    - PackageX\lib
    - PackageX\resource folders
* PackageX's class loader finds the class in PackageX's resource folder.

Classpath : A classpath is a list of directories to be searched.

Integration Server uses the Integration Server classpath.

When you start Integration Server, startup.bat/sh runs.

the startup.bat/sh file builds an Integration Server classpath variable (How?)

How/Where to Specify Integration Server Classpaths:

* Software AG\_directory/profiles/IS\_instance\_name/configuration/custom\_wrapper.conf
  + wrapper.java.additional.202=-Dwatt.server.prepend.classes=
    - directories to prepend to the beginning of the classpath
  + wrapper.java.additional.203=-Dwatt.server.append.classes=
    - directories to append to the end of the classpath
* Software AG\_directory\IntegrationServer\instances\instance\_name \bin\ini.cnf
  + Specifies classes in the following directories, in this order:
    - IntegrationServer\lib
    - SoftwareAG\common\lib
* Software AG\_directory\IntegrationServer\instances\instance\_name \lib\jars\custom
  + Use this directory to store your custom or third party .jar/.zip files that you want to make available to a server specific instance.
* Software AG\_directory\IntegrationServer\instances\instance\_name \packages\package\_name \code\jars\static
  + Contained in this directory will be loaded even if the associated package is disabled.
* Software AG\_directory\IntegrationServer\instances\instance\_name \packages\package\_name \code\libs : If this directory exists, then:
  + code\classes is included, if it exists.
  + code\classes.zip is included, if it exists.
* Software AG\_directory\IntegrationServer\instances\instance\_name\updates
  + Valid, and non-voided updates and fixes.
* Software AG\_directory\IntegrationServer\lib\jars
  + All .jar and .zip files in this directory.
* Software AG\_directory\IntegrationServer\lib\jars\custom
  + This directory store custom and third party .jar and .zip files that is available to all server instances.
* Software AG\_directory\IntegrationServer\updates
  + The updates and fixes in this directory are available to all server instances.

Changing Classpath Information at Startup

When Integration Server starts, it checks the custom\_wrapper.conf files for modifications

to some of the Java system parameters and variables used by the server.

If the custom\_wrapper.conf file contains overrides to the classpath information, the server uses those settings during startup.

The custom\_wrapper.conf file contains following variables that you can change to add directories to the Java and Integration Server classpaths.

Beginning of the classpath: wrapper.java.additional.202=-Dwatt.server.prepend.classes=

End of the classpath: wrapper.java.additional.203=-Dwatt.server.append.classes=

**Integration Server security**

Integration Server’s security mechanisms can be configured to

* Prevent its unauthorized administration,
* Prevent data from being intercepted during transmission, and
* Protect Integration Server services from unauthorized access.

You can configure IS to:

* Use an Enterprise Gateway Server to intercept requests from external clients before passing the requests to your Integration Server.
* This allows you to isolate Integration Server behind an internal firewall.
* Require clients to present valid credentials (user-name/password or a client-certificate) to authenticate a connection.
* Authorize access to individual services by user groups, through use of ACLs that you associate with a service.
* For the greatest security, associate all services with an ACL.
* Provide transport-level security through SSL, and
* messagelevel security for web services through WS-Security.
* Digitally sign documents and verify digital signatures.
* Control access to services based on the port through which a service request is received.
* Restrict who can access Integration Server Administrator, and who can use
* Software AG Designer to connect to the Integration Server.
* Require clients to present valid user names (with passwords) that have Administrator privileges before allowing access to the IS-Administrator functions.
* Simplify security administration by storing Integration Server SSL certificates and private keys in industry-standard keystore files.
* Allow different client certificates to be used for different connections.

Integration Server security also depends on the security of its underlying OS. Make sure you do the following:

* Follow all vendor recommendations for secure configuration.
* Remove unnecessary network services that may contain security flaws.
  + telnet.
* Regularly check for and install updates and patches from the operating system vendor that might affect security.

To start Integration Server on Windows

Start ->All Programs->Software AG(SAG912) ->Start Servers ->Start Integration Server ->Click Start instanceName (Start default).

Starting a Server Instance from the Command Prompt

cd Software AG\_directory\profiles\IS\_instance\_name\bin

startup.bat -switch -switch ...

* -port portNumber : This switch overrides the value assigned to watt.server.port.
* -debug level : level of server log for this session.
  + value of level is Fatal, Error, Warn, Info, fatal, Debug, error and Trace
* -log destination : overrides the value (of server log information)assigned to watt.debug.logfile.
  + default value is serveryyyymmdd .log
  + Destination could be
    - destination (filename) Specify the fully qualified path to the file
    - none Display server log information on the computer screen.
* -quiesce Specifies to start the server in quiesce mode.

What Happens When You Start the Server?

It performs a series of initialization steps to make itself ready for client requests.

* Establishes the operating environment by using the configuration parameters.
  + Configuration parameters location instances\instance\_name\config\server.cnf.
* Initializes processes that perform internal management.
* Loads information about all the enabled packages and their services
  + Package location - \instances\instance\_name\packages directory.
  + If a package depends on other packages, the prerequisite packages are loaded first.
  + The server does not load disabled packages.
* Executes the startup services for each loaded package.
* Initializes the guaranteed delivery engine.
  + The server checks the job store for pending guaranteed delivery transactions.
  + It retries the pending transactions as the guaranteed delivery configuration settings specify.
* Schedules internal system tasks

IS as a Windows service

* You do not have to manually restart Integration Server following a machine restart.
* Integration Server automatically initialize when the machine on which it is installed initializes.

IS as a Windows application

* When Integration Server is a Windows application, you must manually start it.
* you can control when the Integration Server initializes.

Switching IS from a Windows Service to a Windows Application

* If the Windows service is running, stop it.
  + using the Integration Server Administrator to shut down the Integration Server or
  + from the Services dialog box in the Microsoft Windows Control Panel.
* Open a command prompt, for Software AG\_directory\profiles\IS\_instance\_name\bin and Execute
  + service.bat -remove

Switching IS Windows Application to a Windows Service

* Edit the Software AG\_directory\profiles\IS\_instance\_name\configuration\custom\_wrapper.conf file to fit your environment.
* Open a command prompt, for Software AG\_directory\profiles\IS\_instance\_name\bin and Execute
  + service.bat -install
  + Verify in the Control Panel (in the Services dialog box), that the service for IS has been created.

**custom\_wrapper.conf**

Software AG\_directory\profiles\IS\_instance\_name\configuration\custom\_wrapper.conf

Contains configuration settings, which is used at IS startup.

These settings can be overridden for a single Integration Server session by starting Integration Server from the command prompt and using switches.

You can permanently change some of the configuration values in the cusotm\_wrapper.conf so that all sessions use the modified configuration values.

Passing Java System Properties to IS

* Add a **wrapper.java.additional.n** property that specifies the property name and value
  + wrapper.java.additional.11=-Dmy.prop1=value1
  + n is a unique sequence number.
  + The property name must be preceded by -D.

Adding switch command

* This required two wrapper.app.parameter.n property need to be added in sequence
  + wrapper.app.parameter.n=switch
    - **switch** is the switch command.
  + wrapper.app.parameter.n=switch\_parameter
    - **switch\_parameter** is the value of the switch.
  + n is the next unused sequential number for the wrapper.app.parameter properties
* For example, to change the default port number to 8080, you would enter the following to custom\_wrapper.conf:
  + wrapper.app.parameter.7=-port
  + wrapper.app.parameter.8=8080

In the custom\_wrapper.conf, update the wrapper.app.parameter**.2** property to reflect the total number of wrapper.app.parameter properties

Save and close the file.

Restart the server for the changes to take effect.

**Server-Log**

The Integration Server log contains information about

* Operations and errors that occur on Integration Server,
* Starting of Integration Server subsystems and
* Loading of packages belonging to Integration Server or other webMethods products

Entries are written to the server log by Integration Server's major subsystems, called facilities.

* IS package facility writes server log entries when it loads and unloads packages,
* The IS flow manager facility writes server log entries when it processes a flow service, and
* Integration Server's HTTP ports write server log entries when they receive requests.

Integration Server is product and package is facility

Because facilities inherit logging levels from the parent product and, in turn, products inherit logging levels from the Default node,

Plan: set the logging level of the Default node to be the default logging level used by most of the products and facilities

Then, change the logging levels for the particular products or facilities

* IS Administrator -> Settings -> Logging
* Server Logger -> Click Edit Server Logger.
* Change the level and Click Save Changes.

By default, all facilities write to the server log,

* You can have only selected facilities write to the log
  + Set the level off
* You can increase or decrease the amount of data the facilities provide.
  + With setting different levels

**Queue Server Log Entries**

By default, Integration Server queues log entries written by its facilities in memory then uses a background thread to write them to the server log.

* Pros: Using a background thread improves Integration Server performance
* Con: Integration Server shuts down abnormally, all log entries in the queue will be lost. For better quality of service, do not queue the entries.

To specify whether to queue server log entries

* IS Administrator, Settings -> Extended -> Show and Hide Keys click.
* From list of the IS configuration properties Select the watt.server.log.queued
* Save Changes.
  + IS Administrator displays the property in the Extended Settings
* Click Edit Extended Settings
* To queue server log entries set watt.server.log.queued = true
* else set it as false
* Save Changes and Restart Integration Server.

To change the number of entries allowed in the server log queue, use the

watt.server.serverlogQueueSize parameter. The default size is 8192 bytes.

To Change the Server Log Location set watt.debug.logfile property to new location.

Default log file location is Integration Server\_directory\instances\instance\_name\logs

**Configuring the Server Log to Rotate Based on Size**

By default, Integration Server writes server log entries for the current day (defined as midnight to midnight) to the server.log file.

To specify the size limit for the server log. use watt.server.serverlogRotateSize server configuration parameter.

* Integration Server rotates the server.log when the server.log file size reaches that size or at midnight, whichever occurs first.
* The archive file name uses the format server.log\_yyyyMMdd\_HHmmssSSSZ ,
  + Where yyyyMMdd\_HHmmssSSSZ is the date and time the log file was created.
* There is no default value
* If no value or an invalid value is specified, Integration Server rotates the server.log file at midnight only.

**Limiting the Number of Server Log Files**

By default, Integration Server keeps each server log file that it creates.

To limit the number of server log files IS provides the **watt.server.serverlogFilesToKeep** server configuration parameter.

* When Integration Server reaches the limit for the number of server log files, Integration Server deletes the oldest archived server log file
* If you set watt.server.serverlogFilesToKeep to 0, or any value less than 1, Integration Server keeps an unlimited number of server log files.
* If you set watt.server.serverlogFilesToKeep to n, Integration Server keeps the current server log file (server.log) and up to n-1 archived server log files.

# Server Configuration

## Licensing Info:

* Viewing Licensing
  + Settings > License > License Details
* To change the License Key
  + Settings > License > License Details > Edit
* Viewing Active Sessions
  + Server > Statistics
    - Total Sessions - displays the current number of active sessions
    - Licensed Sessions - maximum number of licensed sessions your license allows

Approximately 30 days before your license expires, Integration Server sends an e-mail message to the administrative-message recipient, reminding him or her to renew the license.

If you need to obtain a new key or renew your license, contact your Software AG sales representative.

## Managing Server Thread Pool:

The server uses threads to execute services, retrieve documents from the messaging provider, and execute triggers.

When the server starts, the thread pool initially contains the minimum number of threads.

The server adds threads to the pool as needed until it reaches the maximum allowed.

If this maximum number is reached, the server waits until processes complete and return threads to the pool before beginning more processes.

To configure the server thread pool: Settings > Resources > Edit Resource Settings

Under “Server Thread Pool”, update the server thread pool settings

* Maximum Threads - default is 75.
* Minimum Threads - default is 0.
* Available Threads Warning Threshold - default is 15%.
* Scheduler Thread Throttle - Percentage of server threads the scheduler function is permitted to use. The default is 75%.

## Setting Remote IS:

Settings > Remote Servers

You can set up aliases for remote servers.

Communication through the alias is optimized, making transactions with the remote server faster.

**Use-Case:**

* Performing package replication.
  + To pull a package from the publisher, you must define the publishing server as a remote server to the subscriber.
  + The alias tells the subscribing server how to connect to the publishing server to set up the subscription or pull the package.
* Invoking services on other IS.
  + you can use the pub.remote:invoke and pub.remote.gd:\* services to invoke services on remote servers by identifying the remote servers by their aliases.
* Presenting multiple client certificates.
  + The IS can present a single client certificate to all servers or
  + It can present different client certificates to different SSL servers.
  + In addition, the IS can present certificates provided for this purpose by other organizations.

Setting up remote aliases for these SSL servers makes it easier to present different certificates to them.

The definition for an alias contains the connection information the server requires to connect to a remote server.

The alias also identifies a user name and password that the server supplies to the remote server.

The remote server uses the user name and password to authenticate the client and to determine if the client is authorized to execute the requested service.

to prevent unauthorized users from accessing services on remote servers, the alias also contains access control information.

You specify an ACL that protects the use of the alias.

If a client that is not authorized to use the alias makes a request, the server rejects the request and does not invoke the service on the remote server.

**Operations Supported:**

* **Adding an Alias** : Settings > Remote Servers. Click Create Remote Server Alias and provide properties.
* **Editing an Alias** : Settings > Remote Servers. Click on the alias name and Update the information
* **Deleting an Alias** : Settings > Remote Servers. Click the icon in the Delete field for alias to be deleted.
* **Testing the Connection** to a Remote Server : Settings > Remote Servers. Click the icon in the Test column.

**Important properties:**

* Execute ACL :
  + ACL that governs which user groups on your server can use this alias for the remote server.
  + Select an ACL from the drop down list. By default, only members of groups governed by the Internal ACL can use this alias.
* Max Keep Alive Connections
  + Sets the default number of client keep alive connections to retain for a given target endpoint.
  + If not specified, five keep alive connections are retained.
  + This field specifies the maximum number of client connections that should be retained for anygiven remote host.
  + In other words, this is not a maximum number of connections that can be established,
  + but a limit on the number of inactive client connections to retain for reuse.
* Keep Alive Timeout: (in minutes)
  + Specifies the length of time that your server maintains an idle connection to a remote server.
  + This value will cause the connection to be retained for possible reuse until it times out.
  + If the specified keep alive timeout value expires, the connection will close and the HTTP Listener will attempt to create a new one.
* Retry Server: Host name of a remote server you want your local IS to connect to if the primary remoteserver is unavailable.
  + The retry server will use the same port as the primary remote server.
  + If the remote server is part of a cluster, the local IS will, by default, try to connect to other IS in the cluster before trying to connect to the retry server.
  + If clients are using the pub.remote:invoke service to run services on a remote server, it is possible to change this default behavior by using the $retryCluster input parameter with the service.
  + If you set this parameter to false, the service will not try to use other Integration Servers in the cluster.
  + Instead, the service will immediately try using the retry server specified on this screen.

## Managing Server Sessions:

Integration Server starts a new session for every remote client that connects to it.

Setting the Session Timeout Limit : If a session is idle for a long period of time, so need to terminate

Settings > Resources > Edit Resource Settings > Under Session

* Session Timeout - enter maximum number of minutes an idle session can remain active.
* Enable Stateful Session Limit - When enabled, the number of stateful sessions that can exist simultaneously on the server is defined by the "Maximum Stateful Sessions"
* Maximum Stateful Sessions - The maximum number of concurrent stateful sessions that can exist on the Integration Server.
  + If this no. is crossed the server rejects the request and returns the error message.;
* Available Stateful Sessions Warning Threshold - Threshold at which Integration Server starts to warn (as server log message)of insufficient available stateful sessions.
  + The default is 25%

## Configuring Outbound HTTP Settings:

Requests that Integration Server issues on behalf of a client

The parameters control behavior such as:

* how long the server waits for a response,
* how many times it retries a failed request, and so forth.

Settings > Resources > Edit Resource Settings > Under Outbound HTTP Settingsas

* User Agent - The string that you want the server to supply in the HTTP User Agent header if the client does not specify a value.
* Maximum Redirects- number of times to allow a request to be redirected before the server returns an I/O exception to the client.
* Timeout - the number of seconds the server waits for a response from the target server before retrying the service or returning a timeout error to the client.
* Retries - number of times the server retries a service that has timed out

## Extended Configuration Settings :

special server property settings which are specified in C:\SAG\SAG912\IntegrationServer\instances\default\config\server.cnf file.

Settings > Extended > Show and Hide Keys

* Select(check) properties to be displayed and click save.
* The server displays the Extended Settings screen with the selected properties with their default values.
* To add, delete, or change a property setting, click Edit Extended Settings (Settings > Extended) and type your changes.

You can remove watt properties from "Extended Settings screen", simply by uncheking it from "Show and Hide Keys" windows.

Another way it to remove it from "Extended Settings screen" and save the changes. By doing this watt properties will be deleted from server.conf file as well. This looks little weird.

## Specifying the JDK or JRE for Integration Server

specify JDK, If you intend to use Designer to develop and compile Java services on Integration Server

specify JRE, If you will not be using this installation of Integration Server to compile Java services,

The JDK and JRE installed with Integration Server are required to run the Software AG Uninstaller, so don't remove it even if you specify a different JDK or JRE.

To specify the Java location for IS Open Software AG\_directory\profiles\IS\_instance\_name \configuration\wrapper.conf file

wrapper.java.command=C:\SoftwareAG\jvm\jvm\bin\java

save the file and restart the server.

also change the value of the

Also change watt.server.compile configuration parameter

To change the heap size Open Software AG\_directory\profiles\IS\_instance\_name \configuration\custom\_wrapper.conf file

wrapper.java.initmemory=256

wrapper.java.maxmemory=512

save the file and restart the server.

# JDBC Pool

Settings > JDBC Pools

The JDBC Pools screen enables you to manage the following:

* Functional aliases
* Pool aliases
* Driver aliases

Managing Pool Aliases

Defines the location and connection parameters for the database.

It uses any of the driver available under driver-alias for configuring connection.

Operations available:

Create a new connection pool

Create a connection pool by copying an existing pool

Edit/Test and Deelete Created pool.

File location : C:\SAG\SAG\_102\IntegrationServer\instances\default\config\jdbc\pool

Managing Driver Alias

The driver alias identifies a database driver to Integration Server. Integration Server uses

this database driver to connect to the defined database resource. The Integration Server

Administrator lists the available driver aliases in the Driver Alias Definitions area of the

JDBC Pools screen.

Operations available:

Create a new Database Driver Alias

Edit/Test and Deelete Created Database Driver Alias.

C:\SAG\SAG\_102\IntegrationServer\instances\default\config\jdbc\driver

While installing IS using SAG installer, when you choose external RDBMS and supply the database connection the installer creates a JDBC connection pool and configures Integration Server to write the following data (called functional alias)

* IS Internal
* IS Core Audit Log
* Cross Reference
* Distributed Locking
* Document History
* Process Audit Log
* Process Engine

You need to have respective tables in configured DB to capture above data.

To create these tables run ddl scripts under respective folders of C:\SAG\SAG\_102\common\db\scripts\derby (isinternal, iscoreaudit etc.)

How to get these scripts?

Execute the SAG-Installer for db configurator.

DB-Configurator :

DB-Configurator is used for creating internal tables for IS.

They are IS version dependent. So install 10.5 DB-Version on top of 10.5 IS

There could be mismatch of tables columns on different version of DB-Configurator.

Uses:

Once installed you can use

C:\SAG\DB\_configurator\_105\common\db\bin>dbConfigurator.bat -a create -d mysql –pr IS -v latest -l jdbc:mysql://localhost:3306/clusterScheduleTasks -u root -p changeit --admin\_user root --admin\_password changeit

This throws error message:

error.message: java.lang.RuntimeException: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver

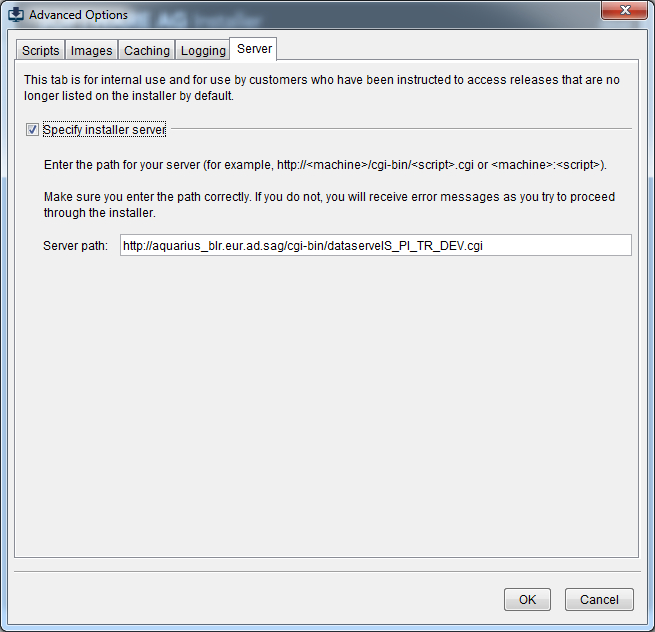
error.stackTrace: [Ljava.lang.StackTraceElement;@76a3e297

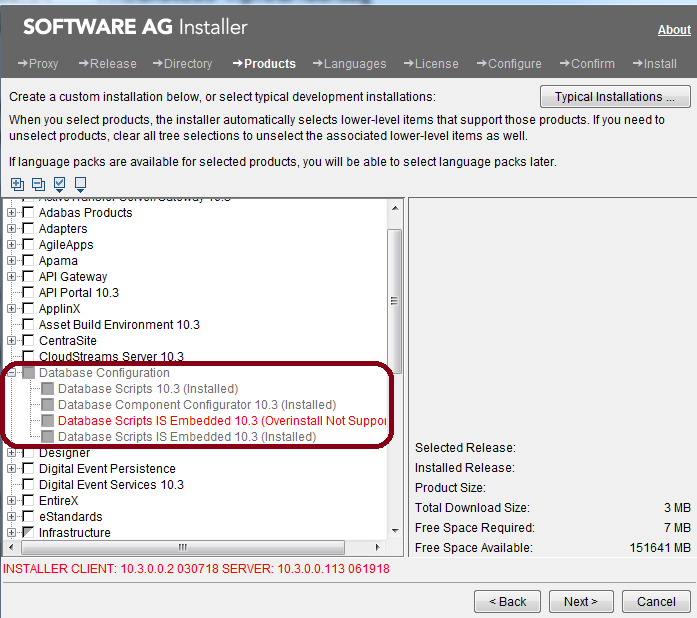
report 0: java.lang.RuntimeException: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver

To make this work Do following:

Copy jar file mysql-connector-java.jar into C:\SAG\DB\_configurator\_105\common\lib\ext

Put this file into - IS\_INSTALLED}/IntegrationServer/lib/jars/custom





You can create additional JDBC connection pools to write different types of data using different pools.

Functional aliases are predefined for the Integration Server during installation. You cannot add

or delete functional aliases, but you can change the JDBC connection pool to which a functional alias is assigned.

You direct each function to write to its database components by pointing the function at the appropriate connection pool.

Operations supported:

* Assign a pool alias to a functional alias
* Restart Function > Restart
* Edit and Test functional Alias

# User Management

Security > User Management

The definition for a user contains the user name, password, and group membership.

The definition for a group contains the group name and a list of users in the group.

After a client is authenticated (whether through basic authentication or client certificates), the server uses the group membership to determine if a client is authorized for the requested action, such as, using the Integration Server Administrator or invoking a service.

By setting up users and groups, you can control who can:

* Configure and manage the server. Only users that are members of the Administrators group (administrator privilege) can access the Integration Server Administrator.
* Create, modify, and delete services that reside on the server. Only users that are members of the Developers group (developer privileges) can connect to the server from Software AG Designer.
* Access services and files that reside on the server. Access to services and files is protected at the group level.

Access to the server's resources is controlled at the group level:

* Users of Administrators group can configure and manage the server using the Integration Server Administrator.
* Users of Developers group can connect to the server from Software AG Designer to create, modify, and delete services.

The server protects access to services and files using Access Control Lists (ACLs).

You set up ACLs that identify groups that are allowed or not allowed to access a resource.

**Predefined User Accounts**

Administrator - Everybody, Administrators, Replicators

to access the Integration Server Administrator to configure and manage the server.

Default Everybody, Anonymous

The server uses Default user when the client does not supply a userid.

Developer Everybody Developers

A user that can connect to the server from Software AG Designer to create, modify, and delete services that reside on the server.

Replicator Everybody Replicators

The user account that the server uses during package replication.

Adding User Accounts:

Security > User Management > Click Add and Remove Users.

Removing User Accounts:

Security > User Management > Click Add and Remove Users.

* When you delete a user, IS automatically removes the user from the members lists of all the groups to which it was assigned.
* These user accounts cannot be deleted: Administrator, Default, Developer, and Replicator.

Adding a User

Administrator : To grant administrator privileges to a user,

* you must assign that user to the Administrators group or
* to a group you have added to the Allow list of the Administrators ACL.
* In addition, you must make sure the user is not a member of a group that is denied access by the Administrators ACL.

Developer

Changing Passwords

Do not change a password if you are outside of the corporate firewall and you did not use SSL to connect to the Integration Server.

You cannot use the IS Administrator or Software AG Designer to administer users or groups stored in an external directory.

This restriction includes changing the passwords of these users.

Setting Password Requirements

You can use the watt.server.password.mode configuration property to specify whether or not

to enforce the password restrictions when setting passwords for administrator or non-administrator users.

When this property is set to strict, the password restrictions are enforced and when set to lax, the password restrictions are not enforced.

Disabling a User

you might disable the user account of a developer who is on vacation.

Because the user has been disabled rather than deleted, you can later reinstate the account without changing the password or resetting permissions Enabling a User.

For deployment, you should disable the Administrator user to prevent someone from

trying to guess the password and gain access to your system. Before disabling the

Administrator user, you must first create another user.

**Locking/Unlocking User Accounts** :

Security > User Management > Account Locking Settings > Edit

For security purposes, it is important to lock a user account when the user fails to provide the correct password after a specified number of failed login attempts to Integration Server.

A locked user account remains locked for a specific period of time, after which the account gets unlocked.

Integration Server allows administrators to configure the account locking settings for administrator and non-administrator users.

**Defining Groups**

A group is a named collection of users that share privileges. The privileges can be:

* Administrator privileges
* Replicator privileges
* Developer privileges
* Privileges to invoke a service
* Privileges to allow the server to serve files

Privileges to invoke a service or access files are granted and denied by Access Control Lists (ACLs) that you set up.

When an administrator creates ACLs, he or she identifies groups that are allowed to access services and files and groups that are denied access to services and files.

You can create new groups and add them to the allow lists of the Administrators, Replicators, or Developers ACLs.

When you create a group definition, you specify a group name and the members of the group.

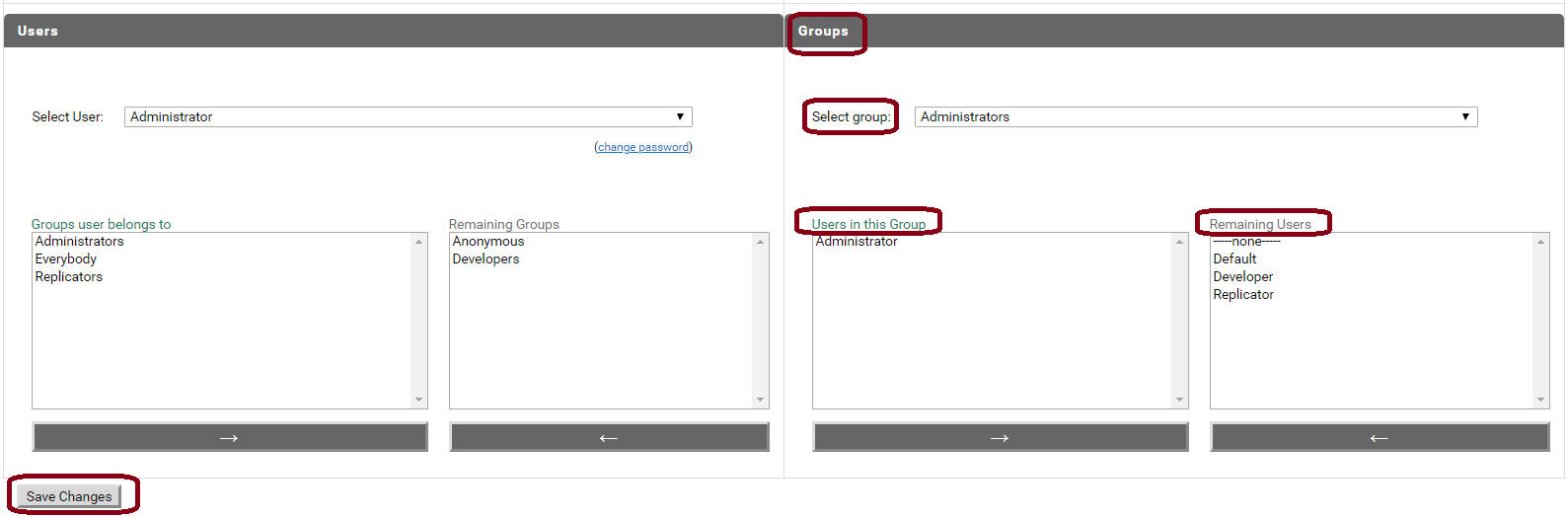
Predefined Groups

* Administrators
* Anonymous
* Developers
* Everybody
* Replicators

Group Operations:

* Adding Groups : IS Admin -> Security -> User Management -> Click Add and Remove Groups.
* Adding Users to a Group : IS Admi -> Security -> User Management -> Groups area
* Removing Users from a Group : IS Admin -> Security -> User Management -> Groups area
* Viewing Group Membership : IS Admin -> Security -> User Management -> Groups area
* Removing Groups : IS Admin -> Security -> User Management -> Click Add and Remove Groups.
  + You cannot delete any of these groups: Administrators, Developers, Replicators, Anonymous, and Everybody.

IS Admin -> Security -> User Management -> Groups area



# ACL

Creating ACLs : Security > Access Control Lists > Add and Remove ACLs

Specify one ACL name per line. At a time more than 1 ACLs can be created

Click Create ACLs

Deleting ACLs: You can delete ACLs that are currently

assigned to packages, folder, or other elements. When a client aempts to access an

element that is assigned to a deleted ACL, the server denies access.

Security > Access Control Lists > Add and Remove ACLs

select the ACL or ACLs you want to remove.

Click Remove ACLs. when prompted Click OK to delete the ACL.

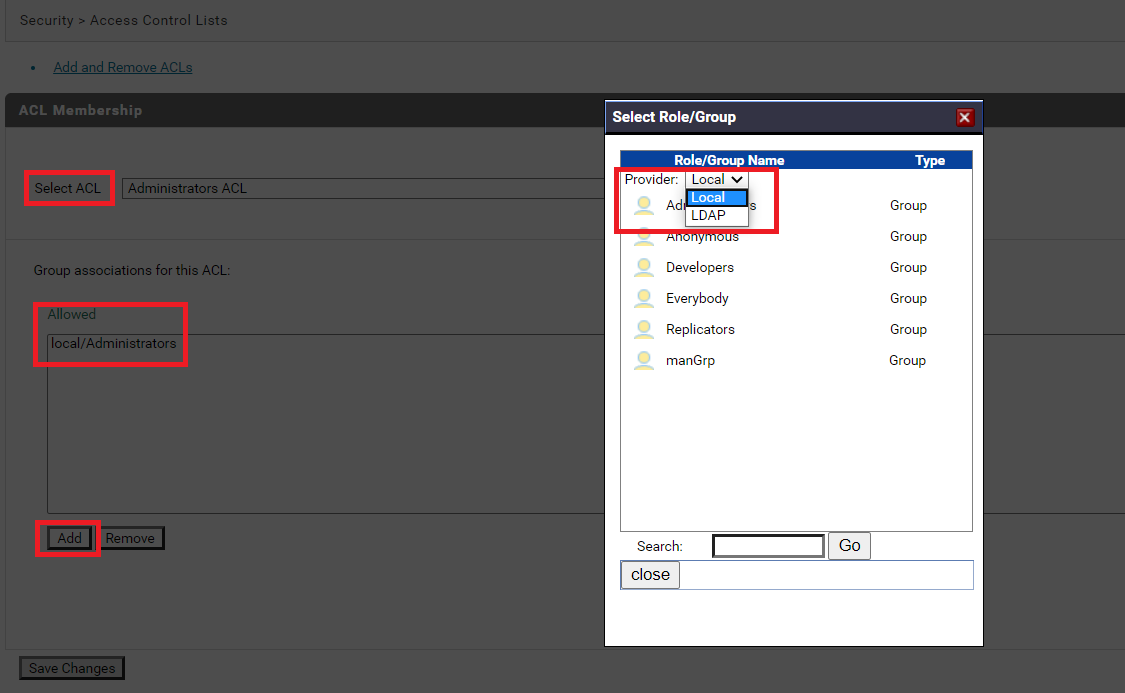
Allowing or Denying Group Access to ACLs

Edit a new ACL

to allow certain groups to access this ACL and

deny permissions to other groups.

Same can be done roles defined externally in a central user directory or in LDAP.



in the Provider list, select the location from which you want to select a user group.

If an external user directory is not configured, the Provider list does not appear.

# Securing Communications with the Server

SSL in IS

An administrator can configure the IS to use SSL to provide secure communications with the server.

Scope:

* How SSL works with Integration Server
* How to configure SSL authentication for the server side.

The request for an SSL connection originates from a client. The client can be one of the following:

* A partner application
* An Internet resource
* A web browser
* An Integration Server

During the SSL handshake process, the entity acting as the SSL server responds (to the request for a connection) by presenting its SSL credentials (an [X.509](#X_509_Cert) certificate) to the requesting client.

Those credentials are authenticated by the client, and one of below follows:

* One-way SSL connection : An SSL connection is established and information can be exchanged. The client must authenticate the server's credentials (For ex, viewing a savings account, or buying items with a credit card) before initiating the transaction
* Two-way SSL connection : The next phase of the authentication process occurs, and the server requests the SSL credentials of the client and verifies those credentials.

Roadmap for Configuring SSL:

* Create Integration Server keys and certificates
  + Generate a public/private key pair.
  + Generate a certificate signing request (CSR) and send to the certificate authority (CA) for signing.
  + Receive validated certificate from the CA.
  + Import signed certificate into a keystore.
  + If certificates contain certificate extensions that you want Integration Server to validate, set the watt.security.cert.wmChainVerifier.enforceExtensionsChecks server configuration property to true.
* Create keystore and truststore for Integration Server: Keystores and truststores are files that function as repositories for storage of keys and certificates necessary for SSL authentication with added layers of security and ease of administration. You can create and manage Integration Server keystores at the command line using keytool, the Oracle Java certificate editor.
  + Create a keystore and import the signed certificate and private key.
  + Create a truststore and import the certificate of the signing CA.
  + Store the keystore and truststore in a secure IS certificates directory(???).
  + Create aliases for the keystore and truststore.
  + If you use Oracle keytool to create the keystore, you cannot import an existing private key. So use of OpenSSL or Portecle is preferable.
* Obtain certificates of partner application or resource - and - Create certificate mapping : Use the Integration Server Administrator to save:
  + Signed certificate of the partner application.
  + Signed certificate of the CA for the partner's SSL certificate.
* Add an HTTPS or FTPS port : If you want to allow only secure connections to the server:
  + Ensure that the primary port uses an HTTPS port.
  + Delete all other non-HTTPS ports.

**Keystore/Truststore File Formats**

The default, certificate file format for an IS keystore/Truststore is JKS (Java keystore), the proprietary keystore implementation provided by Oracle.

Other keystore types (PKCS12) can be made available by

* Loading additional security providers
* Setting the watt.security.keyStore.supportedTypes property.
* Integration Server supports HSM-based (Hardware Security Module) keystores for ports, but not for other components.

How Integration Server Uses a Keystore and Truststore

For an Integration Server component to be SSL authenticated, it must have a valid, authorized [X.509](#X_509_Cert) certificate installed in a keystore file and the private key and signing certificate for the certificate issuer (CA) installed in a truststore file.

**Configuring Server-Side SSL Settings**

Prerequisite:

* Create at least one keystore, in JKS or PKCS12 format,
  + Containing an IS key pair to use for SSL and its corresponding key alias.
* Create at least one truststore, in JKS format,
  + Containing the trusted root certificate of the signing CA (and certificate chain, if necessary).
* Create a keystore and a truststore alias (Security > Keystore > Create Keystore Alias).

Definition:

* SSL Key: determines the Integration Server's SSL identity.
  + specifies the Integration Server private and public key pair
* Signing Key: specifies the private key with which to sign outgoing data streams from IS.
* Decryption Key : specifies the private key to use for decrypting incoming documents,

To configure the server for SSL authentication

* In the Integration Server Administrator goto Security > Certificates > Edit
* Select a Keystore Alias and Key Alias for the SSL Key, Signing Key, and Decryption Key.
* Select a Truststore Alias for the Truststore.
* Click Save Changes.

**Controlling Server SSL Security Level by Port**

You can configure IS to present different server certificates with different ports.

One reason to do this is so that different ports can provide different SSL security levels.

You determine the security level of a certificate during the certificate signing process. i.e. You tell the certificate authority which class of certificate you need and it creates a certificate with those attributes.

**Specifying** [**Cipher Suites**](#CipherSuittes) **with SSL**

IS provides following server configuration parameters that you can use to specify the cipher suites with inbound and outbound SSL requests.

* watt.net.jsse.client.enabledCipherSuiteList Specifies the cipher suites used on JSSE sockets that are used while making outbound HTTPS or FTPS requests.
* watt.net.jsse.server.enabledCipherSuiteList Specifies the cipher suites used on Integration Server ports that use JSSE and handle inbound requests.
* watt.net.ssl.client.cipherSuiteList Specifies the [cipher suites](#CipherSuittes) for outbound SSL connections.
* watt.net.ssl.server.cipherSuiteList: Specifies the cipher suites for inbound SSL connections.

You can also use a file (cipher suites list is too long) as the value for any of the parameters.

* Specifying file : watt.net.jsse.server.enabledCipherSuiteList=file:c:\ssl\ciphers.txt
* In the file, specify each cipher suite on a different line.
* Integration Server loads the file and its list of supported cipher suites at start up.
* You cannot specify a combination of comma-separated list, default, or the absolute file-path for a single parameter.

Enabling SSL debugging on IS : There are two ways

**First way:**

Add "-Djavax.net.debug=ssl" to the customer\_wrapper.conf {IS\_InstallDir}\profiles\IS\_{instanceName}\configuration\customer\_wrapper.conf

This will capture the details of SSL event on the server.

**Second way:**

Set watt property "watt.config.systemProperties=javax.net.debug=ssl" in IS extended settings (i.e. Settings > Extended), and restart the IS. Without restart SSL debug will not be enabled.

If it still fails share me below files:

- ~\{IS\_InstallDir}\profiles\IS\_{instanceName}\logs\wrapper.log

Bookmarks

**DEFINITION** **X.509 certificate**

<https://searchsecurity.techtarget.com/definition/X509-certificate>

Many of the certificates that people refer to as Secure Sockets Layer (SSL) certificates are in fact X.509 certificates.

An X.509 certificate contains information about the identity to which a certificate is issued and the identity that issued it.

* Issuer distinguished name
* Subject (certificate is issued to) distinguished name
* Subject public key information
* Validity period of the certificate
* Algorithm information

**Cipher Suites**

<https://en.wikipedia.org/wiki/Cipher_suite>

A cipher suite is a set of algorithms that help secure a network connection that uses Transport Layer Security (TLS) or its now-deprecated predecessor Secure Socket Layer (SSL).

The set of algorithms that cipher suites (for ex TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA) usually contain include:

* A key exchange algorithm (RSA): to exchange a key between two devices.
  + key is used to encrypt and decrypt the messages being sent between two machines.
* A bulk encryption algorithm (3DES\_EDE\_CBC): algorithm used to encrypt the data being sent.
* A message authentication code - MAC0(SHA) algorithm: data integrity checks to ensure that the data sent does not change in transit.

TLS defines the protocol that this cipher suite is for; it will usually be TLS.

In addition, cipher suites can include signatures and an authentication algorithm to help authenticate the server and or client.

To use cipher suites, the client and the server must agree on the specific cipher suite that is going to be used in exchanging messages.

handshake process for TLS1.2 is different than TLS1.3

TLS 1.0–1.2 handshake

* Client starts the process by sending a clientHello message to the server that includes
  + the version of TLS being used and
  + A list of cipher suites in the order of the client’s preference.
* In response, the server sends a serverHello message that includes
  + the chosen cipher suite and
  + The session ID.
* Next the server sends a digital certificate to verify its identity to the client.
* Client then sends an encrypted message to the server that enables the client and the server to be able to compute which secret key will be used during exchanges.
* After successfully verifying the authentication of the server and, if needed, exchanging the secret key, the client sends a finished message to signal that it is done with the handshake process.
* After receiving this message, the server sends a finished message that confirms that the handshake is complete.

Now the client and the server are in agreement on which cipher suite to use to communicate with each other.

Cipher suites defined for TLS 1.2 cannot be used in TLS 1.3, and vice versa, unless otherwise stated in their definition.

Doubts:

Port remapping: 36

If your Integration Server runs on a UNIX system, using a port number below 1024 requires that the server run as "root." For security reasons this is discouraged.

Solution: run your Integration Server using an unprivileged user ID on a high number port (for example 1024 or above) and use the port remapping capabilities present in most firewalls to move requests to the higher numbered ports.

How OSGi bundle class loader is Shipped with webMethods Integration Server? : 41

Each package has its own class loader : 41

Need to check.

zotezo.com/partners

Keystore Vs Trustsore:

<http://www.java67.com/2012/12/difference-between-truststore-vs.html>

Certificate chaining:

<https://www.thesslstore.com/blog/root-certificates-intermediate/>

SAML:

<https://developers.onelogin.com/saml>