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 Integration Server startup.

These settings can be overriden 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.

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