

Automation of J2EE Product Certification Process

Iddalgave Sandeep, Annapurna P. Patil
Department of Computer Science and Engineering,
M. S. Ramaiah Institute of Technology
Bangalore – 540 006, Karnataka, India.

Abstract

Automation is a process of developing a tool or a product to perform on its own, in order to achieve all the functionalities required of a complex system, without human intervention. The current Certification Process of the J2EE product in IBM is consuming unnecessary time even for the valid products. In order to reduce this processing overhead, the process of automation was proposed by the technologist to save time so as to accomplish the task early.

The need for automating the tool of Certification arose due to unwanted utilization of man-hour and man-power. The existing system halts after the tool finishes the comparison of the listings (which contain details of the build, its environment and the list of files present within the build), until the administrator logs-in. Administrator checks each transaction ID individually for intimating the clients about the status of their products. The proposed system aims at sending the intimation to the clients as soon as the comparison is done successfully, without the intervention of administrator i.e., an automated notification will be sent after successful comparison of the listings. The tool will also provide an option to generate chart/graph whenever the managerial team needs it at just one click.

The work carried out here is aimed at automating the process of certifying the Java products or assets developed by IBM Software Product Groups. It also aims at generating graphs at real-time for the number of products certified till date. The certification process is for IBM JRE/SDK used by IBM Software Product Groups. It will be done by comparing the listings submitted by the IBM Software Product Groups and the reference listings present in the Certification server. JRE/SDK should adhere to the constraints as per the Oracle license. The IBM product or asset is certified after the Certification Team sends the “Java Compatibility Logo”.

Here Javapath, listpath, build listing, product details for automation and start-and-end dates for graph generation act as inputs in Product Certification Process. After submitting the request, a transaction ID is generated which keeps track of certification status. The process ends with sending a notification about successful submission of the request. The report is generated as bar-graph with x-axis showing the duration in months, quarters or years and y-axis showing the number of products/builds certified.

After the implementation and the testing process, it is noticed that the time-consumption and man-power required for Certification Process is reduced.

Keywords: JRE/SDK, Java Compatibility Logo, Transaction ID, Javapath, Listpath.

1. Introduction

1.1. Product Certification

Product Certification is a process of certifying the products developed based on builds used and verifying them according to the license agreement. It is an internal process to a particular organization, to check whether the development tools adhere to the license agreement with the organization, which is providing the build for developing the product.

1.2. Importance of Certification

As per the TLDA any IBM Product or IBM Asset distribution that incorporates this Java Technology must -

1. be bundled with IBM value add which can be an IBM Product, IBM Service or IBM Asset
2. be able to pass the applicable Java Compatibility tests before it can go General Available (GA).
3. display the appropriate Java SE compatibility logo and the Java attribution statement on the IBM product/service/asset packaging.

1.3. Why Automation?

The process of Product Certification is time consuming in the present scenario as each product has to be certified by the administrator after successful comparison of listings of customer and reference by the Certification tool. The Certification tool would need 7 working days to certify a product. For report generation, the Certification tool needs 7-8 hours, which has been done manually.

To overcome these shortcomings, the Certification tool is been automated to make the request turnaround faster and available at any time. Automating this tool helps the product development teams to have product GA very soon. Automation involves certifying the products by sending the Java logo if the listings are Java compliant without the intervention of the administrator, which will shorten the time required to just few minutes from the time of request submission. This tool will also provide an option to generate graph as and when required by the managerial teams.

1.4. Definitions

Automation is a process of developing a tool or a product to perform on its own, in order to achieve all the functionalities required of a complex system, without human intervention.

1.5. Problem Definition

In this work, it is proposed to automate the process of Product Certification, by automating the process of build (JRE/SDK) download, list generation, communicating with different machines, and so on. It also includes generation of an online chart for the count of products as well as JRE/SDKs certified in a particular duration.

2. Literature Survey

2.1. Product Certification Process

Certification process is used for certifying the IBM deliverables which are using IBM Java as per the agreement with Oracle. Under the terms of Java Technology License and Distribution Agreement (TLDA), with Oracle, IBM (including IBM subsidiaries) can use, modify and distribute Java Technology with IBM Deliverables[2]. Java Technology is any Java code that implements Java Specifications. An IBM Deliverable is any IBM Products, IBM Service and IBM Assets.

2.2. Issues Related to Certification

The purpose of certification is to check for Java Compliance of JRE/SDK used by IBM Deliverables, directly or indirectly. This is confirmed by running JCK compliance test suite as provided by Oracle. Here are few issues that need to be addressed before the product gets certified:

- Binary or library files should be present within JRE/SDKs.
- Properties file should not be missing.
- Copyrights and notices should be present. Otherwise, a warning will be given at the time of certification process.
- License files must be present.

Removal of certain files/directories will not compromise the Java Compliance and will not affect the Certification process, like removal of docs, demo, man and sample folders and removal of *.pdb files[6].

2.3. Pre-submission Check List

- Major releases of product on Java 142/Java50 are not allowed without an explicit approval, as both these have passed end-of-take-up-date.

- From submitting the listing License guidelines have to be followed strictly.

- Java dependency has to be raised from SWG clearing House, if the product is directly bundling Java.

- Listings of JRE/SDKs have to be generated on the respective machines/architecture. For example: Linux AMD64 SDK/JRE listing is generated on a Linux OS machine of AMD64 architecture hardware.

- Only single listing has to be submitted for a particular JRE/SDK, as the product in turn supports different flavors of an OS.

2.4. Steps for Listing Submission

The following steps help for submitting the listing for certification of Java Products/ IBM deliverables.

1. First the customer has to generate the listing using either the PERL script or Java class file and has to login into the Certification tool, to submit the request for Certification.
2. After logging in, customer has to provide the details of product in appropriate fields given. Software Groups (SWG) option will be “checked” by default, if the product doesn’t belong to any SWG brand, then it has to be unchecked.
3. Customer has to provide End of Service (EOS), General Availability (GA) of the product and the expected date of certification.
4. Customer has to specify whether the product is of Major Release or not, by checking the box for Major release.
5. Version of the JRE/SDK has to be provided. If version being used is Java142/Java50, then “Exception for Shipment” button should get enabled.
6. After filling all the details, listings of the JRE/SDK’s used are to be uploaded and submitted.
7. Next, the page will display the details of each build. After confirming the details, click on submit button.
8. The last page will display the transaction ID for the request.
9. When the transaction ID is generated, certification process is triggered.
10. As the customer listing is available, Certification tool checks for its respective reference listing in the repository.
11. If the reference listing is available, then both the listings are compared for ‘Java Compliance’. If everything is correct as per the License agreement with Oracle, then an automated intimation is sent to the customer with “Java Logo”.
12. If reference listing is not available for a particular build, then the build is downloaded from the main repository on the respective reference machine, un-

jarred and the reference listing is generated and the repository is updated by the administrator.

After submission of the listings by the clients, the process needs a minimum of 7 working days to certify the product[3].

At the end of each month, quarter and year, the certification team has to prepare the graph of the certified products count and submit it to the higher managerial teams, which used to take a minimum of 4-5 man hours.

2.5. Need for Enhancement

- Checking each and every request manually for Java compatibility is a very time consuming process and requires more man-power like administrator checking, for Clearing House Dependency manually, for each and every request.
- Number of families in IBM Software Product Group was restricted to only 5 families. So, there is need for inclusion of other families of IBM.
- While submitting certification request, there is a need to separate the product details and dependency details into 2 different pages.
- Whenever the managerial team needs to view the status of certified products, an option should be provided to view the graph on monthly, quarterly and yearly basis.

3. System Requirement Specification

3.1. Introduction

The SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the administrator understand the other's requirements from their perspective at any point of time.

3.2. Purpose

The SRS document states in precise and explicit language the functions and capabilities a system must provide, as well as states any required constraints by which the system abide. This document provides a description of the introductory requirements, objectives, and characteristics of Automation of J2EE Product Certification Process. The expected audience of this document includes the system designers, testers and the system software developers. It will serve as an official reference to start the rest of the development phases.

3.3. Definition, Acronyms and Abbreviation

- 1) *General Available (GA)*: Release of the product, which is been requested for Certification, to the market by the client is termed as General Available.
- 2) *Build*: Either JRE or SDK is referred to as build.
- 3) *Listing*: Listings are the .txt files which are generated after Perl script is executed with its valid inputs. These files contain the details of the build, its environment and the list of files present within the build.
- 4) *Clearing House Dependency (CH Dependency)*: Each version of JRE/SDK will meet the end-of-take-up-date. An approval has to be taken for using such version of JRE/SDK, for developing product, from the higher authority. Such an approval is called as Clearing House Dependency.
- 5) *CH Dependency ID*: ID generated for requesting the approval is called as CH Dependency ID.
- 6) *End-of-take-up-date*: Maintenance of older version Java packages is frozen after certain period of time, which is referred to as End-of-take-up-date. Currently in IBM, Java3 and Java142 have met end-of-take-up-date.
- 7) *Product Bundling*: Sometimes a product is bundled within another product of the same family. This process is known as Product Bundling.
- 8) *Product Families*: The companies acquired by IBM or the different departments of IBM are called as Product Families. Few product families existing in IBM are Lotus, WebSphere, etc.

3.4. Specific Requirements

List Generation

- List to be generated for certifying the build has to be generated on the same system on which the product was developed by the IBM Software Product Group.
- The build used by the IBM Software Product Group must be downloaded from main repository only.
- The build should not be nearing end-of-take-up date (i.e., builds should be of Java 5 or higher version.).
- The build should have NOTICES and copyright files intact.
- Files in directory structures like *demo*, *sample*, *man*, and *docs* can be missing.

List Comparison

- Reference listing should be present in the database for the respective Customer listing submitted.
- Both the listing should be of same version, release and fixpack along with architecture, OS, etc.

- While comparing, *.pdb files and files in docs, demo, sample, and man folders have to be ignored.

Build Download

- The input to this module should be a String in the form of a URL.
- The URL should contain the build ID, OS, machine architecture, Java Version along with service, release and fixpack details.
- The build must be downloaded from main repository only.
- Different machines of different architecture should have a secure communication channel established using ssh.
- Reference machines should not prompt for username and password, each time the Certification tool tries to communicate with them.

Sending Mail

- All the listings of a particular transaction ID should be clarified from the concerned authority, if needed.
- All the listing of a particular transaction ID can be of any Java build versions.
- All the listings of a particular transaction ID should have status set to OK.
- None of the listings of a particular transaction ID should have been rejected.

Report Generation

- Database tables should be updated after each and every listing is updated.
- Tables should be updated in terms of month, quarter and year.
- Graph generation has to be requested for a particular duration, not from the starting of the Certification till date.
- Graph should be generated in JPEG or PNG format only.
- The report should be available to managerial teams as and when required.

3.5. Functional Requirements

Functional requirements describe the list of functional inputs and outputs to a program, the behavior, of the functions. As defined in requirements engineering, functional requirements specify particular results of a system.

Inputs to the program:

- Product details
- Build details
- CH dependency
- Build listing upload.

Output for the above points:

- Transaction ID generation
- Reference listing generation
- Sending “Java Logo” through automated mail.

3.6. Non-Functional Requirements

Security: The system is very secure as both client and server are configured during the time of installation. The product works only in intranet so will be inaccessible outside the network.

Scalability: This automated tool can be accessed through any platform and compatible on any machine. It is a machine independent architecture. This tool is connected to 23 different machines of different architectures.

Maintainability: Maintainability is one of the important phases of any software. Isolating defects, solving them, meeting new requirements, making the future maintenance easier and many more tasks sum up to maintainability.

4. System Design

4.1. Architectural Design

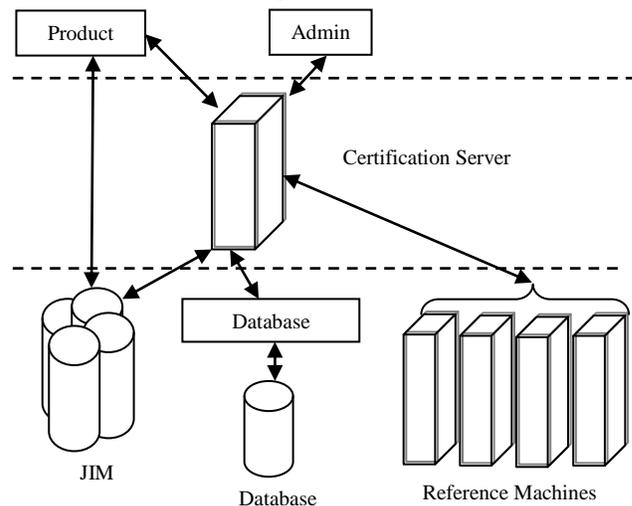


Figure 4.1: System Architecture of Certification Process

The below Figure 4.1 shows the overall System Architecture of the Certification Process, showing interaction of administrator and product team with various servers.

In the figure 4.1, Administrator is referred to the Certification team and Product team is the IBM Software Development Teams. Certification server is the main server which will manage the whole process of Certification. Database used in this work is MySQL, which is used for storing data of the Certification Process. JIM (Java Information Manager) is the main server of IBM, wherein all the Java builds are stored and maintained. In this project JIM acts as the main repository of Java builds. Reference machines are different machines of different platforms used for downloading builds in the automated

process of the Certification process. There 23 different machines have been identified for this propose within the JTC (Java Technology Center).

5. Results

The enhancements discussed previously have reduced the burden of certifying the product which satisfies the constraints as per TLDA. It has reduced the work from 7 person-days to just 15 minutes from the time of submission. Now the administrator will have lesser number of transaction IDs to be verified. Graph generation is now just a matter of button click. Time required to generate the graph has been reduced from 7 man-hours to just a click of button. It also reduced the workload on Certification team, who were involved in creating graph at the end of every month, quarter or year.

6. Conclusion and Future Work

Java Product certification process is a step at certifying the java based IBM products or assets. A certified product has to pass the java compliance test as per the agreement with Oracle. It speeds-up the number of products to certify and helps IBM Software development teams to make their product or asset for GA. Graphs are generated in real-time for managerial team on one click. The above mentioned enhancements have been implemented in the product and is been deployed for use in main repository.

In the implemented system, if there is a difference between the listings (which is supposed to be the unsuccessful comparison), manual intervention of the administrator is needed for process to continue. This manual intervention of administrator can be overcome in future by automating the system so as to respond by sending a mail to the client, which contains the list of differences. By implementing the above proposed enhancement, it can be seen that the delay in sending the notification by the administrator to the client is been reduced contributing towards better performance of the system.

References

- [1] Certification Team Profile: <http://w3.ibm.com/ibm/>
- [2] Java License Agreement: <http://www-07.ibm.com/in/isl/index.html>
- [3] Certification Help Page: <http://w3.hursley.ibm.com/java/tech/certify/>
- [4] Patrick Naughton, "The Java Handbook", Mc Graw Hill Osborne Media, March 22 1996
- [5] James Lee & Brent Ware, "Open Source Web Development with LAMP", Pearson Addison Wesley Professional, 2003
- [6] Vishal Prabhu, "CertificationGuide_draft3.odt"
- [7] Grady Booch, Ivar Jacobson, and James Rumbaugh, "The Unified Modeling Language User Guide", 2nd Edition, Addison-Wesley, 2005
- [8] Javascript: <http://www.roseindia.net/javascript/>
- [9] Roger.S.Pressman, "Software Engineering-A Practitioners Approach", 7th Edition, McGraw-Hill, 2007.
- [10] Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", 3rd Edition, Auerbach Publications, 2008.
- [11] Robert W. Sebesta, "Programming the World Wide Web", 4th Edition, Pearson Education.
- [12] Chris Bates, "Web Pogramming Building Internet Applications", 3rd edition, Wiley India, 2006.
- [13] Grady Booch et al, "Object Oriented Analysis and Design with Applications", 3rd Edition, Pearson, 2007.
- [14] Michael Blaha, James Rumbaugh, "Object Oriented Modelling and Desing with UML", 2nd edition, Pearson Education, 2005.
- [15] Jim Keogh, "J2EE : The Complete Reference", Tata McGraw-Hill, 2002.