Design of a Remote Software Distribution for Business Units

Ranjitha R

M,Tech, Information Technology, The National Institute of Engineering, Mysuru-09

---***---

Abstract - *Distributing, installing, updating and uninstalling software applications are common tasks in all enterprises. These tasks include a lot of processes and usually take up a lot of administrators' time and resources. Each of these tasks has to be completed in individual computers. The Software Deployment feature in Central location enables administrators to distribute, install, update and uninstall software applications remotely as well as automatically. Software repositories are used to store software packages.*

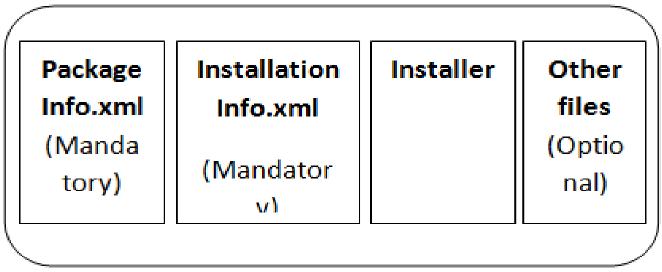
***Key Words*: Download, Installation, Enterprise, Package, Attributes, Business Units(BIU).**

1.INTRODUCTION

Software distribution is a feature provided to support seamless and automated distribution of various software updates to the devices that are connected to the enterprise. Software distribution can be considered to have the following sequence of steps 1)Publishing the software package(s) on the enterprise that are applicable to the modality systems connected to the enterprise. 2)Download of the relevant packages to the device. 3)Installation of the downloaded packages.

The software package is a compressed file that consists of the following:

PackageInfo.xml - This file contains attributes that describe the package. These attributes are now manually updated on the enterprise, going forward they will be part of the package. Ex. Name, description, download type, version, Authorization level, no of reboots, dependencies (if any), total installation time etc.


Package Info.xml (Manda tory)
Installation Info.xml (Mandator y)
Installer
Other files (Optio nal)

<Package Name>.ZIP
Fig-1 Package Structure

These attributes serve the following purposes

a. The attributes are downloaded to the device first. The attributes will be made available to the BU plug-in to check if the package is applicable to the device, so that only required packages gets downloaded to the device.

b. These attributes will be used to decide the type of download (manual or automatic)

c. These attributes provide details of the package to display to the user.

InstallationInfo.xml - This file contains metadata that provides details about the installation of the package.

Installation of a package can be Manual or automatic or Semi-automatic. BUs can use this file to configure how the package should be installed. If installation is manual, then instructions need to be provided in this file.

Installer – MSI, BAT, EXE etc. There can be one or more of these files.

Other files (Optional)- Any other files that BUs may deem necessary for an installation.

Remote Software Distribution is tool which enables user to download the latest software. The RSD tool is designed to have client-server architecture. The server is designed to be up and running all the time. This is to ensure it continuously monitor any package applicable for the device and performs automatic downloads/installations without any user interventions. The tool also provides a UI for the end user to perform various operations and also to get an insight of the history of upgrades performed on the device. RSD tool is a common tool with adapters. It exposes certain interfaces for the BU's to plug in BU specific functionalities to customize the tool behaviour.

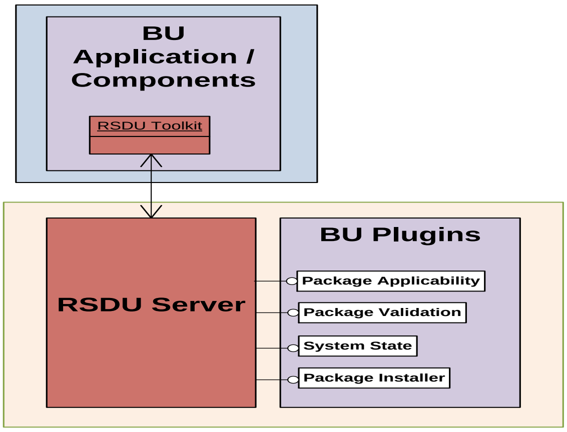


Fig-2: RSD Architecture Diagram

The above diagram depicts that the RSDU tool shall have client-server architecture. There shall be certain interfaces that the BIU will have to implement to achieve the complete functionality of the tool. There shall also be interfaces that may be consumed by the BIU components/applications. To make the BIU components/Applications agnostic of the server a toolkit shall be provided. The BIU plug-ins will be loaded in the RSDU Server process space. It has five components:

Package applicability, Package Validation, Package Installer, system state and toolkit.

2. SYSTEM DESIGN

2.1. Package Applicability:

Once the Package is available for download on the enterprise, RSDU Tool downloads the attributes of the packages applicable for the device and then passes the attributes to the BIU Plug-in to check the download eligibility criteria for the package.

This interface is designed for BIU plug-in to implement the various BIU-Specific checks if a package is eligible for download. RSDU Tool calls IsPackage Applicable method to decide whether to download the package or not. If the package is not applicable it shall be added to the rejected downloads list. Using the Toolkit a request may be made to RSDU tool to re-evaluate the rejected download list.
In case Package Applicability Exception is thrown by the plug-in then RSDU tool shall assume that the package is not applicable for the device.

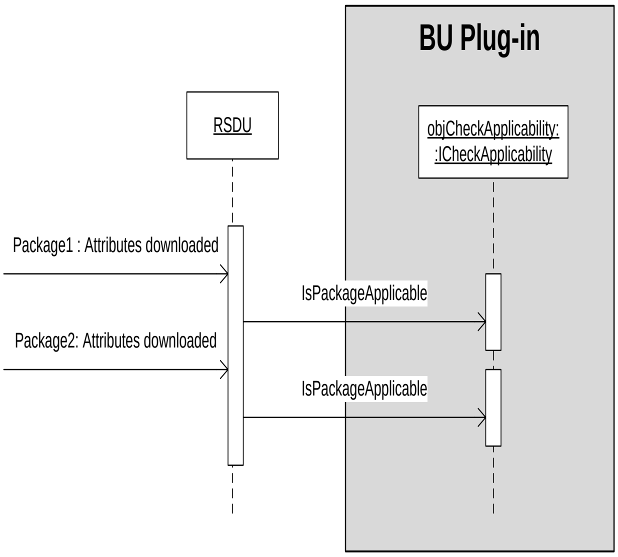


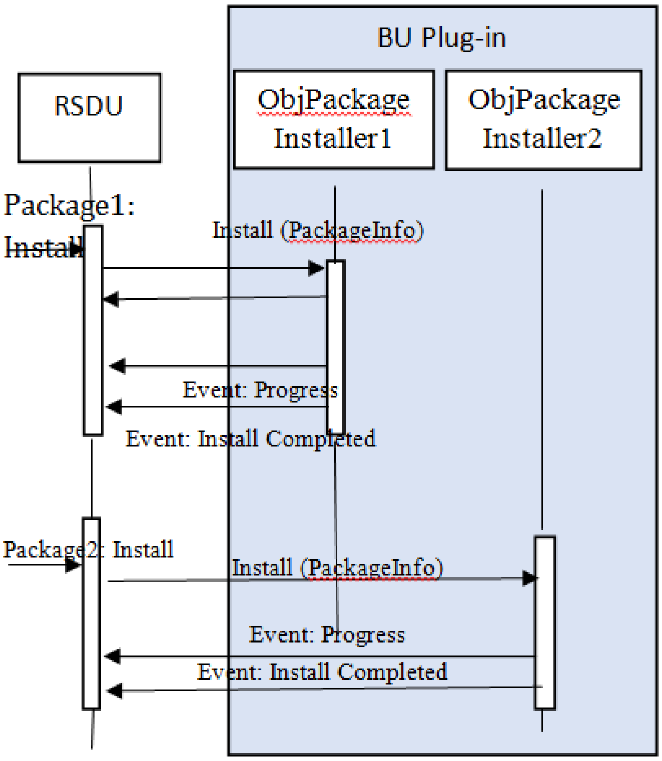
Fig-3: Package Applicability

2.2. Package Validation:

After the package is downloaded onto the device RSDU performs the checksum validation to ensure the package was downloaded without any errors. After successful validation the package shall be made available to the IPackage Validation plug-in for BIU-specific validation. When a package is uploaded on enterprise for a device on which FIPS is enabled, RSDU will skip the checksum validation for the package and package is made available to BIU specific Validation.

2.3. Package Installer

IPackage Installer shall be implemented by the BIU to handle the installation of a package. Any method call to this BIU plug-in is expected to return immediately. BIU plug-in may send progress updates by raising Installation Progress Event. BIU plug-in shall raise Installation Completed Event after completion of installation. On receiving the Installation Completed Event, Dispose() method call shall be made on the IPackage Installer Plugin.



If a Package Installer Exception is thrown then RSDU tool shall stop listening to Installation Progress Event and Installation Completed Event. RSDU tool shall assume that the Installation Failed.

2.4. System State

ISystem State shall be implemented by the BIU to provide current system state. System state basically means if the environment is feasible for the RSDU tool to carry out download/installation and the clinical user will not get affected by the tool's actions. For ex: only when the system is in idle mode RSDU tool is permitted to perform the download operation.

System state decides whether Download or installation can be performed at that point in time. Is Download Permitted method will be invoked before performing a download and Is Installation Permitted before performing an installation. BIU plug-in at any later point of time will be able to indicate the change in system state using call back event – System State Changed Event.

In case of System State Exception RSDU tool shall assume that installation/download is not permitted.

2.5. Toolkit

All interface points in the rsd are exposed through the toolkit. Toolkit has two interfaces:

1.INotification is a consumable interface through which BIU shall be able to get notifications from the RSDU. Notifications from RSDU tool such as Package Available For Download and Package Available For Install may have to be used to notify the user. Server Down can be used for notification of RSD server down. The idea is such notifications to the user shall be taken care by the BIU. In case of Lockout and UndoLockOut notificatons the BIU is expected to take actions accordingly.

2. ITrigger *Actions* is a consumable interface using which a BIU component may request the RSDU tool perform certain actions or request RSDU tool to get information such as list of pending installations and downloads. It is to be noted that pending installations/downloads in this context mean the installations/downloads for which the user has to take actions (accept the download/install or else perform manual task and update the result in UI). This list does not contain the list of installs/downloads which are pending due to non-permissible system state.

3.CONCLUSION

The present invention provides a method of remotely storing software executed locally, system, server and client.SSL VPN provides remote-access connectivity from almost any Internet-enabled location using web browser natively supports SSL encryption, the connection SSL VPN can help enterprise improve efficient production information protection security access, and decrease management maintenance cost. A service engineer is sent to the hospital to resolve the issue with the device. In this scenario, the Mean Time to Repair (MTTR) is more and also involves cost. By being able to access the console of the device remotely, the MTTR and the cost can be reduced.

ACKNOWLEDGEMENT

I would like to thank Dr. H.D. Phaneendra, Professor & Head, Dept. of CS&E at NIE, Mysore for his guidance and support in doing case study of the above paper.

REFERENCES

[1]. Rafael Prikladnicki, Roberto Evaristo, Jorge Luis Nicolas Audy, "Global software development in practice lessons learned" Wiley Online library, Volume8,Issue4, October/December 2003.

[2]. "Method and system for remote client installation", United States Patent.

[3]. Han Rong-zhen, "SSL VPN Technology Used Enterprise Remote Office System", Computers & Security[IF: 2.85], 2009.

[4]. Yin Shu-ling, "Research Application SSL VPN Technology", Computer Technology and Development, 2013.

[5]. Ping Chen , Li Jun Shen , Tao Zhang, "Design Implementation HTTP Proxy Server SSL VPN", Applied Mechanics and Materials, 2013.

[6]. Ling Li Zhu , Qing Tan, "Research Remote Fault Diagnosis System Design Mine Enterprise Equipment Based VPN", Advanced Materials Research, 2014.