RFP 53 - MEG Deployment

Draft

22 Pages

Abstract

The OSGi Mobile Expert Group (MEG) aims to apply and to extend the OSGi specifications for the operational management of mobile device platforms. This document is part of that overall effort and addresses use cases and requirements for a software component deployment model.

0 Document Information

0.1 Table of Contents

All Page Within This Box

0.2 Status

This document proposes deployment-related requirements and uses cases for the OSGi MEG extensions. These use cases are part of a bigger system that aims to specify application model suitable for mobile devices and rich mobile management solution for an OSGi-based system. Discussion is requested. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgement

The OSGi Mobile Expert Group Deployment Workstream would like to acknowledge the valuable input of the following participants:

Sinisha Djukic, ProSyst

Pavlin Dobrev, ProSyst

Magdolna Gerendai, Nokia

Per Gustafson, Gatespace

Mark Hansen, Motorola

James Jennings, IBM

Peter Kriens, aQute

Gabor Paller, Nokia

Kevin Riff, Espial

Jordan Simeonov, ProSyst

0.4 Revision History

The last named individual in this history is currently responsible for this document.

All Page Within This Box

Revision	Date	Comments
Initial	Jan 6. 2004	Gabor Paller, Nokia, gabor.paller@nokia.com.
0.1	Jan. 19 2004	Gabor Paller, Nokia, gabor.paller@nokia.com Handling comments from Andre.Kruetzfeldt (Sun) Adding text to Introduction, Application Domain and Problem Description sections.
0.2	Jan 28 2004	Gabor Paller, Nokia, gabor.paller@nokia.com Adding two requirements from Nokia Reformatting the requirements section to the same format used in other work streams
0.3	Feb 02 2004	Mark Hansen, Motorola, mark.hansen@motorola.com Edited sections 2 and 3 to clarify Scope. Edited requirements to align terms with other MEG Workstreams and removed redundant requirements.
0.31	Feb 17 2004	Mark Hansen, Motorola, mark.hansen@motorola.com Gabor Paller, Nokia, gabor.paller@nokia.com Revised document based upon discussions during Feb. 12, 2004 MEG Deployment call.
0.32	Feb 24 2004	Mark Hansen, Motorola mark.hansen@motorola.com Revised document based upon discussions during Feb. 19, 2004 MEG Deployment call.
0.33	Mar. 03 2004	Mark Hansen, Motorola mark.hansen@motorola.com Revisions from Mar 01 2004 MEG Deployment call.
0.34	Mar. 03 2004	Sinisha Djukic, ProSyst, s_djukic@prosyst.bg Mark Hansen, Motorola mark.hansen@motorola.com Gabor Paller, Nokia, gabor.paller@nokia.com Addition of Unit of Execution definition. SIM-related use cases and requirements. Subscription-based deployment requirements.
0.40	Mar. 04 2004	Mark Hansen, Motorola mark.hansen@motorola.com Revisions from March 04, 2004 call. Final draft before Cologne meeting.
0.41	Mar. 31 2004	Mark Hansen, Motorola mark.hansen@motorola.com Minor revisions from Cologne meeting and March 18 call.

All Page Within This Box

Revision	Date	Comments
0.50	Apr. 19 2004	Magdolna Gerendai, Nokia magdolna.gerendai@nokia.com Mark Hansen, Motorola mark.hansen@motorola.com Revised requirement for Controlled Availability (REQ-DEP-01-09) and added use cases for the requirement.
0.9	23 Apr. 2004	BJ Hargrave, hargrave@us.ibm.com Formatted for external review.

1 Introduction

OSGi Mobile Expert Group (MEG) aims to apply and to extend the OSGi specifications for the operational management of mobile device platforms. This document is part of that overall effort and addresses use cases and requirements for a software component deployment model.

2 Application Domain

2.1 Description

Mobile devices are rapidly becoming more complex as their capabilities increase. One such capability is the ability to install new software components after the time of manufacture. For example, device users already are, or soon will be, accustomed to installing applications on their devices from remote servers wirelessly or via local connectivity. Operators and other providers also wish to perform software repair and upgrades remotely to subscriber devices. To enable these deployment capabilities, a powerful management framework is required. OSGi provides such a framework, and this document will address the use-cases and requirements for extending OSGi for software component deployment for mobile devices

A software component lifecycle can be divided into several phases. First, a Developer must create and package a software component. The Developer then delivers the packaged component to a means of distribution, called a component Store. On the Store, the packaged component may be modified further. For example, the Store Owner may put multiple components into a single package and tailor them for a particular configuration. Once the packaged component is available on the Store, an Adminstrator or User initiates a request to a management server to deliver the component to the device. The management server has access to the Store and distributes a packaged component to the mobile device, which subsequently installs it. During installation, the component package provides all the necessary environmental setup required, such as initial configuration and data store setup. The device may also need to resolve issues such as dependency of component on available services and libraries. Once successfully installed, the component's runtime is managed by a management framework. Eventually, the component reaches end-of-life, perhaps due to obsolescence or undesirability by the user, and is uninstalled from the device by the action of an Administrator or User. When uninstalled, the component is completely and automatically removed from the device.

On some Stores, the component may be held in an intermediate packaging format not suitable for being directly distributed to the mobile device. A component packaged in an intermediate format is said to be a "unit of staging". Before being delivered to the mobile device, the component must be repackaged into a suitable format for the device, termed a "unit of deployment" or "unit of delivery".

This document will cover the use cases and requirements for a subset of the overall component lifecycle described previously. In particular, it will address those areas that central to the device and its immediate environment, which includes: "unit of delivery" packaging, "unit of deployment" packaging, installation, updating,

and uninstallation of software components on mobile devices. Installation will include the necessary environment setup required as well as dependency resolution.

While an important part of component deployment, developer delivery of a packaged component to a Store and any intermediate “unit of staging” packaging format will not be covered. This area is too broad to properly cover in a reasonable timeframe in addition to the core deployment scenarios to the device. These aspects may be addressed in a future phase of MEG activity or through a liaison with external standards bodies. Run-time management is out of the scope of this document and will be addressed in the Application Model and Device Management workgroups.

2.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

Caller – An entity accessing the management system. The call may be a conventional local method call but it may be also an access of the management system by means of a remote protocol (like SyncML DM).

Configuration parameters – Parameters specific to software components that allow configuration of certain software component functionalities.

Data store – A collective term for all data storage areas that may be used by a software component, such as configuration data, user preferences, and other component-specific data files.

High-level data services – Network-based services that require configuration (such as access points, credentials, transport selection, etc.)

Library – A software component that can be accessed by services and other units of execution but does not have its own execution state. When called, it uses the caller’s execution context.

Local Administrator – A local (to the device) entity authorized to perform management operations on the mobile device.

Remote Administrator – A remote (to the device) entity authorized to perform management operations on the mobile device.

Script – Program code written in a scripting language that satisfies the install, uninstall and upgrade script requirements on functionality and security. The script is normally part of the application package.

Service – A software component that can be accessed by services and other units of execution and has its own execution state .

Software component – A unit of software functionality (including code and static data resources). Services, libraries, and units of execution are collectively called software components.

Store – A database that stores packaged components, ready for delivery to the device.

Unit of Delivery - A collection of OSGi entities used by the management system to deliver an application to the device, but not necessarily known to the OSGi framework.

All Page Within This Box

Unit of Deployment - An OSGi artifact associated with the minimal, separately deployable entity, which is currently associated with an OSGi bundle.

Unit of Execution – An OSGi artifact associated with its own execution context.

Unit of Staging – A collection of OSGi entities used by the management system to stage a software component on a server, making it available for device download but not necessarily known to the management system on the device.

URL – Uniform Resource Locator

User – The end user of the device. Depending on the device policy, an end user may also have access to management functionality over a local API.

2.3 Acronyms and Abbreviations

API – Application Programming Interface

DMS – Device Management Server

DRM – Digital Rights Management

GUI – Graphical User Interface

JVM – Java Virtual Machine

KPI – Key Performance Indicator

MEG – Mobile Expert Group

OM – Operational Management

OSGi – Open Services Gateway Interface

OTA – Over The Air

UDL – Unit of Delivery

UDP – Unit of Deployment

UE – Unit of Execution

URL – Uniform Resource Locator

3 Problem Description

This document provides a detailed description of part of the process flow described in section 2. The use cases and requirements described in this document will focus on the device and its immediate environment . The main areas of concern are:

- Packaging of software components in a format suitable for deployment to a device (“UDL and UDP”)
- Initiating deployment-related management operations from the server or from the device
- Interaction between the management server and the management system on the device within the framework of existing standards
- Interaction between the local management entities and the management system on the device
- Operation of the management system during the deployment-related management operations.

The relevant actions are installation, updating, uninstallation, and reinstallation. Interaction with the policy system is also an important concern.

Deployment scenarios can be triggered from many sources such as a User or Administrator and can happen over many types of transports such as a radio network or short-range connection.

4 Use Cases

4.1 Installation

Use Case	U1. Installation
Actor(s)	Administrator (Remote or Local), User
Preconditions	• Device is configured and operational
Description	The Administrator or User requests UDL or UDP installation. The dependencies of the UDL or UDP and the capabilities of the device are analysed and the list of software components needed for the UDL or UDP to be installed on this device is created.[1] If any software component in this list is missing on the device or is not the correct version, that software component is added to the list of software components to be installed on the device. The software components in this list are then installed or updated in dependency order (software components that other software components depend on are installed or updated first). If there is a previously installed version of the same software component, the process continues according to U3. If the software component package is not yet on the device, then the UDL or UDP containing it is downloaded to the device, is installed and the installation script is executed. If there is data associated with the component, it is loaded into the appropriate location. The UDL or UDP may be transferred to the device over a variety of transport methods, such as (but not limited to) a cellular data network, local wireless connection, local wired connection to a PC, or device-based storage accessory. Policy rules may declare that in certain scenarios (e.g. standalone installation from a PC or e-mail) the package must be signed. In this case the signature is verified and the installation continues only if the signature was valid. End user may also be prompted so that he/she can be informed about the signature data. Status response is returned to the caller.
Exceptions	• The User or Administrator can't access the UDL or UDP specified. • The installation script could not be executed • Signature was mandated for the installation of a package in the dependency chain and the signature was invalid or the signer unknown. • Installation of any of the UDLs or UDPs fails. The device state is rolled back to the state before the management operation was started, the outcome is that no UDLs or UDPs will be installed on the device (even though there may be some whose installation may have been successful).
Post conditions	• Software component is installed on the device • Status is returned to the caller

[1] The device capabilities may determine which software components are needed for this device. For example certain devices may require a specific driver component.

All Page Within This Box

4.2 Reinstallation

Use Case	U2. Reinstallation
Actor(s)	Local Administrator, Remote Administrator, User
Preconditions	• Device is configured and operational
Description	The Administrator or User requests a software component reinstallation. It is possible to select whether the reinstallation should also create new data store or reuse the data store from the previous installation. It is also possible to specify whether the whole dependency chain is reinstalled or only the software component in question. If the UDL or UDP containing the component is not yet on the device, or its integrity is not trusted, then the package is downloaded to the device. The previous installation is removed. If data store is to be recreated, the uninstall script of the component is executed. The UDL or UDP is then installed again and the install script is executed only if the data store is to be recreated. If the data store is to be recreated and there is component-specific data associated with the software component, the component-specific data is loaded into the data store. This process is repeated on all software components in the dependency chain if this option was selected. Status response is returned to the caller. The scenario has the same subcases as U1. See the note about signature checking at U1.
Exceptions	• The User or Administrator can't access the software component specified. • The installation script could not be executed
Post conditions	• Software component is reinstalled on the device • Data store is recreated if it was requested • All software components in the dependency chain are recreated if it was requested • Status is returned to the caller

4.3 Update

Use Case	U3. Update
Actor(s)	Local Administrator, Remote Administrator, User
Preconditions	• Device is configured and operational • The software component to be updated is installed on the device

All Page Within This Box

Description	The Administrator or User requests a software component update. Dependency analysis described at U1. is executed. If the software component package (UDP or UDL) is not yet on the device, it is downloaded to the device. The UDL or UDP may be transferred to the device over a variety of transport methods, such as (but not limited to) a cellular data network, local wireless connection, local wired connection to a PC, or device-based storage accessory. The update scripts are executed. If component - specific data associated with the software component exists, this data is also given to the upgrade script. It is the responsibility of the upgrade script to combine the old data in the data store and the component-specific data in the new format so that the result is suitable for the new version of the software component. The updated version of the software component is installed; then previously installed version of the software component is uninstalled. It is possible to require in the request that software components that are not needed anymore as result of the update operation are uninstalled (for example a certain software component was needed in the previous version but is not needed anymore in the current version). Uninstallation of the unneeded software components is described at U4. See the notes about signature checking at U1. Status response is returned to the caller.
Exceptions	• The User or Administrator can't access the software component specified. • Previous version of the software component doesn't exist on the device so it cannot be updated. • The update script could not be executed. • Update of any of the software components fails. The device state is rolled back to the state before the management operation was started, and the outcome is that no software component will be updated or uninstalled on the device (even though there may be some software components whose update operation may have been successful or there may be software components that were selected for uninstallation as result of the update operation).
Post conditions	• Updated version of the software component is installed on the device • Status is returned to the caller

4.4 Uninstallation

Use Case	U4. Uninstallation
Actor(s)	Local Administrator, Remote Administrator, User

All Page Within This Box

Preconditions	• Device is configured and operational • Software to be uninstalled was installed previously
Description	The Administrator or User requests uninstallation of a software component. This software component must be in installed state for the operation to be successful. It is possible to specify in the request whether only the targeted software component is uninstalled or the whole dependency chain. If the dependency chain is uninstalled, software components that were installed as result of installing this software component (see U1. and U3.) and are not shared with any other software component (no other software component depends on this component) will also be uninstalled and a list is created that contains each software component to be uninstalled. If only the targeted software component is uninstalled, the list contains only the targeted software component. When uninstalling a software component in this list, first the uninstall script associated with the software component is executed (if there is any) then the software component is made unavailable in the application environment and is permanently removed from the device. If any component-specific data exists, the data store for that data (database table(s), file(s)) is removed. If an uninstall script encounters non-fatal problems, such as an error in the script or a desired component to uninstall is missing, then the uninstall process should continue on a best-effort basis. Status response is returned.
Exceptions	• The User or Administrator has no right to uninstall this software component • The software component is not available on the device.
Post conditions	• The software component is permanently removed from the device • Status response is returned to the caller

Use Case	U4.1. Uninstallation by the administrator on multiple devices
Actor(s)	Remote Administrator
Preconditions	Devices are selected for mass operation.
Description	The Remote Administrator launches mass uninstallation operation on the management server. The server uninstalls the selected software component on the devices selected for the mass operation. Before uninstallation, the user may be prompted to accept the action via the user-interface.
Exceptions	
Post conditions	• The software component is removed permanently from the devices where the operation could be executed successfully. • Status information is available which are the devices where the uninstallation was not successful.

4.5 Integrity check

Use Case	U5. Software integrity check
Actor(s)	Local Administrator, Remote Administrator, User
Preconditions	• Device is configured and operational. • The software component to be checked is installed and the User or Administrator has access rights to it.

All Page Within This Box

Description	The Administrator or the User invokes the software component integrity check function on the device management system. This also ensures that the software component files are available on the file system and software component dependencies are satisfied. This can be done remotely over the management protocol and over some local API. The device management system checks the known dependencies of the software component (what version of other software components needs to be installed) in order for the software component to function correctly and returns the result to the caller.
Exceptions	The User and Administrator doesn't have access right to the software component.
Post conditions	The device management system returns "integrity OK" or "integrity not OK" response to the caller

4.6 Software component discovery

Use Case	U6. Software component browsing
Actor(s)	User
Preconditions	• Device is configured and operational.
Description	The User requests a list of available software components (from the service operator, internet/intranet repository, mass media, e-mail, local connection, etc.). A list of software components is received containing also the software component dependencies, pricing and other useful information. A check should be made against software components already installed on the device and the user should be notified appropriately.
Exceptions	• The User doesn't have access right to list or modify software components. • The software component list could not be retrieved.
Post conditions	List of available software components is displayed to the User.

4.7 Software component subscription

Use Case	U7. Software component subscription
Actor(s)	User
Preconditions	• Device is configured and operational.
Description	Depending on the type of software component repository and subscription models provided by the operator, the User might wish to subscribe or unsubscribe for certain software components. Subscribing for a single component, for example, means that a user is charged for the component only once, when the subscription is confirmed. Then the User may freely install and uninstall the component many times, without being charged for the installation each time. This proves useful, when the User wishes to temporarily free some storage space on the device or migrates to a new device.
Exceptions	The User doesn't have access right to list or modify subscriptions.
Post conditions	Subscriptions are modified and take effect immediately.

4.8 Software component unsubscription

Use Case	U8. Software component unsubscription

All Page Within This Box

Actor(s)	User
Preconditions	• Device is configured and operational. • The user has one or more active subscriptions.
Description	The User doesn't wish to use the software provided by the operator anymore and unsubscribes it. If currently installed, any related software components are being uninstalled and the operator cancels the User's subscription. See U7 also.
Exceptions	The User doesn't have access right to list or modify subscriptions.
Post conditions	Subscriptions are modified and take effect immediately. Unsubscribed software components are automatically uninstalled.

4.9 Demand-Loading

Use Case	U9. Demand Loading
Actor(s)	Local Administrator, Remote Administrator, User
Preconditions	• Administrator or User has started the installation of a UDP/UDL, and the UDP/UDL refers to a missing component, a library for example. • The management agent is configured to demand-load the missing library.
Description	During the installation of the UDP/UDL, the management agent discovers that the UDP/UDL refers to a missing library. The management agent determines that it is configured to fetch the missing library autonomously. The management agent uses a simple mechanism to determine the download location and fetches the missing library. The missing library is downloaded as a UDP, installed, and then the installation of the original UDP/UDL continues.
Exceptions	• Demand-loading for this Java package name or download location is disabled. • The device is not able to discover the download location. • The UDP containing the library cannot be downloaded (because of wireless network problem, origin server problem, etc.)
Post conditions	The missing library is installed, which allows the installation of the original UDP/UDL to continue.

4.10 Operator Characteristic

Use Case	U10. Operator Characteristic
Actor(s)	Local Administrator, Remote Administrator, User
Preconditions	Local/remote Administrator has installed an UDP/UDL that is tied to an operator characteristic, such as IMSI.
Description	User selects a different operator (e.g. changing a SIM card or activating a subscription at a different operator). The operator characteristic associated with the UDP/UDL changes. After a configurable timeout (e.g. 1 week) the management agent removes the UDP/UDL because it was delivered to the User as part of a subscription plan.

All Page Within This Box

Exceptions	User reactivates the previous subscription so he/she can continue using the UDP/UDL.
Post conditions	The UDP/UDL is removed from the device.

4.11 Controlled availability of a software component

Use Case	U11. Controlled availability of a software component
Actor(s)	Administrator (Remote or Local), User
Preconditions	• Device is configured and operational.
Description	A service provider wishes to deploy a software component and wants to control the point of time at which the downloaded component becomes accessible on a device. This control is desirable because the download may take a long period of time or a server side counterpart may not yet be available. The service provider can download the desired component to a device, but the downloaded component will not be accessible to other components in the framework nor the end-user. When desired, an Administrator can request that the downloaded software component become accessible to other software components and/or the end user. A status response is returned.
Exceptions	• The User or Administrator has no right to make accessible this software component
Post conditions	• The software component is now accessible to other software components and/or to the end user. • Status response is returned to the caller.

4.12 Controlled unavailability of a software component

Use Case	U12. Controlled unavailability of a software component
Actor(s)	Administrator (Remote or Local), User
Preconditions	• Device is configured and operational. • The software component is installed on the device.
Description	A software component needs to be made inaccessible to users and/or other installed components during a period of time but is not removed from the device. A possible reason is the server-side part of the service requires updating. After the end of this desired period of unavailability, an Administrator can request that the software component be made accessible again to the end users and/or other components quickly. A re-download of the software is not necessary Status response is returned to the caller.

All Page Within This Box

Exceptions	• The Administrator has no right to make this software component inaccessible. • The software component is already inaccessible.
Post conditions	• The software component is accessible again to other software components and/or to the end user. • Status response is returned to the caller.

5 Requirements

5.1 Basic operations on applications and libraries

REQ-DEP-01-01. It MUST be possible to install an UDP and an UDL on the device.

REQ-DEP-01-02. It MUST be possible to uninstall an UDP and an UDL from the device.

REQ-DEP-01-03. It MUST be possible to upgrade an existing UDP to a higher version number.

REQ-DEP-01-04 It MAY be possible to upgrade an existing UDL to a higher version number.

REQ-DEP-01-05. It SHOULD be possible to downgrade an existing UDP to a lower version number.

REQ-DEP-01-06 It MAY be possible to downgrade an existing UDL to a lower version number.

REQ-DEP-01-07. It MUST be possible to reinstall an existing UDP.

REQ-DEP-01-08 It MAY be possible to reinstall an existing UDL.

REQ-DEP-01-09. It MAY be possible to control the point of time when the downloaded component becomes accessible to other components.

REQ-DEP-01-10. It MUST be possible to control access to the deployment functionality by a flexible policy system. *For example it is possible to restrict, who can install what software over local API calls or what management server can update what software remotely.*

5.2 Off-line operations

REQ-DEP-02-01. It MUST be possible to install an UDP or an UDL from a PC without a management server being involved, such as from a CD-ROM drive.

REQ-DEP-02-02. It MUST be possible to install an UDP or an UDL locally, without the help of the management server. *For example it is possible to download a package from the Internet and install it without management server interaction*

REQ-DEP-02-03. It SHOULD be possible to identify all the parts of an installed software component (code, resource, database, etc.) using dependency information stored in the management framework. *For example, a selective backup utility may use this information to save only the relevant parts during the backup process.*

REQ-DEP-02-04. It MUST be possible to access management functionalities over a local API. *For example it is possible to write a manager that allows installation of software components, provided that the invoking user-application has installation rights.*

5.3 Server-assisted installation

REQ-DEP-03-01. It MUST be possible to install/update UDPs and UDLs over the network from a server.

REQ-DEP-03-02. It MAY be possible to install/update UDPs and UDLs on the user's request. The user selects the component to download, and the installation happens automatically

REQ-DEP-03-03. It MUST be possible for a server to initiate the installation/update of an UDP or UDL independently of the user's actions. *For example a server administrator can install new software on the device.*

REQ-DEP-03-04 If a server initiates the installation/update of an UDP or UDL independently of the user, device MAY require user confirmation before completing the installation.

REQ-DEP-03-05. It MUST be possible that a server can assemble the contents of an UDL or UDP package with parts (i.e. binaries, settings, resources) custom-selected for a specific device. For example, the relevant application parts may be selected based on the device capability, pre-installed software, etc.

REQ-DEP-03-06. Management model MUST be supported where the secure communication between the remote manager and the mobile device can be established.

REQ-DEP-03-07. During an UDP installation or update operation, the management system SHOULD support dependency resolution to initiate the acquiring and installation of a missing component.

REQ-DEP-03-08. It MAY be possible for software components to start management sessions if they detect a malfunction that may be corrected by management interaction. *For example, if an e-mail user-application detects that the e-mail server cannot be contacted, it may start management session with the management server to "repair" the e-mail server settings.*

REQ-DEP-03-09. It MUST be possible for the mobile device to be managed by more than one remote manager (or local manager, e.g. the user). Such managers should possibly have different rights, e.g. manage different applications or perform different operations.

REQ-DEP-03-10. SyncML DM as management protocol MUST be supported.

REQ-DEP-03-11. Java MIDlet OTA MUST be supported as download protocol for MIDlets.

REQ-DEP-03-12. OSGi-based provisioning MUST be supported for applications (non-MIDlets).

REQ-DEP-03-13. The management protocol MUST be supported over wide-area radio networks (like cellular networks).

REQ-DEP-03-14. The management protocol MAY support short-range protocols like WLAN, Bluetooth, etc.

REQ-DEP-03-15. The management protocol MAY support local cable connections.

5.4 Deployment/undeployment/upgrade customization

REQ-DEP-04-01. It MUST be possible to customize the installation process by scripts, add-ons, etc

REQ-DEP-04-02. It MUST be possible to assign custom installer, uninstaller, upgrader and downgrader logic to installation packages that can add component-specific intelligence to the installation process. *For example, the custom installer can modify settings so that the user doesn't have to bother with configuration. The custom installer/uninstaller/upgrader/downgrader logic can be implemented in any convenient language like Java.*

All Page Within This Box

REQ-DEP-04-03. A scripting language MAY be available that provides conditional installation and user interaction (prompting, selection) features.

REQ-DEP-04-04. It MUST be possible to restrict the access rights of the install script. The install script doesn't run on behalf of the installer and may have limited rights.

REQ-DEP-04-05. It SHOULD be possible to associate component-specific data to installed components. This data is loaded into the appropriate data store during installation process. *For example, it is possible to declare that 3 database tables need to be created during installation and they are initialized from initial data that the package contains.*

5.5 Packaging

REQ-DEP-05-01. It MUST be possible to package all data (code, settings, resources, other component-specific data) into one file that can be expanded into individual files on the device.

REQ-DEP-05-02. The UDL MAY contain multiple software components. *For example, it is possible to package multiple units of execution within a single UDL package, such as a GUI-oriented user application and a background service associated with it.*

REQ-DEP-05-03. It MUST be possible to digitally sign UDLs and UDPs to check their authenticity. *For example it is possible to add company X's signature to the package so that the user or the management system can be sure of the package's origin.*

REQ-DEP-05-04. It MUST be possible to support of digital content signing compatible with MIDP2.0. Specifically, it should be possible to assign roles to signers (e.g. operators, manufacturers).

5.6 Software Component Discovery and Subscription

REQ-DEP-06-01. It MUST be possible to list available software components from the specified component repository or management server by a device.

REQ-DEP-06-02. It MAY be possible to subscribe for a specified software component if the component repository supports a subscription model.

REQ-DEP-06-03. It MUST be possible to unsubscribe for a subscribed software component if the component repository supports a subscription model and the specified software component is provided by the selected component repository.

5.7 Operator and Device Characteristics

REQ-DEP-07-01. It MUST be possible to attach an operator-specific subscriber identifier (e.g. IMSI) to an UDL/UDP.

REQ-DEP-07-02. It SHOULD be possible to attach an operator identification code to an UDP/UDL.

REQ-DEP-07-03. It MAY be possible to attach a device identification code (e.g. IMEI) to an UDP/UDL.

REQ-DEP-07-04. The operator or device characteristic MUST be attached to the UDL/UDP in such as way that it cannot be removed or thwarted by a reasonable effort from a user.

REQ-DEP-07-05. If the operator or device characteristic of a UDP/UDL does not match the current aspects of the device (e.g. the user changed subscription plan), the management agent MUST guarantee that the UDP/UDL is removed after a reasonable time.

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

6.2 End of Document