RFP 58 - MEG Device Management

Draft

20 Pages

0 Document Information

0.1 Table of Contents

0.2 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].

`UDP – Unit of Deployment, a MEG-defined entity deployable on mobile devices`

`UE – Unit of Execution, a MEG-defined entity associated with the execution context.`

Software component – a UE, a UDP, or any of the standard OSGi components such as Service or native library

Application framework, application environment – an OSGi framework, a MEG-specific extension of the OSGi framework or a MIDP environment

0.3 Revision History

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

Revision	Date	Comments
V 0.1	Jan 07 2004	Gabor Paller, Nokia, gabor.paller@nokia.com
V 0.2	Jan 16 2004	Vadim Draluk, Motorola, vdraluk@motorola.com 1. Use cases re-formatted and edited 2. Requirements restructured and extended

All Page within this Box

All Page Within This Box

Revision	Date	Comments
V 0.3	Jan 20 2004	Vadim Draluk, Motorola, vdraluk@motorola.com 1. “Application Domain” and “Problem Description” chapters added
V 0.4	Jan 26 2004	Vadim Draluk, Motorola, vdraluk@motorola.com 1. Use case and requirement added for SIM-based configuration policies and parameters support 2. Use case and requirement added for device re-set to the factory state
V 0.5	Feb 14 2004	Vadim Draluk, Motorola, vdraluk@motorola.com Added DMT-specific requirements for meta-data model and API
V 06	Feb 16 2004	Vadim Draluk, Motorola, vdraluk@motorola.com 1. Added explicit definition of terms MUST, MAY and SHOULD 2. Explicitly separated OMA-standard meta-model features from extensions 3. Accommodated comments from Nokia (Gabor Paller), Sun (Andre Kruetzfeld) and Dimitriy Trifinov (ProSyst)
V 0.9	23 April 2004	BJ Hargrave, hargrave@us.ibm.com Formatted for external review.

1 Introduction

This document is part of overall effort of the *Mobile Expert Group* (MEG) under OSGi to come up with use cases and requirements for operational management of mobile device platforms based on CDC configuration of J2ME and OSGi. The effort is subdivided into several *Work Streams* (WS) as follows:

- Application model
- Device Management
- Policies

All Page within this Box

All Page Within This Box

- Deployment
- MIDP

It also encompasses two horizontal efforts, which manifest themselves in all vertical aspects of the platform:

- Security
- Non-functional requirements and features

This document is devoted to consideration of use cases and requirements pertaining to the Device Management (DM). It addresses the issues of uniform access to the DM info by both local applications and remote management servers.

2 Application Domain

Device Management (DM) is a multi-faceted application domain that generally includes:

- Device configuration parameters setting
- Device configuration parameters querying
- Application configuration parameters setting
- Application configuration parameters querying
- Device characteristics monitoring
- Software provisioning (addressed to a large degree by the Deployment WS)
- Software repair

DM also covers various modes of management information delivery, such as OTA, local via a PC (tethered or wireless) and through the device's own user interface. It addresses the content's deployment on servers and PCs, protocols used to bring it to the device, and actions performed on the device to process this information.

DM issues are addressed by several standard bodies, including, but not limited to, JCP, Open Mobile Alliance (OMA), 3GPP and 3GPP2.

DM work stream of OSGi is determined to re-use the existing standards to the extent possible. It's goal is to define a DM infrastructure on the device so that it can be consistently used by application developed by different vendors, while remaining fully interoperable with and supportive of de-jure and de-facto standard DM protocols.

3 Problem Description

Today's mobile devices are characterized by constantly growing wealth and complexity of configuration and management data and operations. This data is accessed, and operations performed:

- Locally
  - by managed application retrieving their configuration parameters
  - users setting parameters via GUI
  - users running operations from menus
  - users installing applications from PCs
  - local agents provisioning parameters from SIM
- Remotely
  - Carriers provisioning parameters
    - SyncML DM
    - OMA CP
    - Proprietary protocols
  - Carriers installing applications
  - Manufacturers repairing firmware

Software components supporting these modes of operations are usually developed independently, are not portable between devices and manufacturers, and have no consistent mechanisms of accessing, managing and securing their data.

DM WS of MEG has a goal of standardizing the device-side infrastructure of device management, covering:

- DM data access and manipulation
- Policies governing DM data access and operation execution
- Interoperability with existing DM standards and protocols
- DM support for MEG-defined application, deployment and policies models

4 Use Cases

4.1 Software inventory query

4.1.1 Description

The User or Administrator wants to query the inventory of the installed software on the device. Request is sent to the device management system and the device management system responds with the name, version and dependency information of software components installed on the device. Only those software components are returned which the caller has access to.

4.1.2 Exceptions

The User or Administrator has no access to any applications on this device

4.2 Managing software component or application framework security policy

4.2.1 Setting and updating software component or application framework security policy

4.2.1.1 Description

The Administrator or User would like to set or update policy rules for a given UDP or UE or the application framework. The request contains the new policy set for the subject. It is possible to set policy rules for a subject that is not yet installed (for example it is possible to set up the policy rule for an application group that will be installed later). The device management system sets the policy rule set on the subject and sends signal to the application framework to activate it. Status response is returned to the caller.

It is the current policy connected to the application framework that controls if it is allowed for an administrator or a user to set, update and retrieve the policy for the UD, UE or application framework.

4.2.1.2 Exceptions

- The User or Administrator doesn't have right to access the software component
- One or more policy rule in the request is invalid

4.2.2 Retrieve software component or application framework security policy

4.2.2.1 Description

The User or Administrator retrieves management and security policy for given UDP or UE or the application framework. The request carries selection criteria that identifies the policy rules that the caller wants to retrieve. The request also carries the identification of the software component or the application framework whose policy rules the caller wants to retrieve. The matching policy rules are selected and are returned to the caller.

4.2.2.2 Exceptions

The User or Administrator has no right to retrieve the software component or application framework policy

4.3 Configuration management

4.3.1 Setting and updating software component configuration

4.3.1.1 Description

The user or the administrator wants to set or update configuration parameters of a software component. Request is sent to the device management system with the new configuration parameter values. The configuration parameter values are changed and the software component receives notification about the parameter change. The device management system returns status code.

4.3.1.2 Exceptions

- The application or service whose configuration parameters are set or updated is not installed
- The User or Administrator doesn't have right to access the software component

4.3.2 Setting and updating application environment configuration

4.3.2.1 Description

The user or the administrator wants to set or update configuration parameters of the application environment like the JVM or the OSGi Framework. These configurable values may be parameters related to JVM memory management (like garbage collection parameters) or application environment resource management parameters like maximum number of network handles per application or the maximum amount of memory that certain application may require. Request is sent to the device management system with the new configuration parameter values. The configuration parameter values are changed and the application environment takes the updated parameters into use. The device management system returns status code. In come cases application environment has to be restarted for configuration parameter changes to take effect.

4.3.2.2 Exceptions

The User or Administrator doesn't have right to reconfigure the application environment component

4.3.3 Querying software component configuration

4.3.3.1 Description

The User or Administrator wants to query the configuration parameters of an installed software component. List of the configuration parameter names belonging to the software component can be queried first, this step is optional. Then a request is sent to the device management system with the names of the configuration parameters to be queried. The device management system responds with the values of the configuration parameters.

4.3.3.2 Exceptions

- The application or service whose configuration parameters are queried is not installed
- The parameter which is queried is not present in the configuration store.
- The User or Administrator doesn't have right to access the software component

4.3.4 Running a DM script during UDP install/uninstall

4.3.4.1 Description

A UDP is installed on the device, and it has a set of configuration parameters associated with it. The parameter settings are represented by a script in a standard DM language, such as OMA CP or SyncML DM. The script is delivered as part of the UDP, and is referenced in the UDP's descriptor, thus allowing the script to be executed by the install routines of the framework.

4.3.5 Configuration parameters from SIM

4.3.5.1 Description

Configuration parameters can be deployed on, and retrieved from, the SIM card. When the SIM card is replaced, DM information on the new one is activated immediately.

4.3.6 Policies describing SIM-based objects

4.3.6.1 Description

If a policy deployed on the device pertains to a SIM-based object, the policy is immediately disabled after the SIM card is removed from the device.

4.3.7 RE-setting the device to the factory state

4.3.7.1 Description

Due to conflict between various settings/policies, or due to some other problems, the device has become unstable, and has to be returned to the factory state. This may be done either from a user menu item (in case such a menu item can function properly in an envirement in need of a reset) or OTA by the operator.

4.3.7.2 Exceptions

The device corruption prevents the factory re-setting software from operating

4.4 Performance indicators

Applications and services may publish indicator variables (Key Performance Indicators, KPI) that management systems can query obtaining information about the behaviour of the application. There is a standardized interface available for software components for publishing KPIs and refreshing KPI values.

Note that although querying device system software indicator is related to the proposed KPI query mechanism, in Java management scenarios we focus entirely on querying application-specific indicators that Java software components make available for query through the management system.

4.4.1 Retrieve application and service KPIs

4.4.1.1 Description

The Administrator or the User wants to inspect the behaviour of the application or service using the KPIs that the application or service published. The device management system is requested to return the values of a certain KPI or group of KPIs. The device management system returns the value(s) of the requested KPI(s)

4.4.1.2 Exceptions

- The application or service whose KPIs are requested is not installed
- The User or Administrator doesn't have right to access the KPIs of the application or service
- The KPI requested was not published by the application

4.4.2 Periodical update of the performance indicators

4.4.2.1 Description

The Administrator or User wants to inspect some KPI regularly to follow the behaviour of the application or service continuously. The device management system is requested to send the KPI value regularly to the caller. The device management system starts sending the KPI values at regular interval.

4.4.2.2 Exceptions

- The application or service whose KPIs are requested is not installed
- The User or Administrator doesn't have right to access the KPIs of the application or service
- The KPI requested was not published by the application
- Policy doesn't allow to the requested regularity (requested update was too frequent, requested data too large, etc.)

4.5 Software component sends a notification

4.5.1.1 Description

An event occurs inside the software component that needs urgent attention from the person with administrative rights. The software component generates a notification toward its administrator. If the software component is owned by the Administrator of a device management server, the notification is sent over the network to the device management server. If the application was installed by the User, the notification appears on the user interface. In any case it is guaranteed that the notification is not lost, if the notification cannot be sent over the network to the management server or if the user interface is busy with another task so that the notification cannot be displayed, the device stores the notification persistently and retries later. If the application is owned by multiple entities (like Administrator and User), each owner receives the notification.

4.5.1.2 Exceptions

- The application has no right to send notification
- The receiver of the notification disabled the reception of notifications

4.6 Retrieve software component log entries

4.6.1.1 Description

The Administrator or the User wants to check what events happened to a certain software component. It is also possible to define event severity or time period. He/she accesses the event log entries related to the software component. The action can be performed over a remote management protocol or local API.

4.6.1.2 Exceptions

- The software component is not installed
- The Administrator or the User has no access to the entries of the software component in question.
- The software component doesn't have entries in the event log

4.7 Manage application, service and application framework execution state

4.7.1 Manage application, service and application framework execution state (lifecycle state)

4.7.1.1 Description

The User or Administrator wants to change the execution state of an application, service or the application framework (JVM, OSGi) itself. Policy check is made to ensure that the management actor can execute this operation. The device management system makes the requested execution state change and returns status of the operation to the caller. According to the policy and the request parameters, notification is displayed to the end user so that he/she knows what is happening.

4.7.1.2 Exceptions

- The application is not installed
- The Administrator or User doesn't have right to make the execution state change for the application, service or the application framework.

4.7.2 Retrieve application, service and application framework execution state (lifecycle state)

4.7.2.1 Description

The User or Administrator wants to examine the execution state of an application, service or the application framework (JVM, OSGi) itself. The device management system returns the lifecycle state of the application or the framework.

4.7.2.2 Exceptions

- The application doesn't have management actor associated with it.
- The management session fails
- The operation is not successful even after the corrective management session and retry.

4.8 OMA-related use cases

The use cases in this section have strong dependency on OMA DM and most possibly belong to OMA DM standardization tracks. There are two reasons to mention these use cases here.

All Page within this Box

All Page Within This Box

- To emphasize that these use cases are necessary to realize the rich Operational Management solution that OSGi MEG envisions.

- To emphasize that the OSGi MEG Operational Management solution provides more complex access to OMA DM-managed resources than remote access by OMA DM protocol. Application and local users can also have access to these resources and this access is also controlled by the policy system.

4.8.1 Update device system software configuration

4.8.1.1 Description

The Administrator, User or an Application wants to change the configuration related to the device's system software. (access points, gateway settings, etc.) There may be multiple setting sets, in this case the setting set on which the operation is executed is also specified. Request is sent with the new settings and optionally with the ID of the setting set and the device management system sets the new configuration parameters for the system software. Status information is returned to the caller.

4.8.1.2 Exceptions

- The User or Administrator doesn't have right to change/add system settings
- The system settings in the request are incorrect.

4.8.2 Set/update user configuration

4.8.2.1 Description

The User or Administrator decides to manage device users (add/remove users) or to manage configuration information owned by device users. Request is sent to the device management system, the device user account is added/removed or the user-owned configuration information is updated. Status information is returned to the caller

4.8.2.2 Exceptions

- The User or Administrator has no access to the device user account in the request
- The User or Administrator has no right to manage user accounts
- The user whose configuration is to be updated doesn't exist

4.8.3 Retrieve device system software configuration

4.8.3.1 Description

The Administrator, User or Application wants to query the configuration related to the device's system software (access points, gateway settings, etc.) Request is sent and the device management system responds with the system software settings. There may be multiple setting sets, in this case the setting set ID of the set to be retrieved is also specified.

4.8.3.2 Exceptions

The User or Administrator has no right to retrieve system software settings.

4.8.4 Retrieve device capabilities

4.8.4.1 Description

The User, Administrator, UE or UDP launches a device management operation that needs device capability retrieval operation. Device capability information is an ordered capability set that describes everything a communicating party should know about the device hardware and software (system and installed). For example device information may contain the size of the device screen, the version of the device operating system and information about the media types that the device is able to display (both system and installed codecs). The management system obtains the device capability information either directly (querying the device or possibly a network-based profile database if network-based profile database is supported) or indirectly by querying the device whether it is able to execute certain operation(s). In the former case the capability manager returns the device profile to the management system, in the second case the device returns information that is implicitly related to the device profile.

4.8.4.2 Exceptions

The Administrator doesn't have right to access the device profile (because for example the Administrator has no right to manage the device).

4.8.5 Set management and security policy for the device system software

4.8.5.1 Description

The User or Administrator sets management and security policy for the device system software. This policy includes what actor can manage the device (locally or remotely) and access rights for the system software settings and information. The policy is set and status information is returned.

4.8.5.2 Exceptions

The User or Administrator has no right to change the device system software policy

4.8.6 Retrieve management and security policy for the device system software

4.8.6.1 Description

The User or Administrator retrieves management and security policy for the device system software. The request carries a selection criteria what are the system software policy rules that the caller wants to retrieve. The matching policy rules are selected and are returned to the caller.

4.8.6.2 Exceptions

The User or Administrator has no right to retrieve the device system software policy

5 Requirements

Most of the use cases listed above apply to both the user of the device and the remote administrator who are expected to have both read and write access to DM data and operations. Implicitly applications on the device have access to this information as well, having to retrieve their own configuration parameters.

Such a multitude of actors and access patterns result in a set of requirements for a uniform access mechanism on the device which can be used by all actors.

We are using the terminology for MUST, MAY, MUST NOT, SHOULD, SHOULD NOT as described in RFC2119. Here is a quick recap of the RFC for completeness.

MUST	This word means that the item/definition is an absolute requirement of the specification.
MAY	This word means that the item/definition is optional, but, if addressed by an implementation, it has to behave strictly in accordance with this specification
MUST NOT SHALL NOT	These words mean that the item/definition is an absolute prohibition of the specification.
SHOULD	This word means that that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
SHOULD NOT	These words mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

5.1 High-level requirements

The requirements in this section are listed in terms of an abstract common meta-data model anticipated for device management for mobile devices

5.1.1 DM meta-data model

REQ-DM-01-01. A single meta-data model (MDM) accommodating DM-related data and operations MUST be used to facilitate uniform access to the DM resources

REQ-DM-01-02. The MDM SHOULD be based on a standard well known in the industry, such as SyncML DM's device management tree

REQ-DM-01-03. The MDM MUST support definition of the access mode to DM objects

REQ-DM-01-04. The MDM MUST support definition of value constraints of individual DM objects

REQ-DM-01-04. The MDM MUST support definition of value constraints of multiple DM objects, such as objects being mutually exclusive or mandatorily present at the same time.

REQ-DM-01-05. The MDM MUST support definition of policies governing access to DM objects

REQ-DM-01-06. The DM objects policies SHOULD follow the semantics of the industry standard, chosen in accordance with **REQ-DM-01-02**.

5.1.2 DM data model

REQ-DM-02-01. The configuration parameters belonging to software components MUST be settable

REQ-DM-02-02. The configuration parameters belonging to software components MUST be available for retrieval

REQ-DM-02-03. There MAY be an interface between the application and the management system where the application can publish Key Performance Indicators (KPIs).

REQ-DM-02-04. There MUST be an interface where applications can write events into an event log accessible for the management system so that the application can be monitored even when it is not running. For example it is possible to examine the log of a certain application to provide post-mortem analysis

REQ-DM-02-05. It MAY be possible to retrieve KPIs of an application on the management server operator's request. For example if the user complains that the application doesn't work correctly, helpdesk personnel can examine the KPIs and decide an action.

REQ-DM-02-06. It MUST be possible to retrieve log records, making their lookup available to both remote administrator and device user

REQ-DM-02-07. There MAY be an interface that allows inspecting applications. For example it is possible to examine the application's memory and execution state or internal variables.

REQ-DM-02-08. There MUST be a method for authorized actors to set the policy belonging to applications, services, libraries, framework or OS services.

REQ-DM-02-09. There MUST be a method for authorized actors to retrieve the policy belonging to applications, services, libraries, framework or OS services.

REQ-DM-02-10. There MUST be a default policy that the device is shipped with. For example the default policy may restrict that unsigned applications cannot access the whole file system

REQ-DM-02-11. The original default policy MUST be changeable to a customized default policy. For example the original default policy may allow end users to install applications but the enterprise customer customizes it in such a way that the end user cannot install new applications.

5.1.3 DM operations

REQ-DM-03-01. Applications MAY send notifications to their owners. For example an application can notify the management server that it is running out of persistent storage

REQ-DM-03-02. Changes to DM objects MUST result in and event generated by the system.

REQ-DM-03-03. The DM object change event MUST carry information on the identity of the affected node and operation that caused the change

REQ-DM-03-04. The DM object change event MUST NOT carry information on either old or new value of the object

REQ-DM-03-05. It MAY be possible to launch a periodic diagnosis update operation. For example it is possible to request that certain KPI(s) be monitored for a certain time and the values be written into the log. The management server then collects the data when the operation is ready and the administrator personnel makes an analysis

REQ-DM-03-06. It MUST be possible to control the system's execution state from the management system. For example it is possible to start, stop, pause and restart system components from the management system

REQ-DM-03-07. It MUST be possible to control services' execution state from the management system.

REQ-DM-03-08. It MAY be possible to identify the running instance of a UE and execute the management action on a certain instance. This requirement is relevant only if the application model allows execution of the same UE in multiple instances.

REQ-DM-03-09. It MUST be possible to control the VM's or framework's execution status from the management system

REQ-DM-03-10. It MAY be possible to manage the list of applications and services that are started when the application environment is started. It is possible to have a list of "autostart" bundles and this list can be managed.

REQ-DM-03-11. There MUST be a way for an authorized actor to grant rights to another authorized actor to manage policy rules that the granting actor has management rights on. For example Device Management Server1 (DMS1) has set policy rules on a certain resource. Now DMS1 would like to make sure that DMS2 is also able to manage these policy rules. DMS1 grants management rights to DMS2 to manage these policy rules.

REQ-DM-03-12. Retrieval of policies and configuration parameters from the SIM card MAY be supported

REQ-DM-03-13. Re-setting configuration parameters and/or policies to their factory-defined state MAY be supported

5.1.4 DM interfaces

REQ-DM-04-01. Interfaces providing access to DM objects MUST be organized as an OSGi service

REQ-DM-04-02. The interface MUST support retrieval of individual objects

REQ-DM-04-03. The interface MUST support retrieval of objects' collections

REQ-DM-04-04. The interface MUST support updating of DM objects

REQ-DM-04-05. The interface MUST support creation of new DM objects

REQ-DM-04-06. The interface MUST support deletion of DM objects

REQ-DM-04-07. The interface MAY support retrieval of meta-data describing DM objects

REQ-DM-04-08. The interface MUST support setting of the principal on whose behalf the DM operation is performed

REQ-DM-04-09. The interface MUST support execution of DM scripts

5.2 DMT-specific requirements

The requirements in this chapter are presented with the assumption that SyncML DM's device management tree (DMT) will be used as the basic meta-data model for device management in MEG, along with necessary extensions.

5.2.1 DM meta-data model

5.2.1.1 OMA-standard features

REQ-MT-01. Device Management Tree (DMT), as defined by SyncML DM standard, MUST serve as the common meta-data model for all types of DM data access

REQ- MT-01-01. Data types of *char* (string), *int*, *bin* and *boolean, or means to enforce such types,* MUST be supported

REQ- MT-01-02. Node operations *GET*, *ADD*, *REPLACE*, *DELETE* and *EXEC* MUST be supported

REQ-MT-01-03. Default values for leaf nodes MUST be supported

REQ-MT-01-04. Enforcement of the valid tree structure, as defined by the DDF meta-data, MUST be supported

REQ-MT-01-05. ACL-based node access control relying on identity of the operation's principal MUST be supported

REQ-MT-01-06. Attributes *Name*, *ACL, Size* and *Title* MUST be supported

REQ-MT-01-07. Attributes *Version* and *Timestamp* MUST be supported

REQ-MT-01-08. String values MUST be encoded in UTF8

5.2.1.2 DMT meta-data model extensions

REQ-MT-02. DMT model extensions SHOULD be supported to provide for data quality and integrity

REQ-MT-02-01. Maximum string value length allowed MUST be supported and enforced

REQ-MT-02-02. Maximum and minimum values allowed for *int* values MUST be supported and enforced

REQ-MT-02-03. List of valid values for *char* and *int* nodes value length allowed MUST be supported and enforced

REQ-MT-02-04. Regular expression constraint on *char* values MUST be supported and enforced

REQ-MT-02-05. Valid string values based on value on plural sibling nodes somewhere in the DMT MUST be supported and enforced

REQ-MT-02-06. Constraint proscribing deletion of a node that is referred to according to the requirement REQ-REQ-MT-02-05 SHOULD be supported and enforced

REQ-MT-02-07. Constraint prescribing modification of a referring of the requirement REQ-MT-02-05 when the referred node is deleted or updated SHOULD be supported and enforced

REQ-MT-02-08. Maximum name length allowed for an individual plural node MUST be supported and enforced

REQ-MT-02-09. List of valid values for plural node names MUST be supported and enforced

REQ-MT-02-10. Regular expression constraint on plural node names MAY be supported and enforced

REQ-MT-02-11. Recursive sub-tree structures MAY be supported and enforced, allowing storage in the DMT of such information as menu trees

REQ-MT-02-12. Recursive sub-tree structures depth limitation MAY be supported and enforced

REQ-MT-02-13. Automatic generation of nodes upon creation of their parent nodes MAY be supported whenever prescribed by their meta-data

5.2.2 DM interfaces

REQ-DM-API-01. Java DMT APIs MUST be available

REQ-DM-API-02. The API MUST support authorization of information access, based on identity of the principal on whose behalf the action is executed

REQ-DM-API-03. For the API a permission type MUST be introduced, so that methods directly manipulating the principal have controlled access

REQ-DM-API-04. The API MUST support identity based either upon the invoking server or upon the signer of the bundle from which the invoking code originated

REQ-DM-API-05. The API MUST feature a class representing access to the DMT

REQ-DM-API-06. The tree class MUST have explicit release and flush member functions, persisting the changes made to the DMT, and releasing all locks

REQ-DM-API-07. The API MUST be thread-safe

REQ-DM-API-08. The API MUST support shared lock mode, under which no write operations on the DMT are allowed

REQ-DM-API-09. The API MUST support auto-lock mode, with initial shared lock acquired per tree/sub-tree, with automatic escalation to an exclusive lock on a write operation

REQ-DM-API-10. The API MUST support exclusive lock mode

REQ-DM-API-11. The API MUST support manipulation of the DMT as a whole, as well as of its sub-trees

REQ-DM-API-12. The API MUST support retrieval of single leaf node values

REQ-DM-API-13. The API MUST support update of leaf node values

REQ-DM-API-14. The API MUST support adding new nodes

REQ-DM-API-15. The API MUST support deleting existing nodes

REQ-DM-API-16. The API MUST throw exceptions to indicate error situations

REQ-DM-API-17. API MUST support inquiring about existence/accessibility of a node without throwing an exception

REQ-DM-API-18. The API MAY support retrieval of maps of leaf nodes of a single parent node

REQ-DM-API-19. The API MAY support setting of maps of leaf nodes of a single parent node

REQ-DM-API-20. The API MAY support retrieval of names of child nodes

REQ-DM-API-21. The API MAY support deep cloning of DMT nodes

REQ-DM-API-22. Changes to the DMT made through the API MUST result in notification broadcast

REQ-DM-API-23. The API MUST support retrieval of the node's ACL

REQ-DM-API-24. The API MUST support setting of the node's ACL

REQ-DM-API-25. The API MUST support retrieval of the node's Title

REQ-DM-API-26. The API MAY support retrieval of the node's meta-data

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 Author's Address

6.3 Acronyms and Abbreviations

6.4 End of Document