



# **Software Component Management Object**

## **Candidate Version 1.0 – 10 Jul 2009**

---

**Open Mobile Alliance**  
OMA-TS-DM-SCOMO-V1\_0-20090710-C

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. Under the terms set forth above.

# Contents

<b>1. SCOPE</b> .....	<b>5</b>
<b>2. REFERENCES</b> .....	<b>6</b>
2.1 <b>NORMATIVE REFERENCES</b> .....	<b>6</b>
2.2 <b>INFORMATIVE REFERENCES</b> .....	<b>6</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>7</b>
3.1 <b>CONVENTIONS</b> .....	<b>7</b>
3.2 <b>DEFINITIONS</b> .....	<b>7</b>
3.3 <b>ABBREVIATIONS</b> .....	<b>7</b>
<b>4. INTRODUCTION</b> .....	<b>8</b>
<b>5. SOFTWARE COMPONENT MANAGEMENT FRAMEWORK</b> .....	<b>9</b>
5.1 <b>DEPLOYMENT COMPONENT</b> .....	<b>9</b>
5.1.1 <b>Deployment Component State Machine and Primitives</b> .....	<b>9</b>
5.1.2 <b>Deployment Component Delivery</b> .....	<b>10</b>
5.2 <b>DELIVERY PACKAGE</b> .....	<b>10</b>
5.2.1 <b>Delivery Package State Machine and Primitives</b> .....	<b>11</b>
<b>6. STANDARDIZED MANAGEMENT OBJECTS</b> .....	<b>14</b>
5.1 <b>Introduction to Management Objects (Informative)</b> .....	<b>14</b>
6.1.1 <b>Definition and description of Management Objects</b> .....	<b>14</b>
6.2 <b>DDF COMPLIANCE</b> .....	<b>14</b>
6.2.1 <b>Conformance Definitions</b> .....	<b>15</b>
<b>7. SOFTWARE COMPONENT MANAGEMENT OBJECT</b> .....	<b>16</b>
7.1 <b>TREE STRUCTURE</b> .....	<b>16</b>
7.2 <b>STATE INFORMATION IN THE TREE</b> .....	<b>16</b>
7.3 <b>FIGURE OF THE MANAGEMENT OBJECT (INFORMATIVE)</b> .....	<b>16</b>
7.4 <b>SOFTWARE COMPONENT MANAGEMENT OBJECT PARAMETERS</b> .....	<b>18</b>
<b>8. BEHAVIOR ASSOCIATED WITH THE MANAGEMENT OBJECT</b> .....	<b>30</b>
8.1 <b>STRUCTURE ELEMENTS OF &lt;ITEM&gt;/&lt;DATA&gt; FOR REPORTING</b> .....	<b>30</b>
8.1.1 <b>ResultCode</b> .....	<b>30</b>
8.1.2 <b>Identifier</b> .....	<b>31</b>
8.2 <b>EXEC COMMAND SEMANTICS ON DOWNLOAD SUB-TREE</b> .....	<b>31</b>
8.3 <b>EXEC COMMAND SEMANTICS ON DELIVERED</b> .....	<b>32</b>
8.4 <b>EXEC COMMAND SEMANTICS ON DEPLOYED</b> .....	<b>34</b>
8.5 <b>USE OF SYNCHRONOUS REPORTING</b> .....	<b>34</b>
8.6 <b>USE OF THE GENERIC ALERT</b> .....	<b>35</b>
8.7 <b>SCOMO RESULT CODE</b> .....	<b>36</b>
8.7.1 <b>Alert Types for the Software Component Management Object</b> .....	<b>37</b>
8.8 <b>USER INTERACTION COMMANDS</b> .....	<b>37</b>
8.8.1 <b>Requesting User Confirmation</b> .....	<b>37</b>
<b>9. SCOMO USAGE (INFORMATIVE)</b> .....	<b>39</b>
9.1 <b>DELIVERY OF DEPLOYMENT COMPONENTS</b> .....	<b>39</b>
9.1.1 <b>OMA DM Delivery</b> .....	<b>39</b>
9.1.2 <b>Alternate Download</b> .....	<b>39</b>
9.2 <b>INSTALLATION OF DEPLOYMENT COMPONENTS</b> .....	<b>39</b>
9.3 <b>ACTIVATION/DE-ACTIVATION OF DEPLOYMENT COMPONENTS</b> .....	<b>39</b>
9.4 <b>REMOVAL OF SOFTWARE COMPONENTS</b> .....	<b>39</b>
9.5 <b>RETRIEVING INVENTORY OF DEPLOYMENT COMPONENTS</b> .....	<b>39</b>
<b>10. SECURITY CONSIDERATIONS (INFORMATIVE)</b> .....	<b>40</b>
<b>APPENDIX A. CHANGE HISTORY (INFORMATIVE)</b> .....	<b>41</b>
A.1 <b>APPROVED VERSION HISTORY</b> .....	<b>41</b>

A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY .....41

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)..... 44

B.1 SCR FOR SCOMO TREE STRUCTURE .....44

B.2 SCR FOR SCOMO CLIENT .....44

B.3 SCR FOR SCOMO SERVER.....47

B.4 SCR FOR DM SYSTEM .....47

## Figures

Figure 1: Deployment Component State Machine.....9

Figure 2: Delivery Package State Machine..... 12

Figure 3: Software Component Management Object..... 17

## Tables

Table 1 – Deployment Component Primitives.....10

Table 2 – Delivery Package Primitives..... 13

Table 3 – Responsibilities associated with Exec on Download sub-tree.....32

Table -- Responsibilities associated with Exec on Delivered sub-tree .....33

Table -- SCOMO Result Codes .....37

# 1. Scope

This specification describes Management Object employed in a Software Component management process that leverages OMA DM protocol [DMPRO]. It provides a standard DM Management Object and associated client and server-side behaviour necessary to manage Device's Software Components.

Software Component Management Object enables delivery and management of Software Components within Device.

Dependency management between Software Components, format and structure of Delivery Package are out of the scope of this specification.

This specification is expected to meet all the requirements defined in [SCOMO-RD] and to comply with [SCOMO-AD].

## 2. References

### 2.1 Normative References

- [DMPRO] “OMA Device Management Protocol, Version 1.2”. Open Mobile Alliance™.  
OMA-TS-DM\_Protocol-V1\_2.  
URL: <http://www.openmobilealliance.org>
- [DMREPRO] “OMA Device Management Representation Protocol, Version 1.2”. Open Mobile Alliance™.  
OMA-TS-DM\_Protocol-V1\_2.  
URL: <http://www.openmobilealliance.org>
- [DMTND] “OMA Device Management Tree and Description, Version 1.2”. Open Mobile Alliance™.  
OMA-TS-DM-TND-V1\_2  
URL: <http://www.openmobilealliance.org>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997,  
URL: <http://www.ietf.org/rfc/rfc2119.txt>
- [RFC3986] T. Berners-Lee, R. Fielding and L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax”,  
IETF RFC 3986, August 1998.  
URL: <http://www.ietf.org/rfc/rfc3986.txt>
- [SCOMO-AD] “SCOMO Architecture”, Open Mobile Alliance™, OMA-AD-SCOMO-V1\_0,  
URL: <http://www.openmobilealliance.org/>
- [SCOMO-RD] “SCOMO Requirements”, Open Mobile Alliance™, OMA-RD-SCOMO-V1\_0,  
URL: <http://www.openmobilealliance.org/>
- [SCR RULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures,  
URL: <http://www.openmobilealliance.org/>

### 2.2 Informative References

- [DLOTA] “Generic Content Download Over The Air Specification Version 1.0”, Open Mobile Alliance™, OMA-  
Download-OTA-v1\_0,  
URL: <http://www.openmobilealliance.org/>
- [RFC2616] “Hypertext Transfer Protocol – HTTP/1.1”. Network Working group. June 1999.  
URL: <http://www.ietf.org/rfc/rfc2616.txt>
- [SSL 3.0] The SSL Protocol, Version 3.0, <draft-freier-ssl-version3-02.txt>, Transport Layer Security Working  
Group, Alan O. Freier et al, November 18, 1996.  
URL: <http://www.netscape.com/eng/ssl3/draft302.txt>
- [TLS 1.0] “The TLS Protocol Version 1.0”, T. Dierks, C. Allen, January 1999.  
URL: <http://www.ietf.org/rfc/rfc2246.txt>

## 3. Terminology and Conventions

### 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

### 3.2 Definitions

See the DM Tree and Description [DMTND] document for definitions of terms related to the management tree.

<b>Delivery Package</b>	An abstract collection of Deployment Components that are delivered as a single unit. Installing a Delivery Package can result in deployment of multiple Deployment Components.
<b>Deployment Component</b>	Deployment Component consists of a Software Component that is associated with metadata. A Deployment Component is dedicated to management purposes..
<b>Device Management System</b>	A background system capable to interact with a (set of) Device(s) for the purpose of Device Management.
<b>Management Object</b>	A MO is the data model for information which is a logical part of the interfaces exposed by DM components
<b>Primitive</b>	Operation that transfers Deployment Component to a different state.
<b>Software Component</b>	A Software Component is a resource utilized by the Device software platform

### 3.3 Abbreviations

<b>DM</b>	Device Management
<b>DMS</b>	Device Management System
<b>MO</b>	Management Object
<b>OMA</b>	Open Mobile Alliance
<b>SCOMO</b>	Software Component Management Object

## 4. Introduction

The Software Component Management Object defined in this specification enables remote Software Component management within a Device. This can be used for example to update antivirus software or browser plug-in etc.

Management operations defined by this specification for the purpose of Software Component management on the Device support delivery, download, installation, update, removal, activation & de-activation. In addition, retrieval of inventory of Software Components on the Device is also supported.



## 5. Software Component Management Framework

### 5.1 Deployment Component

Deployment Component is a Software Component that is dedicated for management purposes. Deployment Component is execution environment neutral so that it MAY be used to describe the Software Components of different platforms.

Deployment Component MAY include any type of software asset as Software Components, such as executables, applications, libraries, UI-elements, certificates, licenses etc.

Deployment Component consists of the Software Component and metadata. The metadata consists of multiple attributes (e.g. component id, name, version, etc.) associated with Software Components. The metadata MUST be provided by the Device Management System and the same values MUST be placed in corresponding nodes under Deployed sub-tree by the SCOMO Client after installation. The metadata MUST be delivered within a Delivery Package.

#### 5.1.1 Deployment Component State Machine and Primitives

Deployment Component management Primitives trigger transition from one state to another. The Device internal operations and state transitions MUST appear to be atomic. If a state transition fails the SCOMO Client MUST reverse the operation and make sure Deployment Components remain in the previous state.

Each Deployment Component MUST have one of the three states as follows:

- Inactive state (reflected in the State node in the Deployed branch)
- Active state (reflected in the State node in the Deployed branch)
- Removed state (which is a logical state reflected by deleting the <x> sub-tree representing the component from the Deployed branch).

The following figure depicts the state transition triggered by Deployment Component Primitives executed:

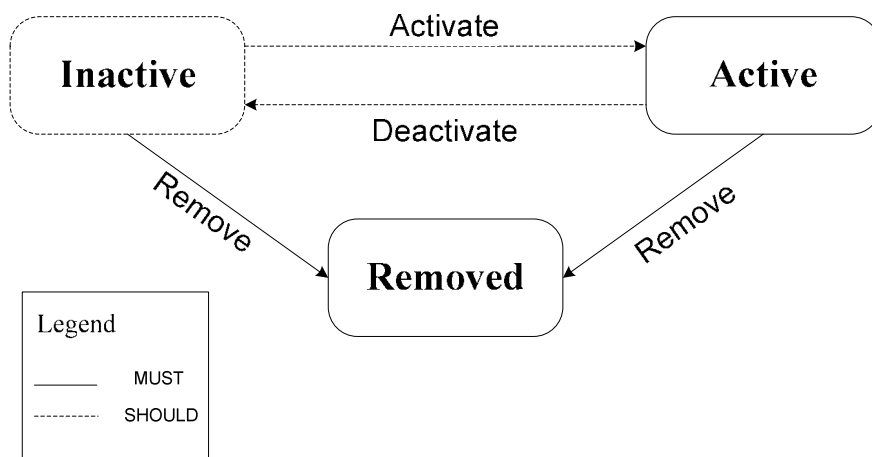


Figure 1: Deployment Component State Machine

##### 5.1.1.1 Inactive state

In Inactive state the Deployment Component has been installed into the execution environment but services or resources it embodies are not accessible to other entities or resources (including end-user).

The main goal of Inactive state is to minimise the downtime of Deployment Component management operations. Interference with external events (e.g. some end-user actions) could disturb or even block some management tasks. Inactive state is a powerful concept when implementing fault tolerant systems since it enables controlled management operations. E.g. safe removals of Deployment Components from runtime-environment.

The SCOMO Client SHOULD support this state.

### 5.1.1.2 Active State

In Active state the Deployment Component has been installed into the execution environment and services or resources it embodies are accessible to other entities or resources (including end-user).

A service that consists of multiple Deployment Components is ready for launch after all the relevant Deployment Components have reached the Active state.

The SCOMO Client MUST support this state.

### 5.1.1.3 Removed State

In Removed state the Deployment Component has been removed from the Device and services or resources it embodies are no longer accessible to other entities or resources (including end-user). The SCOMO Client MUST remove the Software Component from the Device and remove corresponding nodes of the Deployment Component from Deployed tree.

The SCOMO Client MUST support this state.

### 5.1.1.4 Deployment Component Primitives

The Exec command semantics related to Primitives are described in Chapter 7.1.

Deployment Component Primitives are used to perform management process and triggers state transition of a Deployment Component.

Primitive	Description	Applicable states	Post-Primitive state	Primitive support
Remove	This Primitive is used to remove a Deployment Component from the Device.	Active Inactive	Removed	MUST be supported
Activate	This Primitive is used to move a Deployment Component from the Inactive state to the Active state.	Inactive	Active	SHOULD be supported
Deactivate	This Primitive is used to move a Deployment Component from the Active state to the Inactive state.	Active	Inactive	SHOULD be supported

Table 1 – Deployment Component Primitives

## 5.1.2 Deployment Component Delivery

- Deployment Components MUST be delivered to the Device in a Delivery Package described in section 5.2.

## 5.2 Delivery Package

It could be desirable to deliver multiple Software Components in a single download operation. The Delivery Package abstraction represents platform specific package formats used to deliver Software Components to the Device. A single

Delivery Package MAY contain one or more Deployment Components. The format and structure of Delivery Package are out of scope of this specification.

The Client MUST support at least one of the following methods for delivery of Delivery Packages:

- Direct delivery: Using OMA DM Replace command. The OMA DM Large Object mechanism MAY also be used.
- Indirect delivery: Using alternate download mechanism, such as [DLOTA].

## 5.2.1 Delivery Package State Machine and Primitives

Delivery Package management Primitives trigger transition from one state to another. The Device internal operations and state transitions MUST appear to be atomic. If a state transition fails SCOMO Client MUST reverse the operation and make sure Delivery Package remains in the previous state.

Two kinds of Primitives are defined as below:

- Atomic Primitives: Download, Install, InstallInactive
- Composed Primitives: DownloadInstall, DownloadInstallInactive

When a Composed Primitive is executed, two state transitions happen in the Device. For example if DownloadInstall is executed, a Deployment Component transits from Not Downloaded State to Delivered State after successful download procedure. It transits to Active State after successful installation procedure. If the latter processing fails it remains in previous state and the second state transition does not happen.

Each Delivery Package MUST have one of the four states as follows:

- a) Not Downloaded state (which is a logical state, reflected by not having any sub-tree for the package, or having it in the Download branch)
- b) Delivered state (reflected in the State node in the Delivered branch)
- c) Installed state (reflected in the State node in the Delivered branch)
- d) Removed state (which is a logical state reflected by deleting the <x> sub-tree representing the Delivery Package from the Delivered branch)

The following figure depicts the state transition triggered by Delivery Package Primitives executed:

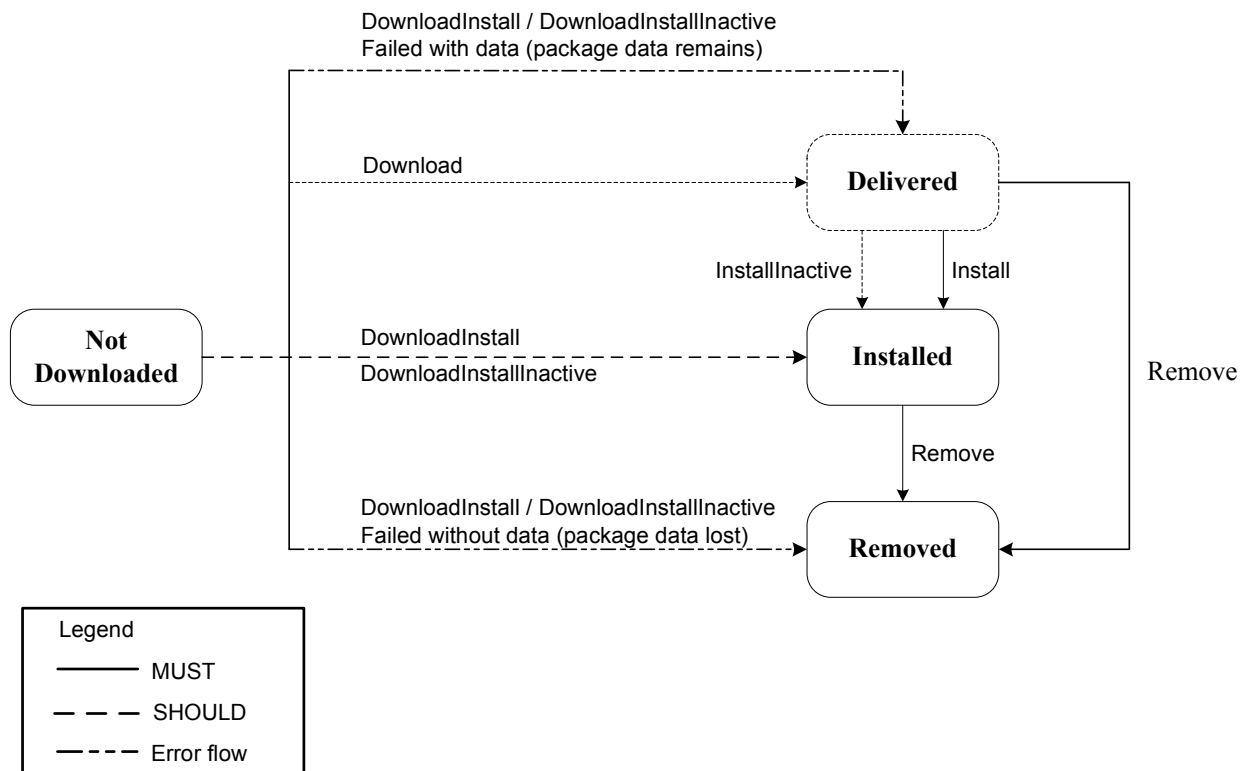


Figure 2: Delivery Package State Machine

### 5.2.1.1 Not Downloaded State

In Not Downloaded State a Delivery Package has not been downloaded to the Device.

The SCOMO Client **MUST** support this state.

### 5.2.1.2 Delivered State

In Delivered State a Delivery Package containing one or more Deployment Components has been delivered to the Device. The Deployment Component within the Delivery Package has not been installed and no corresponding nodes are added in the Deployed tree.

Delivered State also enables “deliver-first-install-later”-like use cases e.g. updates of mobile office solutions requiring all the components to be activated immediately after back-end update. In this case the Delivery and Deployment are discrete.

The SCOMO Client **SHOULD** be support this state.

### 5.2.1.3 Installed state

In Installed state the Delivery Package has been installed which means that Deployment Component(s) within it have been installed into the execution environment. The SCOMO Client **MUST** create corresponding nodes of each Deployment Component in the Deployed tree. The state of Deployment Components depends on the Delivery Package Primitive that has been executed. If the Primitive is ‘Install’ or ‘DownloadInstall’ then all Deployment Components have been installed and placed in Active state. If the Primitive is ‘InstallInactive’ or ‘DownloadInstallInactive’ then all Deployment Components have been installed and placed in Inactive state.

The SCOMO Client **MUST** support this state.

### 5.2.1.4 Removed State

In Removed state the Delivery Package has been removed from the Device. The 'Remove' Primitive of Delivery Package MUST NOT affect the state of installed Deployment Components. The SCOMO Client MUST delete the Delivery Package from the Device and delete corresponding nodes which represented it from Delivered tree.

The SCOMO Client MUST support this state.

### 5.2.1.5 Delivery Package Primitives

Delivery Package Primitives are used to trigger management process and state transition of a Delivery Package.

#### Note 1

A Client that supports alternate download mechanism MUST support at least one of the operations: Download, DownloadInstall, DownloadInstallInactive.

Primitive	Description	Applicable states	Post Primitive state	Primitive support dependency
Download	This Primitive is used to download a Delivery Package. The Delivery Package is placed in the Delivered state.	Not Downloaded	Delivered	SHOULD be supported See Note 1 above
DownloadInstall	This Primitive is used to download and install a Delivery Package. The Deployment Component(s) contained in the Delivery Package will be installed to Active State.	Not Downloaded	Installed	SHOULD be supported See Note 1 above
DownloadInstallInactive	This Primitive is used to download and install a Delivery Package. The Deployment Component(s) contained in the Delivery Package will be installed to Inactive State.	Not Downloaded	Installed	SHOULD be supported See Note 1 above
Install	This Primitive is used to install a Delivery Package. The Deployment Component(s) contained in the Delivery Package will be installed to Active State.	Delivered	Installed	MUST be supported
InstallInactive	This Primitive is used to install a Delivery Package. The Deployment Component(s) contained in the Delivery Package will be installed to Inactive State.	Delivered	Installed	SHOULD be supported
Remove	This Primitive is used to remove Delivery Package from the Device.	Delivered Installed	Removed	MUST be supported

Table 2 – Delivery Package Primitives

Note that installation of a Delivery Package means installing of the Deployment Components contained in the package, and these MAY be either Deployment Components that were not previously installed on the Device (install from scratch) or update to previously installed Deployment Components (install an update).

## 6. Standardized Management Objects

### 5.1 Introduction to Management Objects (Informative)

Management objects are the entities that can be manipulated by management actions carried over the OMA DM protocol. A Management Object can be as small as an integer or large and complex like a background picture, screen saver, or security certificate. The OMA DM protocol is neutral about the contents, or values, of the Management Objects and treats the node values as opaque data.

#### 6.1.1 Definition and description of Management Objects

OMA DM Management Objects are defined using the OMA DM Device Description Framework [DMTND], or DDF. The use of this description framework produces detailed information about the device in question. However, due to the high level of detail in these descriptions, they are sometimes hard for humans to digest and it can be a time consuming task to get an overview of a particular object's structure.

In order to make it easier to quickly get an overview of how a Management Object is organized and its intended use, a simplified graphical notation in the shape of a block diagram is used in this document. Even though the notation is graphical, it still uses some printable characters, e.g. to denote the number of occurrences of a node. These are mainly borrowed from the syntax of DTDs for XML. The characters and their meaning are defined in the following table.

Character	Meaning
+	one or many occurrences
*	zero or more occurrences
?	zero or one occurrences

If none of these characters is used the default occurrence is exactly once.

There is one more feature of the DDF that needs to have a corresponding graphical notation, the un-named block. These are blocks that act as placeholders in the description and are instantiated with information when the nodes are used at run-time. Un-named blocks in the description are represented by a lower case character in italics, e.g. *x*.

Each block in the graphical notation corresponds to a described node, and the text is the name of the node. If a block contains an *x*, it means that the name is not known in the description and that it will be assigned at run-time. The names of all ancestral nodes are used to construct the URI for each node in the Management Object. It is not possible to see the actual parameters, or data, stored in the nodes by looking at the graphical notation of a Management Object.

For a further introduction to this graphical notation, please refer to [DMStdObj].

## 6.2 DDF compliance

The Management Object descriptions in this document are normative. However, the descriptions also contain a number of informative aspects that could be included to enhance readability or serve as examples. Other informative aspects are, for instance, the ZeroOrMore and OneOrMore elements, where implementations may introduce restrictions. All these exceptions are listed here:

- All XML comments, e.g. “<!--some text →”, are informative.

- The descriptions do not contain an RTProperties element, or any of its child elements, but a description of an actual implementation of this object MAY include these.
- If a default value for a leaf node is specified in a description, by the DefaultValue element, an implementation MUST supply its own appropriate value for this element. If the DefaultValue element is present in the description of a node, it MUST be present in the implementation, but MAY have a different value.
- The value of all Man, Mod, Description and DFTitle elements are informative and included only as examples.
- Below the interior nodes Ext and BearerParams, an implementation may add further nodes at will.
- The contents of the AccessType element MAY be extended by an implementation.
- If any of the following AccessType values are specified, they MUST NOT be removed in an implementation: Copy, Delete, Exec, Get, and Replace.
- If the AccessType value Add is specified it MAY be removed in an implementation if the implementation only supports a fixed number of child nodes.
- An implementation MAY replace the ZeroOrMore or OneOrMore elements with ZeroOrN or OneOrN respectively. An appropriate value for *N* must also be given with the ...Or*N* elements.

### 6.2.1 Conformance Definitions

The status definition in the node definitions indicates if the client supports that node or not. If the status is “Required” then the client MUST support that node in the case the client supports the parent node.

## 7. Software Component Management Object

The Management Objects associated with Software Component management are assembled under an unnamed interior node *x*, dynamically or statically created.

**Protocol Compatibility:** This object is compatible with OMA Device Management protocol specifications, version 1.2 [DMPRO].

### 7.1 Tree Structure

SCOMO tree has a well-defined structure, with designated Ext nodes to allow non-standard extension nodes. The general structure of the tree is as follows:

1. Download – a sub-tree containing pre-delivery information and actions that are used for triggering the delivery and installation of Delivery Packages using indirect delivery mechanism, specified in section 5.2.
2. Inventory – a sub-tree containing post-delivery information and actions.
  - 2.1. Inventory/Delivered – a sub-tree containing post-delivery (but pre-installation) information of Delivery Packages. This sub-tree is created either by the Device after using indirect delivery mechanism or by the server before using direct delivery mechanism specified in section 5.2. This sub-tree contains actions for installation and removal.
  - 2.2. Inventory/Deployed – a sub-tree containing post-installation information of Deployment Components, as well as actions for activation, deactivation and removal.

### 7.2 State Information in the Tree

SCOMO specifies two state machines – one for Delivery Package (see 5.2.1) and one for Deployment Component (see 5.1.1).

The state of a given Delivery Package can be known by examining the location of its <X> branch (within the Download or Delivery sub-tree) and querying the value of Delivered/<X>/State node.

The state of a given Deployment Component can be known by the presence (or absence) of the its <X> branch and by querying the value of Deployed/<X>/State node (note that the Not Downloaded state and Removed states are logically different but the differentiation between them is mostly abstract).

### 7.3 Figure of the Management Object (Informative)



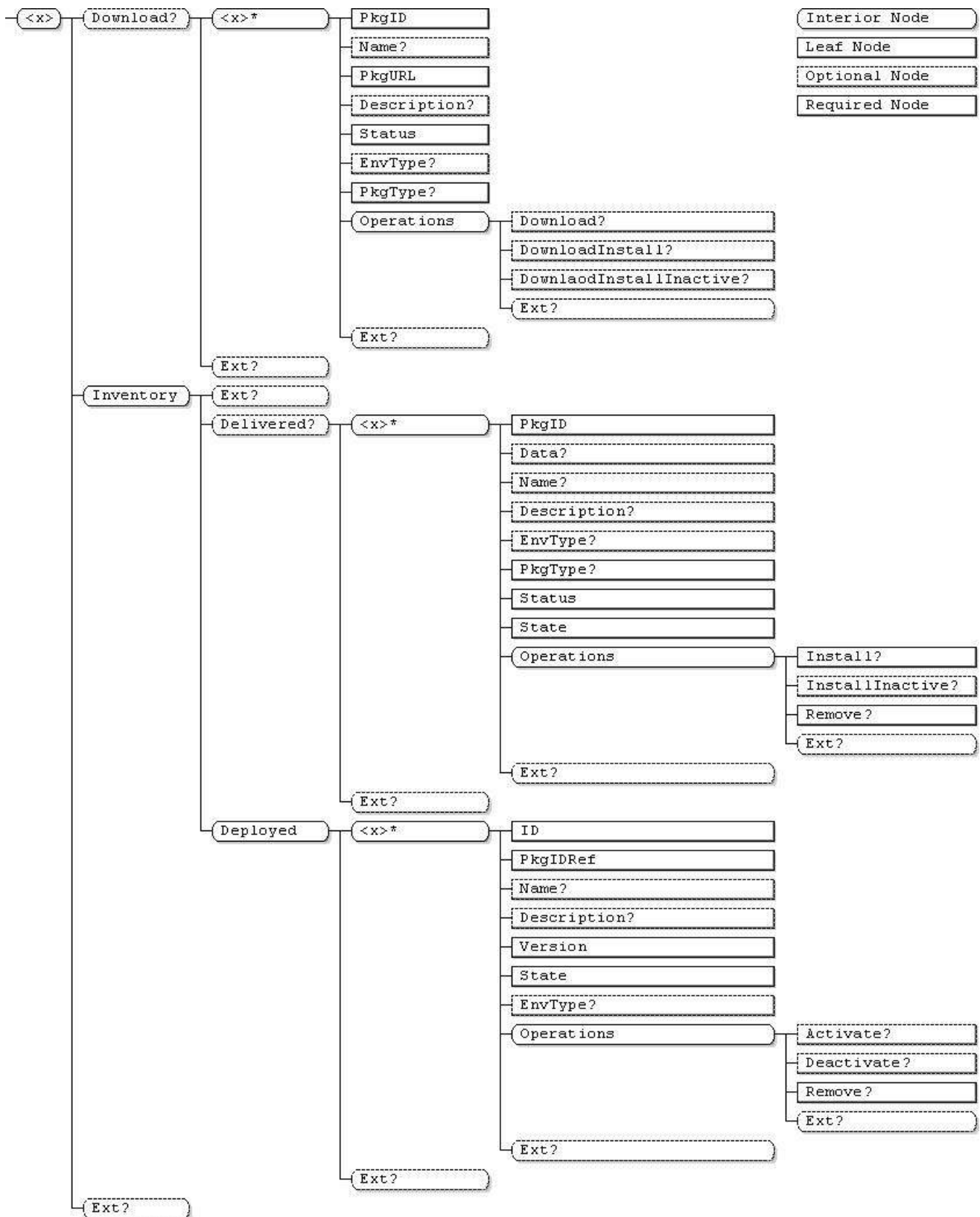


Figure 3: Software Component Management Object

## 7.4 Software Component Management Object Parameters

Software Component Management Object consists of following parameters:

<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups together the parameters of a Software Component Management Object. The ancestor elements of this node define the position in the Management Tree of this Management Object. But the structure of the DM tree and hence positions in the tree of Management Objects are out of scope of this specification.

The type of this node MUST be the SCOMO Management Object ID “urn:oma:mo:oma-scomo:1.0”.

### Download

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node is a parent node of the downloadable Delivery Packages that are ready for download or are in the process of downloading. This node MUST be used with the alternate download mechanism described in section 9.1.2 (as opposed to usage of OMA DM delivery, as described in section 9.1.1).

### Download/<X>

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This node groups pre-delivery information and actions that are used for triggering the delivery and installation of Delivery Packages using indirect delivery mechanism specified in section 5.1.2. Nodes in this sub-tree are used with Exec commands to start the Download operation to download the Delivery Package; to optionally install the Delivery Package; and to optionally move the Deployment Components into either the Active or the Inactive state.

### Download/<X>/PkgID

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the identifier for a Delivery Package. The PkgID MUST be provided by the Device Management System and uniquely identify the Delivery Package within the SCOMO tree.

### Download/<X>/Name

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Name for the Delivery Package. The Package Name MUST be provided by the Device Management System.

**Download/<X>/PkgURL**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the downloadable URL of the Delivery Package or its download descriptor. This URL is used for alternative download mechanisms (such as HTTP Get [RFC2616] or Descriptor Based Download [DLTA]).

**Download/<X>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node provides a description of the Delivery Package.

**Download/<X>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the Status of the Download and/or Install operation. This value is set by the SCOMO Client and the possible values are:

<u>Integer Value</u>	<u>Status</u>	<u>Description</u>
10	Idle / Start	There is no data available and download is about to start
20	Download Failed	Download failed and there is No data received
30	Download Progressing	Denotes that a download has started and that 0 or more bytes of data have been downloaded
40	Download Complete	Have data after Download has been completed successfully
50	Install Progressing	Denotes that an installation has started
60	Install Failed with data	Install failed and the downloaded package is still in the Device
70	Install Failed without data	Install failed and the downloaded package is deleted

**Download/<X>/EnvType**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Environment type of a Delivery Package. If a value is available for the EnvType then it MUST be provided by the Device Management System. The value of this leaf node is an URI [RFC3986] which unambiguously identifies the target environment for the package. Standardized values for the URI value MUST be registered with OMNA.

As SCOMO applies to Software Components and unlike complete programs and applications, these components might not be functional without an environment that is installed in the device to allow Software Components to be used on the device (in general this environment can consist of the OS, frameworks, libraries or other software elements). In such a case the EnvType can be used to address the environment that component is applicable to.

In cases that the component is targeted for a device which does not support the environment, the delivery package MUST be rejected; meaning that it cannot be downloaded, installed or activated in any form.

**Download/<X>/PkgType**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node specifies the Content Type of a Delivery Package. The value MUST be a MIME Content Type. The PkgType MUST be provided by the Device Management System.

**Download/<X>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a parent node for Primitives that can be executed to start a download of a Delivery Package into the Device and possibly its subsequent installation.

Each of the direct child nodes of this parent are marked Optional, but a Client that supports alternate download mechanism MUST support at least one of these child nodes.

**Download/<X>/Operations/Download**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the Download operation; to download the Delivery Package into the Device and into the Delivered state.

**Download/<X>/Operations/DownloadInstall**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the download operation to download the Delivery Package into the Device and to install the Deployment Component(s) (extracted from the Delivery Package) into the Active state.

**Download/<X>/Operations/DownloadInstallInactive**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the download operation to download the Delivery Package into the Device and to install the Deployment Component(s) (extracted from the Delivery Package) into the Inactive state.

**Download/<X>/Operations/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the download operations sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Download/<X>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the Download Delivery Package sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Download/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the download sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees

**Inventory**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a parent node for all the Delivery Packages and Deployment Components in the Device.

**Inventory/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the inventory sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Delivered**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node is a parent node of the delivered Delivery Packages in the Device.

**Inventory/Delivered/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This node acts as a placeholder for a Delivery Package in the Device.

The DM System provides the values for supported child nodes of this parent in either of two ways:

- When using OMA DM Replace as the delivery mechanism – these values are set by the DM Server using DM commands.
- When using alternate download mechanism – the Client MUST copy these values from the corresponding branch of the Download sub-tree.

**Inventory/Delivered/<X>/PkgID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the identifier of a Delivery Package in the Device.

The PkgID is provided by the Device Management System and uniquely identifies the Delivery Package within the SCOMO tree. A Delivery Package MUST preserve the same identifier after a state transition. For example, the same value of Download/<X>/PkgID that was used for the Delivery Package while it was in the Download sub-tree – MUST also be used as the value for Inventory/Delivered/<X>/PkgID in the Delivered sub-tree.

**Inventory/Delivered/<X>/Data**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	No Get

This leaf node specifies the actual binary Data of a Delivery Package. When using OMA DM for delivery, this node is targeted with a Replace command containing the data of the Delivery Package to be delivered.

**Inventory/Delivered/<X>/Name**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Name of a Delivery Package in the Device. The Package Name is provided by the Device Management System. A Delivery Package MUST preserve the same name after a state transition. For example, the same value of Download/<X>/Name that was used for the Delivery Package while it was in the Download sub-tree MUST also be used as the value for Inventory/Delivered/<X>/Name node.

**Inventory/Delivered/<X>/Description**

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node provides a description of the Delivery Package. A Delivery Package MUST preserve the same description after a state transition. For example, the same value of Download/<X>/Description that was used for the Delivery Package while it was in the Download sub-tree MUST also be used as the value for Inventory/Delivered/<X>/Description node.

**Inventory/Delivered/<X>/EnvType**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Environment type of a Delivery Package in the Device. The value of EnvType is provided by the Device Management System. The value is an URI [RFC3986] which unambiguously identifies the target environment for the package. Standardized values for the URI value MUST be registered with OMNA.

A Delivery Package MUST preserve the same EnvType after a state transition. For example, the same value of Download/<X>/EnvType that was used for the Delivery Package while it was in the Download sub-tree MUST also be used as the value for Inventory/Delivered/<X>/EnvType in the Delivered sub-tree.

As SCOMO applies to Software Components and unlike complete programs and applications, these components might not be functional without an environment that is installed in the device to allow Software Components to be used on the device (in general this environment can consist of the OS, frameworks, libraries or other software elements). In such a case the EnvType can be used to address the environment that component is applicable to.

In cases that the component is targeted for a device which does not support the environment, the delivery package MUST be rejected; meaning that it cannot be installed or activated in any form.

**Inventory/Delivered/<X>/PkgType**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node specifies the Content Type of a Delivery Package in the Device. The value MUST be a MIME content type. The PkgType MUST be provided by the Device Management System. A Delivery Package MUST preserve the same PkgType after a state transition. For example, the same value of Download/<X>/PkgType that was used for the Delivery Package while it was in the Download sub-tree MUST also be used as the value for Inventory/Delivered/<X>/PkgType in the Delivered sub-tree.

**Inventory/Delivered/<X>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the status of Remove or Install Primitive. This value is set by the SCOMO Client and the possible values are:

<u>Integer Value</u>	<u>Status</u>	<u>Description</u>
10	Idle / Start	The Device hasn't started the Remove or Install Primitive
20	Remove Failed	Remove failed and the Deployment Component is still in the Device
30	Remove Progressing	Denotes that Remove has started
40	Install Progressing	Denotes that an installation has started
50	Install Failed with data	Install failed and the downloaded package is still in the Device
60	Install Failed without data	Install failed and the downloaded package is deleted

**Inventory/Delivered/<X>/State**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the state of a Delivery Package in the Device. The value of this node is one of the following:

<u>Integer Value</u>	<u>State</u>	<u>Description</u>
10	Delivered	The Delivery Package is in the Delivered State
20	Installed	The Delivery Package is in the Installed State

**Inventory/Delivered/<X>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a parent node for Primitives that can be executed for Delivery Package in the Device.



**Inventory/Delivered/<X>/Operations/Install**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

This node is used with Exec command to start the Install operation; Deployment Component(s) from a single Delivery Package are installed into the Active state.

**Inventory/Delivered/<X>/Operations/InstallInactive**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the InstallInactive operation; Deployment Component(s) from a single Delivery Package are installed into the Inactive state.

**Inventory/Delivered/<X>/Operations/Remove**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

This node is used with Exec command to start the Remove operation; to remove the Delivery Package from the Device.

**Inventory/Delivered/<X>/Operations/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the Delivered operations sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Delivered/<X>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the Delivery Package parameters sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Delivered/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the delivered sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Deployed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a parent node of the Deployment Components in the Device.

**Inventory/Deployed/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This node acts as a placeholder for a Deployment Component in the Device. The node name is assigned by the SCOMO Client.

**Inventory/Deployed/<X>/ID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the identifier of a deployed Deployment Component in the Device. The ID MUST be provided by the Device Management System and uniquely identify the Deployment Component within the SCOMO tree. The implementation MAY generate execution environment specific ID for the Deployment Component during installation in the Device and maintain the mapping with Component ID provided by the Device Management System.

**Inventory/Deployed/<X>/PkgIDRef**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node provides the reference to the Delivery Package ID as specified in Download and/or Delivered tree in the Device. The value of this node is the same as the value of PkgID for corresponding Delivery Package if the Software Component is deployed via SCOMO mechanism. If the Software Component is deployed using other mechanism (such as the user install an application), the value of PkgIDRef MAY be empty.

**Inventory/Deployed/<X>/Name**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Name of a Deployment Component in the Device.

**Inventory/Deployed/<X>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node provides a description of the Deployment Component in the Device.

**Inventory/Deployed/<X>/Version**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Version of a Deployment Component in the Device.

The format of the version string is implementation specific and MAY depend on the characteristics of the Deployment Component, however this opaque string SHOULD encapsulate information that can differentiate between different versions (e.g. different editions) of the same Deployment Component.

A common and widely accepted version string contains a sequence of two or more positive integers, separated by ‘.’ character, for example: “2.1.1”.

A Deployment Component that does not have any version information at the time of installation MAY use an empty string, to denote that it is version-less.

**Inventory/Deployed/<X>/State**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the State of a Deployment Component in the Device. The value of this node is one of the following:

<u>Integer Value</u>	<u>State</u>	<u>Description</u>
10	Inactive	The Deployment Component is in the Inactive State
20	Active	The Deployment Component is in the Active State

**Inventory/Deployed/<X>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the Status of RemovePrimitive. This value is set by the SCOMO Client and the possible values are:

<u>Integer Value</u>	<u>Status</u>	<u>Description</u>
10	Idle / Start	The Device hasn't started the Remove operation
20	Remove Failed	Remove failed and the component is still in the Device

30	Remove Progressing	Denotes that Remove has started
40	Activate Failed	Activate failed and the component is still in the Inactive state
50	Activate Progressing	Denotes that the Activate operation has started
60	Deactivate Failed	Deactivate failed and the component is still in the Active state
70	Deactivate Progressing	Denotes that the Deactivate operation has started

**Inventory/Deployed/<X>/EnvType**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the Environment type of a Deployment Component in the Device. The value of this leaf node is an URI [RFC3986] which unambiguously identifies the target environment that was used for installation of the Deployment Component. Standardized values for the URI value MUST be registered with OMNA

**Inventory/Deployed/<X>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a parent node for Primitives that can be executed for a Deployment Component in the Device.

**Inventory/Deployed/<X>/Operations/Activate**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the Activate operation; to transfer the Deployment Component from the Inactive state to the Active state.

**Inventory/Deployed/<X>/Operations/Deactivate**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get,

This node is used with Exec command to start the Deactivate operation; to transfer the Deployment Component from the Active state to the Inactive state.

**Inventory/Deployed/<X>/Operations/Remove**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

This node is used with Exec command to start the Remove operation; to Remove the Deployment Component from the Device. The presence of this node means the Deployment Component supports the Remove operation.

**Inventory/Deployed/<X>/Operations/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the Deployment Component operations sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Deployed/<X>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the Deployment Component parameters sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Inventory/Deployed/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the deployed sub-tree into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

**Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the SCOMO root into which platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Software Component Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

## 8. Behavior associated with the Management Object

In the SCOMO tree the Exec command is only allowed to target Primitives under Operations nodes. After an Exec command to one of the Primitives under Operations node, the SCOMO Client MUST send response to the SCOMO Server in either of the following ways:

1. Asynchronously if the Exec command is acceptable and will be executed asynchronously.
  - The SCOMO Client MUST return status code 202 (“Accepted for processing”) for the Exec command as defined in [DMREPRO]
  - Upon completion of the asynchronous operation, the SCOMO Client MUST send an alert back to the SCOMO Server by using the Generic Alert defined in [DMPRO].
2. Synchronously: If the Exec command is acceptable and executed synchronously.
  - If the Exec command is performed successfully, the SCOMO Client MUST return SCOMO Result Code 1200 (“Command completed successfully”) for the Exec command.

In SCOMO Client’s response message for reporting purpose, the SCOMO Client MUST include the Delivery Package or Deployment Component information in addition to the status code described above:

- URI(s) of newly created dynamic node(s) <x> for Delivery Package or Deployment Component as a result of the Primitives executed in <Item>/<Target> element
- URI of the Primitive node on which the Exec command was invoked in <Item>/<Source> element
- For asynchronous reporting, Delivery Package PkgID or Deployment Component ID in <Item>/<Data> element which format is ‘xml’
- For asynchronous reporting, SCOMO Result Code of the Primitive executed in <Item>/<Data> element which format is ‘xml’

After an Exec command to one of the Primitives under Operations node it is the SCOMO Client’s responsibility to reflect the changes in the Inventory sub-tree as follows:

- After successful installation of Deployment Components, the SCOMO Client MUST create their nodes in Deployed sub-tree.
- The SCOMO Client MUST create nodes for Delivery Package in the Delivered sub-tree, in the case a Download Primitive was executed successfully.
- The SCOMO Client MAY create nodes for Delivery Package in the Delivered sub-tree, in the case a DownloadInstall or DownloadInstallInactive Primitive was executed successfully. In this case it MAY also remove these nodes after Deployment Components were installed successfully.

The detailed Exec command semantics and SCOMO Client responsibility are described below.

### 8.1 Structure Elements of <Item>/<Data> for Reporting

As described above, the Delivery Package PkgID or Deployment Component ID will be included in xml structure in <Item>/<Data> element. The xml structure is defined below:

#### 8.1.1 ResultCode

Usage: Specify the SCOMO Result Code of the Primitive executed

Parent Elements: none

Restrictions: The Format of this element MUST be int.

Content Model: (#PCDATA)

## 8.1.2 Identifier

Usage: Specify the Delivery Package ID or Deployment Component ID

Parent Elements: none

Restrictions: The Format of this element MUST be chr.

Content Model: (#PCDATA)

## 8.2 Exec command semantics on Download sub-tree

Before receiving the Exec command, the following pre-condition needs to be satisfied:

- Download/<X> dynamic node creation
- At least the following nodes need to be set with an appropriate value:
  - Download/<X>/PkgID
  - Download/<X>/PkgURL
  - Download/<X>/Operations... (at least one Operation sub-node for the Primitive to be executed)
- In case that Download/EnvType is specified and is incompatible with the environment on the device, the SCOMO Client MUST respond with a status code indicating a failure (in case of a synchronous response) or with SCOMO Result Code 1413 (Operation rejected – unsupported environment) in case of an asynchronous operation.

The Exec command targeting a child node of an Operations node starts the execution of a chosen Primitive. For example:

```
<Exec>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI>./SCM/Download/Package123/Operations/DownloadInstall</LocURI>
    </Target>
  </Item>
</Exec>
```

If the Exec command is targeting any Primitives under the Download sub-tree, the SCOMO Client MUST start the download of the Delivery Package. If the Exec command is targeting the DownloadInstall or the DownloadInstallInactive Primitive, after successful download the SCOMO Client MUST start the installation of the Deployment Components.

The SCOMO Client responsibilities are described in the following table. Grey cells describe the case and white cells show the resulting responsibilities:

Primitive	Download	DownloadInstall or DownloadInstallInactive	DownloadInstall or DownloadInstallInactive
<b>download procedure</b>	succeeds	Succeeds	Succeeds
<b>installation procedure</b>	N/A	Fails	Succeeds
<b>create nodes in Delivered sub-tree and send Delivery Package information to the SCOMO Server</b>	MUST	MAY	MAY
<b>Create nodes in the Deployed sub-tree and send Deployment Component information to the SCOMO Server</b>	N/A (no installation)	N/A (installation failed)	MUST
<b>Remove nodes from the Download sub-tree</b>	SHOULD	MAY	SHOULD

**Table 3 – Responsibilities associated with Exec on Download sub-tree**

After the nodes are created in the Delivered or the Deployed sub-tree as described above, the SCOMO Client **MUST** assign ACL permissions to the new <x> node and its sub-nodes to ensure the originally authorized DM Servers have the same permission to the Delivery Package or Deployment Component as before download or installation.

In addition to applying ACL inheritance rules as defined in [DMTND], the SCOMO Client copies permissions into the ACL property value of new dynamic node <x> and its sub-nodes as described below:

- The ACL permissions of parent node <x> of the Primitive node on which the Exec command was invoked are copied to the new <x> node in Delivered or Deployed sub-tree. These copied permissions are added to those inherited as per [DMTND]
- The ACL permissions of Primitive node on which the Exec command was invoked are copied to new Primitive nodes under new <x>/Operations node in Delivered or Deployed sub-tree. These copied permissions are added to those inherited as per [DMTND].
- The ACL permissions of other sub nodes under new <x> node inherit the ACL permissions from its parent node according to ACL inheritance rules as defined in [DMTND].
- The ACL permissions of Ext nodes are implementation specific.

When trying to process an installation request - if the Deployment Component ID already exists, the SCOMO Client **MUST** either replace the value of existing nodes in the Deployed sub-tree to avoid ID clashes within SCOMO tree (which implies an update of an existing Deployment Component) or respond with an appropriate SCOMO Result code to indicate an error (which implies that an update is not possible).

## 8.3 Exec command semantics on Delivered

Before receiving the Exec command, the following pre-condition needs to be satisfied:



- Delivered/<X> dynamic node creation
- At least the following nodes need to be set with an appropriate value:
  - Delivered /<X>/PkgID
  - Delivered/<X>/Data
  - Delivered/<X>/Operations... (at least one Operation sub-node for the Primitive to be executed)
- In case that Delivered/EnvType is specified and is incompatible with the environment on the device the SCOMO Client MUST respond with a SCOMO Result Code 1407 (failed package validation).

The Exec command targeting a child node of an Operations node starts the execution of a chosen Primitive. For example:

```

<Exec>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI>./SCM/Inventory/Delivered/Package456/Operations/Install</LocURI>
    </Target>
  </Item>
</Exec>
    
```

If the Exec command is targeting Install Primitives, the SCOMO Client MUST start the installation of the Deployment Components.

The SCOMO Client responsibilities in that case are described in the following table. Grey cells describe the case and white cells show the resulting responsibilities:

Installation procedure	succeeds	failed with data (Delivery Package data remains in the Device)	failed without data (Delivery Package data lost)
<b>Create nodes in the Deployed sub-tree and send Deployment Component information to the SCOMO Server</b>	MUST	MUST NOT	MUST NOT
<b>Remove nodes from the Delivered sub-tree</b>	MAY	MUST NOT	MUST

**Table 4 - Responsibilities associated with Exec on Delivered sub-tree**

If the Exec command is targeting Remove Primitives under the Delivered sub-tree, the SCOMO Client MUST remove the Delivery Package from the Device. In this case the SCOMO Client MUST remove nodes of the Delivery Package from Delivered sub-tree.

After the nodes are created in the Deployed sub-tree as described above, the SCOMO Client MUST assign ACL permissions to new <x> node and its sub-nodes to ensure the originally authorized DM Servers have the same permission to the Delivery Package or Deployment Component as before installation.

In addition to applying ACL inheritance rules as defined in [DMTND], the SCOMO Client copies permissions into the ACL property value of the new dynamic node <x> and its sub-nodes as described below:

- The ACL permissions of the parent node <x> of the Primitive node on which the Exec command was invoked are copied to the new <x> node in the Deployed sub-tree.

- The ACL permissions of the Primitive node on which the Exec command was invoked are copied to the new Primitive sub nodes under new <x>/Operations node in the Deployed sub-tree.
- The ACL permissions of other sub nodes of the new <x> node inherit the ACL property value from its parent node according to ACL inheritance rules as defined in [DMTND].
- The ACL permissions of Ext nodes are implementation specific.

When trying to process an installation request - if the Deployment Component ID already exists, the SCOMO Client MUST either replace the value of existing nodes in the Deployed sub-tree to avoid ID clashes within SCOMO tree (which implies an update of an existing Deployment Component) or respond with an appropriate SCOMO Result code to indicate an error (which implies that an update is not possible).

## 8.4 Exec command semantics on Deployed

Before receiving the Exec command, the following pre-conditions need to be satisfied:

- Deployed/<X> dynamic node creation
- At least the following nodes need to be set with an appropriate value:
  - Deployed /<X>/PkgIDRef
  - Deployed/<X>/Operations... (at least one Operation sub-node for the Primitive to be executed)

The Exec command targeting a child node of an Operations node starts the execution of a chosen Primitive. For example:

```
<Exec>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI>./SCM/Inventory/Deployed/Component789/Operations/Activate</LocURI>
    </Target>
  </Item>
</Exec>
```

If the Exec command is targeting Remove Primitive under the Deployed sub-tree, the SCOMO Client MUST remove the Deployment Component from the Device and remove all the nodes of the Deployment Component from the Deployed sub-tree.

## 8.5 Use of Synchronous Reporting

If the Exec command is executed synchronously and succeeds, the SCOMO Client MUST send Delivery Package or Deployment Component information in a <Status> command for Exec command as described below:

- The <Data> element MUST contain a valid SCOMO Result Code (Sec 8.7)
- The URI of the Primitive node on which the Exec command was invoked – Used to identify the source MUST be included in the <Source>/<LocURI> of <Status>/<Item> element.
- The URI(s) of the newly created dynamic node(s) <x> under the Delivered or Deployed tree for the Delivery Package or Deployment Component – used to identify the result MUST be included in the <Target>/<LocURI> of <Status>/<Item> element.

In case nodes are created for multiple Deployment Components after package installation, they MUST be individually represented in multiple Item elements in a <Status>. In each Item element, the Target/LocURI element contains the URI of the newly created dynamic node(s) <x> under the Deployed sub-tree, and the Source/LocURI contains the URI of the executable Primitive node on which the Exec command was invoked.

The following is the example message:

```
<Status>
  <MsgRef>1</MsgRef>
  <CmdRef>2</CmdRef>
  <Cmd>Exec</Cmd>
  <Data>1200</Data>   --- SCOMO Result Code -->
  <Item>
    <Source>
      <LocURI>./SCOMO/Download/Package1/Operations/DownloadInstall</LocURI>
    </Source>
    <Target>
      <LocURI>./SCOMO/Inventory/Deployed/Component1</LocURI>
    </Target>
  </Item>
  <Item>
    <Source>
      <LocURI>./SCOMO/Download/Package1/Operations/DownloadInstall</LocURI>
    </Source>
    <Target>
      <LocURI>./SCOMO/Inventory/Deployed/Component2</LocURI>
    </Target>
  </Item>
</Status>
```

## 8.6 Use of the Generic Alert

If the Exec command is executed asynchronously, the SCOMO Client MUST send a notification to the SCOMO Server about the outcome of the operation via a Generic Alert [DMPRO] message. The SCOMO Client MAY send the alert message during the same or subsequent DM session. The alert message includes the following data:

- Delivery Package PkgID or Deployment Component ID in <Item>/<Data> element which format is 'xml' (see section 8.1).
- An integer Result Code – Used to report status of the operation. The appropriate SCOMO Result Code defined in section 8.7 MUST be included in an Item/Data element which format is 'xml' (see section 8.1).
- The URI of the Primitive node on which the Exec command was invoked – Used to identify the source MUST be included in the Source/LocURI element within the Generic Alert.
- The URI(s) of the newly created dynamic node(s) <x> under the Delivered or Deployed tree for the Delivery Package or Deployment Component – used to identify the result MUST be included in the Target/LocURI within the Generic Alert.
- An alert type – Used to identify the operation MUST be included.

Alerts that are reporting an error or failure condition SHOULD report an importance level defined in [DMPRO] higher than Informational in the Mark field of the Meta information. If the SCOMO Server needs to retrieve additional information, such as download or install status described in Download/<X>/Status node, then the SCOMO Server MAY query the Device for those specific nodes.

In case nodes are created for multiple Deployment Components after package installation, they MUST be individually represented in multiple Item elements in a single Generic Alert message. In each Item element, the Target/LocURI element contains the URI of the newly created dynamic node(s) <x> under the Deployed sub-tree, and the Source/LocURI contains the URI of the executable Primitive node on which the Exec command was invoked.

The following is the example message:

```

<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>--- Generic Alert -->
  <Correlator>correlator1</Correlator>
  <Item>
    <Source><LocURI>./SCM/Inventory/Delivered/Package456/Operations/Install</LocURI></Source>
    <Target><LocURI>./SCM/Inventory/Deployed/Component1</LocURI></Target>
    <Meta>
      <Type xml"s="syncml:met"nf">
        urn:oma:at:scomo:1.0:OperationComplete
      </Type>
      <Format xml"s="syncml:met"nf">text/plain</Format>
      <Mark xml"s="syncml:met"nf">warning</Mark>
    </Meta>
    <Data>
      <![CDATA[
        <ResultCode>1200</ResultCode>      <!--SCOMO Result Code -->
        <Identifier>Component1ID</Identifier>
      ]]>
    </Data>
  </Item>
  <Item>
    <Source><LocURI>./SCM/Inventory/Delivered/Package456/Operations/Install</LocURI></Source>
    <Target><LocURI>./SCM/Inventory/Deployed/Component2</LocURI></Target>
    <Meta>
      <Type xml"s="syncml:met"nf">
        urn:oma:at:scomo:1.0:OperationComplete
      </Type>
      <Format xml"s="syncml:met"nf">text/plain</Format>
      <Mark xml"s="syncml:met"nf">warning</Mark>
    </Meta>
    <Data>
      <![CDATA[
        <ResultCode>1200</ResultCode>      <!--SCOMO Result Code -->
        <Identifier>Component2ID</Identifier>
      ]]></Data>
    </Item>
  </Alert>

```

## 8.7 SCOMO Result Code

The Result Code of the operation MUST be sent as an integer value in Item/Data element of the GenericAlert [DMPRO] message or in response to an Exec command in case of synchronous execution. The Result Code MUST be one of the values defined below:

Result Code	Result Message	Informative Description of Status Code Usage
1200	Successful	Successful-1 - The Request has Succeeded
1250-1299	Successful – Vendor Specified	Successful Operation with vendor specified Result
1400	Client Error	Client error – based on User or Device behaviour

1401	User cancelled	User chose not to accept the operation when prompted
1402	Download Failed	The Software Component download failed
1403	Alternate Download Authentication Failure	Authentication was Required but Authentication Failure was encountered when downloading Software Component
1404	Download failed due to Device is out of memory	The download failed due to insufficient memory in the Device to save the Delivery Package.
1405	Install Failed	Software Component installation failed in the Device
1406	Install failed due to Device out of memory	The install failed because there was not sufficient memory to install the Software Component in the Device.
1407	Failed package validation	Failure to positively validate digital signature of the Delivery Package
1408	Remove failed	The Software Component Remove operation failed
1409	Activate failed	The Software Component Activate operation failed
1410	Deactivate failed	The Software Component Deactivate operation failed
1411	Not Implemented	The Device does not support the requested operation.
1412	Undefined Error	Indicates failure not defined by any other error code
1413	Operation rejected – unsupported environment	The Operation has been rejected because the device does not support the target environment type.
1450-1499	Client Error – Vendor Specified	Client Error encountered for Operation with vendor specified Result Code
1500	Alternate Download Server Error	Alternate Download Server Error Encountered
1501	Alternate Download Server Unavailable	The Alternate Download Server is unavailable or does not respond
1550-1599	Alternate Download Server Error – Vendor Specified	Alternate Download Server Error encountered for Operation with vendor specified Result Code

Table 5 - SCOMO Result Codes

## 8.7.1 Alert Types for the Software Component Management Object

The following alert type MUST be used in a Generic Alert [DMPRO] message originating from a Software Component Management Object. The alert type is used to identify the operation that was performed on the Device.

- urn:oma:at:scomo:1.0:OperationComplete

## 8.8 User Interaction Commands

This section deals with the SCOMO Client behavior associated with user interactions commands sent to the Device.

### 8.8.1 Requesting User Confirmation

The SCOMO Server MAY request user confirmation before performing SCOMO operations within the Device. However it is important to ensure that the SCOMO Client supports the ability to handle the user confirmation request.

In particular the SCOMO Client:

- SHOULD support DISPLAY Alert as described in [DMPRO].
- SHOULD support CONFIRM OR REJECT Alert as described in [DMPRO].
- SHOULD support USER INPUT Alert as described in [DMPRO].
- SHOULD support USER CHOICE Alert as described in [DMPRO].

The SCOMO Server SHOULD support all these Alerts.

## 9. SCOMO Usage (Informative)

This section describes some common flows for the purpose of Software Component management.

### 9.1 Delivery of Deployment Components

SCOMO allows a Deployment Component to be delivered either via OMA DM delivery mechanism or using an alternate download mechanism such as [DLOTA]

#### 9.1.1 OMA DM Delivery

OMA DM delivery is achieved by the DMS sending a Replace command containing the actual Deployment Component data to the /Inventory/Delivered/<X>/Data node. The DM Large Object mechanism MAY be used.

#### 9.1.2 Alternate Download

Step 1: DMS sets the appropriate values in Download/<X>/ sub-tree as specified in section 8.2

Step 2: DMS performs an Exec on one of the nodes under Download/<X>/Operations

Step 3: The Device returns an appropriate Generic Alert indicating status of the requested operation as specified in section 8.6 Use of the Generic Alert.

### 9.2 Installation of Deployment Components

Step 1: The DMS performs an Exec on one of the supported installation nodes such as Inventory/Delivered/<X>/Operations/Install or Inventory/Delivered/<X>/Operations/InstallInactive

Step 2: The Deployment Component installation process is initiated on the Device and upon completion an appropriate status is returned to the DMS using a Generic Alert as specified in section 8.6 Use of the Generic Alert.

### 9.3 Activation/De-activation of Deployment Components

Step 1: The DMS performs an Exec on Inventory/Deployed /<X>/Operations/Activate or Inventory/Deployed /<X>/Operations/Deactivate depending on whether activation or deactivation is desired.

Step 2: The Deployment Component activation or deactivation process is initiated on the Device and upon completion an appropriate status is returned to the DMS using a Generic Alert as specified in section 8.6 Use of the Generic Alert.

### 9.4 Removal of Software Components

Step 1: The DMS performs an Exec on Inventory/Deployed /<X>/Operations/Remove or Inventory/Delivered /<X>/Operations/Remove

Step 2: The Deployment Component removal process is initiated on the Device and upon completion an appropriate status is returned to the DMS using a Generic Alert as specified in section 8.6 Use of the Generic Alert.

### 9.5 Retrieving Inventory of Deployment Components

Step 1: The DMS performs a GET on the Inventory/ node (or the Inventory/Delivered or Inventory/Deployed nodes directly) to find out what components are delivered or deployed on the Device.

Step 2: The Device returns the inventory to the DMS.

## 10. Security Considerations (Informative)

Security for delivery and management of Software Components is of paramount importance. It is envisioned that authentication mechanisms supported by OMA DM protocol [DMPRO] will be used to ensure that authenticated entities can deliver/perform management operations on Software Components on the Device. Similarly, authorization for management operations on Software Components under the purview of this enabler will be based on the ACL mechanisms defined by OMA DM TND [DMTND].

The SCOMO enabler does not mandate nor restrict any mechanism to guarantee authenticity, confidentiality and integrity of Software Components delivered to the Device. It is envisioned that existing security mechanisms for this purpose such as Digital Signatures, SSL [SSL3.0], TLS [TLS1.0], etc. can easily work in conjunction with SCOMO.



## Appendix A. Change History

(Informative)

### A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

### A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-TS-MO-Software_Component-V1_0	01 Mar 2005	All	Initial version of the document.
	04 Apr 2005	All	Minor changes based on comments from and after Paris meeting.
Draft Versions: OMA-TS-DM-SCOMO-V1-0	21 Feb 2006	AK	Clerical changes to prepare candidate for baseline document. State Diagrams updated. Accepting several questions, as no comments have been received.
	01 Aug 2006	Section 6.1, section 7.4.1	Clerical change: MO diagram made consistent with textual descriptions Incorporated CR: OMA-DM-SCOMO-2006-0024R02
	08 Jan 2007	All	Incorporated OMA-DM-SCOMO-2006-0044R04 OMA-DM-SCOMO-2006-0045R01 OMA-DM-SCOMO-2006-0049R01
	02 Mar 2007		Incorporated: OMA-DM-SCOMO-2007-0001R01 OMA-DM-SCOMO-2007-0002R01 OMA-DM-SCOMO-2007-0003R01 OMA-DM-SCOMO-2007-0009-
	06 Apr 2007		Incorporated: OMA-DM-SCOMO-2007-0010
	03 May 2007	6, 7	Incorporated: OMA-DM-SCOMO-2007-0013R02 OMA-DM-SCOMO-2007-0022 OMA-DM-SCOMO-2007-0023 OMA-DM-SCOMO-2007-0029R01 OMA-DM-SCOMO-2007-0031
	25 May 2007	All	Incorporated: OMA-DM-SCOMO-2007-0013R02 OMA-DM-SCOMO-2007-0021R02
	29 Jun 2007	1, 2, 3, 5, 6	Incorporated: OMA-DM-SCOMO-2007-0033 OMA-DM-SCOMO-2007-0034R02 OMA-DM-SCOMO-2007-0036R01 OMA-DM-SCOMO-2007-0038R02 OMA-DM-SCOMO-2007-0040 OMA-DM-SCOMO-2007-0041R02
	25 July 2007	All (e.g. 2.1, 7, 7.1, 7.2, 7.3, 7.4, 7.4.1, 7.4.2, 7.4.3, 7.4.5)	Incorporated: OMA-DM-SCOMO-2007-0038R02 OMA-DM-SCOMO-2007-0046R03 OMA-DM-SCOMO-2007-0049
	01 Aug 2007	5.1, 6.4, 7.2	OMA-DM-SCOMO-2007-0046R03- OMA-DM-SCOMO-2007-0048R03 OMA-DM-SCOMO-2007-0050
	08 Aug 2007	5.3, 5.3.* 6.3, 6.4 7, 7.1-7.5	OMA-DM-SCOMO-2007-0051R01 OMA-DM-SCOMO-2007-0044R04

Document Identifier	Date	Sections	Description
	09 Sep 2007	6.4, 1, 5.2	OMA-DM-SCOMO-2007-0011R08- OMA-DM-SCOMO-2007-0054
	30 Sep 2007	5.3.1, 5.3.4, 5.4.5, 7, 7.2, 7.3, 7.5	OMA-DM-SCOMO-2007-0067 OMA-DM-SCOMO-2007-0068R02 OMA-DM-SCOMO-2007-0071R01
	01 Oct 2007	5.*	OMA-DM-SCOMO-2007-0069
	17 Oct 2007	5.1.1 7.2, 7.3 7.5.1	OMA-DM-SCOMO-2007-0052R03 OMA-DM-SCOMO-2007-0075 OMA-DM-SCOMO-2007-0073
	30 Oct 2007	7 New 7.2	OMA-DM-SCOMO-2007-0055R03
	11 Nov 2007	Relocate section 7.2 as 7.5 All sections	OMA-DM-SCOMO-2007-0055R03 OMA-DM-SCOMO-2007-0082 OMA-DM-SCOMO-2007-0081 OMA-DM-SCOMO-2007-0079 OMA-DM-SCOMO-2007-0078 OMA-DM-SCOMO-2007-0080R01 Editorial – device → Device
	20 Dec 2007	6.4 7.6.2 Appendix B	OMA-DM-SCOMO-2007-0086R01 OMA-DM-SCOMO-2007-0091 OMA-DM-SCOMO-2007-0090R01 OMA-DM-SCOMO-2007-0066R07 OMA-DM-SCOMO-2007-0087 OMA-DM-SCOMO-2007-0083 OMA-DM-SCOMO-2007-0085R01
	08 Feb 2008	6.3, 7.2	OMA-DM-SCOMO-2008-0001 OMA-DM-SCOMO-2008-0002
	20 Feb 2008	All	Editorial changes
	26 Feb 2008	5.1.2, 5.2.1.5, Appendix B	OMA-DM-SCOMO-2008-0005 OMA-DM-SCOMO-2008-0009 OMA-DM-SCOMO-2008-0007R01
	13 Mar 2008	All	Editorial changes based on initial closure review.
	20 Mar 2008	1, 6.2, 6.3, 8.5	Editorial changes based on continued closure review.
	27 Mar 2008	6-2 - 7.6	Editorial changes based on continued closure review.
	06 Apr 2008	all	Editorial changes based on continued closure review.
	16 Apr 2008	2.2, 5.1.2, 5.2, 6.2, 6.3, 7, 7.6	OMA-DM-SCOMO-2008-0006R01 OMA-DM-SCOMO-2008-0008 OMA-DM-SCOMO-2008-0010 OMA-DM-SCOMO-2008-0011R01 OMA-DM-SCOMO-2008-0012 OMA-DM-SCOMO-2008-0013R01 OMA-DM-SCOMO-2008-0014 OMA-DM-SCOMO-2008-0016 OMA-DM-SCOMO-2008-0018R01
	3 June 2008	6.1, 6.2, 7	OMA-DM-SCOMO-2008-0017 OMA-DM-SCOMO-2008-0026 OMA-DM-SCOMO-2008-0024
	7 July 2008	Template & Diagram	Update to the latest TS template Fixed editorial errors in the MO diagram
	15 July 2008	6, 7.2, 7.3	OMA-DM-SCOMO-2008-0025R02 Editorial change to the MO diagram
	22 July 2008	2.1, 6.4, 7.2, 7.3	OMA-DM-SCOMO-2008-0019R02 OMA-DM-SCOMO-2007-0076R04
	13 Aug 2008	5.2, 7.6.2,	Editorial changes, including the fix of wrong references that should

Document Identifier	Date	Sections	Description
		Annex B	now point to the correct parts of the spec.
	03 Sep 2008	All	OMA-DM-SCOMO-2008-0033 OMA-DM-SCOMO-2008-0034R01 OMA-DM-SCOMO-2008-0037R01 OMA-DM-SCOMO-2008-0035R02 OMA-DM-SCOMO-2008-0036 OMA-DM-SCOMO-2008-0039 OMA-DM-SCOMO-2008-0040
	07 Sep 2008	Annex B	Editorial update of references (due to changed numbering)
	08 Sep 2008	8.6	
	17 Sep 2008	5.2.1.5 8.5 8.6.1 Annex B	OMA-DM-SCOMO-2008-0041 OMA-DM-SCOMO-2008-0043 OMA-DM-SCOMO-2008-0046- OMA-DM-SCOMO-2008-0048 Minor editorial changes
	19 Sep 2008	7.4	Editorial letter-case
	20 Sep 2008	MO Diagram	Editorial changes (to support automatic generation of DDF and diagram)
	22 Sep 2008	7.4	Editorial (restored a sentence that was chopped)
	29 Sep 2008	All	Final editorial clean up prior to Candidate approval
	01 Oct 2008	5.2.1.2 7.3 7.4 Annex B1	OMA-DM-SCOMO-2008-0049 OMA-DM-SCOMO-2008-0050
	10 Oct 2008	8, 8.2, 8.3, 8.5, 8.6	OMA-DM-SCOMO-2008-0051 Editorial changes (lettercase)
	13 Oct 2008	All	Final clean up prior to Candidate approval
Candidate Versions: OMA-TS-DM-SCOMO-V1-0	24 Oct 2008	n/a	Status changed to Candidate by OMA TP TP Ref# OMA-TP-2008-0414- INP_Software_Component_Management_Object_V1_0_ERP_for_Can didate_Approval
Draft Versions: OMA-TS-DM-SCOMO-V1-0	25 Feb 2009	All	Updated with agreed CRs: OMA-DM-SCOMO-2008-0052R01 OMA-DM-SCOMO-2009-0001 OMA-DM-SCOMO-2009-0003
Candidate Versions: OMA-TS-DM-SCOMO-V1-0	17 Mar 2009	n/a	Status changed by TP OMA-TP-2009-0120-INP_SCOMO_V1_0_ERP_for_notification
Draft Version OMA-TS-SCOMO-V1_0	29 Apr 2009	7.4	Editorial Fix
	30 Apr 2009	All	Editorial Fixes
Candidate Version: OMA-TS-SCOMO-V1-0	19 May 2009	n/a	Status changed to Candidate by TP TP Ref# OMA-TP-2009-0220R01- INP_SCOMO_V1_0_ERP_for_Notification
Draft Version: OMA-TS-SCOMO-V1-0	06 Jul 2009	All	Incorporation of CR: OMA-DM-SCOMO-2009-0007R02-CR_SCOMO_TS_Bug_Fixes
Candidate Version: OMA-TS-SCOMO-V1-0	10 Jul 2009	All	Status changed to Candidate by TP TP Ref# OMA-TP-2009-0318- INP_SCOMO_V1_0_ERP_for_Notification

## Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

### B.1 SCR for SCOMO Tree Structure

Item	Function	Reference	Requirement
SCOMO-T-001-M	Use of appropriate Management Object identifier for the SCOMO node	Section 7.4	
SCOMO-T-002-M	Support for Required nodes under root node	Section 7.4	
SCOMO-T-003-O	Support for Optional nodes	Section 7.4	
SCOMO-T-004-M	Support for Required nodes under an Optional node if the Optional node is supported	Section 7.4	

### B.2 SCR for SCOMO Client

Item	Function	Reference	Requirement
SCOMO-C-001-M	Support for Download	Section 5.2	SCOMO-C-002-O OR SCOMO-C-005-O
SCOMO-C-002-O	Support for Alternate Download	Section 5.1.2	SCOMO-C-026-O AND SCOMO-C-027-O
SCOMO-C-003-O	Support for Add of <x>/Download/<x>/Pkg URL	Section 7.4	
SCOMO-C-004-O	Support for Replace of <x>/Download/<x>/Pkg URL	Section 7.4	
SCOMO-C-005-O	Support for OMA DM Download based on Replace mechanism	Section 5.1.2	SCOMO-C-028-O AND SCOMO-C-029-O
SCOMO-C-006-O	Support for Add of <x>/Inventory/Delivered/<X>/Data	Section 7.4	
SCOMO-C-007-O	Support for Replace of <x>/Inventory/Delivered/<X>/Data	Section 7.4	
SCOMO-C-008-O	Support Inactive State for Deployment Component	Section 5.1.1.1	
SCOMO-C-009-M	Support Active State for Deployment Component	Section 5.1.1.2	
SCOMO-C-010-M	Support Removed State for Deployment Component	Section 5.1.1.3	

SCOMO-C-011-O	Support Activate Primitive for Deployment Component	Section 5.1.1.4	
SCOMO-C-012-O	Support Deactivate Primitive for Deployment Component	Section 5.1.1.4	
SCOMO-C-013-M	Support Remove Primitive for Deployment Component	Section 5.1.1.4	
SCOMO-C-014-M	Support Not Downloaded State for Delivery Package	Section 5.2.1.1	
SCOMO-C-015-O	Support Delivered State for Delivery Package	Section 5.2.1.2	
SCOMO-C-016-M	Support Installed State for Delivery Package	Section 5.2.1.3	
SCOMO-C-017-M	Support Removed State for Delivery Package	Section 5.2.1.4	
SCOMO-C-018-O	Support Download Primitive for Delivery Package	Section 5.2.1.5	
SCOMO-C-019-O	Support DownloadInstall Primitive for Delivery Package	Section 5.2.1.5	
SCOMO-C-020-O	Support DownloadInstallInactive Primitive for Delivery Package	Section 5.2.1.5	
SCOMO-C-021-O	Support Install Primitive for Delivery Package which is in the Delivered sub-tree	Section 5.2.1.5	
SCOMO-C-022-O	Support InstallInactive Primitive for Delivery Package	Section 5.2.1.5	
SCOMO-C-023-M	Support Remove Primitive for Delivery Package	Section 5.2.1.5	
SCOMO-C-024-O	Support synchronous result reporting	Section 8.5	
SCOMO-C-025-O	Support asynchronous result reporting	Section 8.6	
SCOMO-C-026-M	Support for OMA-DM Download using PkgURL node	Section 8.6	SCOMO-C-003-O OR SCOMO-C-004-O
SCOMO-C-027-O	Support for Primitives on Download sub-tree	Section 7.4	SCOMO-C-018-M OR SCOMO-C-019-O OR SCOMO-C-020-O
SCOMO-C-028-O	Support for OMA-DM Download using Data node	Section 9.1.1	SCOMO-C-006-O OR SCOMO-C-007-O

SCOMO-C-029-O	Support installation Primitives on Delivered sub-tree	Section 5.1.2	SCOMO-C-021-O OR SCOMO-C-022-O
SCOMO-C-030-O	After installation reflect the Deployment Component metadata (provided by the DM System) in corresponding nodes under the Deployed sub-tree	Section 5.1	
SCOMO-C-031-M	Support state transitions that appear to be atomic (for Deployment component). If a state transition fails the - reverse the operation and make sure the Deployment Components remain in the previous state	Section 5.1.1	
SCOMO-C-032-M	Support state transitions that appear to be atomic (for Delivery Package). If a state transition fails the - reverse the operation and make sure the Delivery Package remains in the previous state	Section 5.2.1	
SCOMO-C-033-M	Create corresponding nodes of each Deployment Component in the Deployed tree	Section 5.2.1.3	
SCOMO-C-034-M	When executing Remove Primitive of Delivery Package - delete the delivery Package from the Device and delete corresponding nodes which represented it in the Delivered tree	Section 5.2.1.4	
SCOMO-C-035-O	Support EnvType node	Section 7.4	SCOMO-C-036
SCOMO-C-036-O	Reject a component if its indicated EnvType is not supported	Section 7.4	

SCOMO-C-037-M	When using alternate download mechanism – copy the values of leaf nodes from the Download sub-tree to the corresponding leaf nodes in the Delivered tree	Section 7.4	
SCOMO-C-038-M	Support result reporting	Section 8	SCOMO-C-024 OR SCOMO-C-025
SCOMO-C-039-M	After successful installation of Deployment Components, create their nodes in the Deployed sub-tree	Section 8	

### B.3 SCR for SCOMO Server

Item	Function	Reference	Requirement
SCOMO-S-001-M	Support for the Software Component Management Object	Section 7.4	
SCOMO-S-002-O	Support triggering of DISPLAY Alert	Section 8.8.1	
SCOMO-S-003-O	Support triggering of CONFIRM OR REJECT Alert	Section 8.8.1	
SCOMO-S-004-O	Support triggering of USER INPUT Alert	Section 8.8.1	
SCOMO-S-005-O	Support triggering of USER CHOICE Alert	Section 8.8.1	

### B.4 SCR for DM System

Item	Function	Reference	Requirement
SCOMO-DMS-001-M	Provide metadata with the Deployment Component	Section 5.1	
SCOMO-DMS-002-M	Provide value for PkgID to uniquely identify the Delivery Package within the SCOMO tree	Section 7.4	
SCOMO-DMS-003-M	Provide Package Name	Section 7.4	
SCOMO-DMS-004-O	Provide value for EnvType	Section 7.4	
SCOMO-DMS-005-O	Provide value for PkgType	Section 7.4	

SCOMO-DMS-005-M	Provide value for ID to uniquely identify the Deployment Component within the SCOMO tree	Section 7.4	
-----------------	--	-------------	--