

06/28/02



J1050 U.S. PTO

7-102

Alphav

PTO/SB/16 (8/96)

Approved for use through 01/31/98. OMB 0651-0037

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Attorney Docket No. 3399P101Z

Type a plus sign (+) inside this box [ + ]

**PROVISIONAL APPLICATION FOR PATENT COVER SHEET**

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

**INVENTOR(s)/APPLICANT(s)**

| LAST NAME | FIRST NAME | MIDDLE NAME/<br>INITIAL | RESIDENCE (CITY AND<br>EITHER STATE OR<br>FOREIGN COUNTRY) |
|-----------|------------|-------------------------|------------------------------------------------------------|
| Apelqvist | Johan      | A.                      | Växjö, Sweden                                              |
| Kjellberg | Rikard     | M.                      | Stockholm, Sweden                                          |
| Skagne    | Anders     | L.                      | Växjö, Sweden                                              |
| Andersson | Tomas      | L.                      | Tävelsås, Sweden                                           |

**TITLE OF THE INVENTION (280 characters max)**

Device Manager

**CORRESPONDENCE ADDRESS (including country if not United States)**

Jordan M. Becker

BLAKELY, SOKOLOFF, TAYLOR &amp; ZAFMAN, LLP

12400 Wilshire Boulevard, Seventh Floor

Los Angeles, California 90025-1030

Telephone: (408) 720-8300

FAX: (408) 720-8383

**ENCLOSED APPLICATION PARTS (check all that apply)**

|                                                   |                    |                                                                                 |
|---------------------------------------------------|--------------------|---------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Specification | Number of Pages 22 | <input type="checkbox"/> Small Entity Statement                                 |
| <input type="checkbox"/> Drawing(s)               | Number of Sheets   | <input checked="" type="checkbox"/> Other (specify)<br>Express Mail Certificate |

**METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)**☒ A check or money order is enclosed to cover the filing fees☒ The Commissioner is hereby authorized to charge filing fees and credit Deposit Account No. 02-2666

Filing Fee Amount (\$) 160.00

This invention was made by an agency of the United States Government or under contract with an agency of the United States Government.

☒ No

Yes, the name of the U.S. Government Agency and the Government Contract Number are:

Respectfully submitted,

SIGNATURE

DATE

6/28/02

TYPED or PRINTED NAME: Jordan M. Becker

REGISTRATION NO. 39,602  
(if appropriate)

Additional inventors are being named on separately numbered sheets attached hereto

12/01/97

- 1 -

Netflix v. VideoLabs

IPR2023-00628

Netflix. Ex. 1006

PTO/SB/05 (12/97)

J1050 U.S. PTO

60/392383



06/28/02

**EXPRESS MAIL CERTIFICATE OF MAILING**

EL821850659US

"Express Mail" mailing label number: \_\_\_\_\_

Date of Deposit: June 28, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Julie Arango  
(Typed or printed name of person mailing paper or fee)

Julie Arango  
(Signature of person mailing paper or fee)

6/28/02  
(Date signed)

Serial/Patent No.: \_\_\_\_\_ Filing/Issue Date: \_\_\_\_\_  
Client: Openwave Systems Inc.  
Title: Device Manager

BSTZ File No.: 3399P101Z Atty/Secty Initials: JMB/jaa  
Date Mailed: 6-28-02 Docket Due Date: \*\*\*

The following has been received in the U.S. Patent & Trademark Office on the date stamped hereon:

- |                                                                                                |                                                                                      |                                                            |
|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------|
| <input type="checkbox"/> Amendment/Response (____ pgs.)                                        | <input checked="" type="checkbox"/> Express Mail No. <u>EL821850659US</u>            | <input checked="" type="checkbox"/> Check No. <u>51058</u> |
| <input type="checkbox"/> Appeal Brief (____ pgs.) (in triplicate)                              | <input type="checkbox"/> _____ Month(s) Extension of Time                            | Amt: <u>\$160.00</u>                                       |
| <input type="checkbox"/> Application - Utility (____ pgs., with cover and abstract)            | <input type="checkbox"/> Information Disclosure Statement & PTO-1449 (____ pgs.)     | <input type="checkbox"/> Check No. _____                   |
| <input type="checkbox"/> Application - Rule 1.53(b) Continuation (____ pgs.)                   | <input type="checkbox"/> Issue Fee Transmittal                                       | Amt: _____                                                 |
| <input type="checkbox"/> Application - Rule 1.53(b) Divisional (____ pgs.)                     | <input type="checkbox"/> Notice of Appeal                                            |                                                            |
| <input type="checkbox"/> Application - Rule 1.53(b) CIP (____ pgs.)                            | <input type="checkbox"/> Petition for Extension of Time                              |                                                            |
| <input type="checkbox"/> Application - Rule 1.53(d) CPA Transmittal (____ pgs.)                | <input type="checkbox"/> Petition for _____                                          |                                                            |
| <input type="checkbox"/> Application - Design (____ pgs.)                                      | <input checked="" type="checkbox"/> Postcard                                         |                                                            |
| <input type="checkbox"/> Application - PCT (____ pgs.)                                         | <input type="checkbox"/> Power of Attorney (____ pgs.)                               |                                                            |
| <input checked="" type="checkbox"/> Application - Provisional ( <u>22</u> pgs.)                | <input type="checkbox"/> Preliminary Amendment (____ pgs.)                           |                                                            |
| <input type="checkbox"/> Assignment and Cover Sheet                                            | <input type="checkbox"/> Reply Brief (____ pgs.)                                     |                                                            |
| <input checked="" type="checkbox"/> Certificate of Mailing                                     | <input type="checkbox"/> Response to Notice of Missing Parts                         |                                                            |
| <input type="checkbox"/> Declaration & POA (____ pgs.)                                         | <input type="checkbox"/> Small Entity Declaration for Indep. Inventor/Small Business |                                                            |
| <input type="checkbox"/> Disclosure Docs & Orig & Copy of Inventor's Signed Letter (____ pgs.) | <input checked="" type="checkbox"/> Transmittal Letter, in duplicate                 |                                                            |
| <input type="checkbox"/> Drawings: _____ # of sheets includes _____ figures                    | <input type="checkbox"/> Fee Transmittal, in duplicate                               |                                                            |

☐ Other: \_\_\_\_\_

PROVISIONAL UNITED STATES PATENT APPLICATION

Device Manager

INVENTORS:

Johan A. Apelqvist  
Rikard M. Kjellberg  
Anders L. Skagne  
Tomas L. Andersson

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
LOS ANGELES, CA 90025-1026  
(408) 720-8300

Docket No. 3399P101Z

"Express Mail" mailing label number: EL821850659US

Date of Deposit: June 28, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Julie Arango

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

6/28/02

## Openwave Systems Inc.

Inventor Name: Johan Apelqvist

Group Name:

# IDEA PROPOSAL

### *Problem*

A system uses a resource (device capabilities). This resource is periodically queried for information that may or may not exist. If the information exists, it shall be returned to the caller. If the information does not exist, the resource must somehow create the information and return it to the caller. Many instances of the system may exist in many different geographical locations. The resource will therefore also be multiplied. If one system instance requires some information it can be expected that another instance requires it as well. How can we automatically update all resources and keep them synchronized without creating a complex synchronization infrastructure and/or unnecessary processing by having each resource creating new information independent of others?

### *Known approaches*

There are a couple of ways to solve this problem:

1. Let each resource maintain its own information repository independent of any other resources and assume sole responsibility for keeping it current and updated.
2. Let each local resource query a central repository (database) through some common protocol like HTTP and always get the information that way. The system can improve performance by implementing a cache that will minimize the network communication from the resource. Implement some synchronization mechanism, such as a watchdog that periodically polls the central to get the latest updates.

The particular implementation in the system provides an algorithm for how to perform an update of the information resource. It is therefore not as simple as just providing a centrally located database with some common interface, such as HTTP. If the centrally located information resource is unable to find the information, it should apply the algorithm to update the information.

### *Comments*

The solution offers the following benefits to the system:

- **Performance control.** A centrally located repository and information gathering algorithm provides a controllable environment for managing performance. Although a distributed model normally would scale better, the algorithm implies network queries, which, in a distributed environment, would amount to  $[n * m]$  network transactions rather than just  $[n]$  transactions (where  $n$  is the number of queries made for missing information and  $m$  the number of resulting queries required to assemble the information).
- **Minimize dependencies.** Since information gathering depends on additional external resources, every such external request can potentially fail. In this solution, even if the initial request fails, it can start a process where the administrator is alerted that the external resource is unavailable. It can then periodically query the external resource and then quickly assemble the information as soon as it becomes available. This actually speeds up the remedy process.
- **Synchronization.** The system implements a lazy synchronization scheme, where each distributed resource is updated if/when the system requests information. The central repository initiates one information gathering activity the first time a new information is requested and provides that information every to all other clients every time they request it thereafter.
- **Stateless management of stateful function.** Since each distributed resource will query the central repository if it is unable to find the information in its local cache, it will not need to persist its state. If the resource fails and is restarted, it will recreate its state over time, using lazy synchronization as described above.

1. The device manager lookup schema could look something like this for an unidentified device:

Check CC/R for a suitable profile;

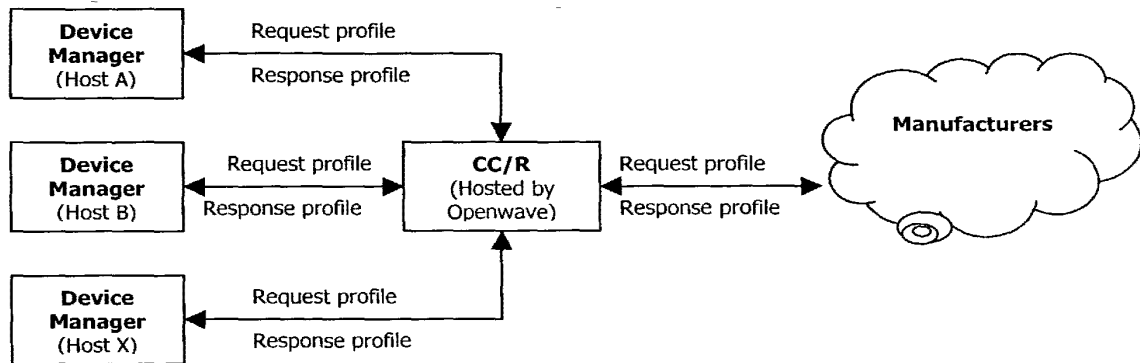
- If found store that as a cached profile locally.
- If not found the CC/R will try to resolve the profile from the manufacturers server (if CC/PP) or if a legacy device, return "Unknown device" that will be put in a quarantine (for the admin to examine manually later). If the CC/R can not connect to the manufacturer server and resolve the profile (connection problems etc), the local device manager will try to connect to the manufacturer server to resolve the profile.

2. The schema could look like this for an already identified device:

Check the cached profile's expire flag for validity;

- If valid, use the found profile
- If not valid perform the schema in (1) above.

Below is a graphical view of the relationships between the different components in the centralized capability repository model.



Openwave Systems Inc.

## INVENTION DISCLOSURE FORM

### TO BE PREPARED BY TECH GROUP ARCHITECT

**1. Proposed title of invention:**

Centralized Capability Repository (CC/R)

**2. Please describe the general technical area of which the invention is a part (e.g., server-client architectures, man-machine-interfaces, signal processing methods, methods for routing data, etc.).**

Server-client architecture.

**3. Please describe your motivation for the idea underlying the invention, i.e., the problem your invention is designed to solve.**

To easy capability management, a specialized device manager may be placed on a central server hosted by Openwave/Ellipsus that has a repository (CC/R) of all devices discovered and verified on the market. The distributed device managers (those incorporated in the sold product) may be configured to look for device profiles on the central server before trying to connect to the URIs provided by the phone. This way we can ensure correctness and availability for distributed device managers. By using the expire flag described above, we can timeout cached profiles and then allow the device manager to reload the profile from the central storage.

**4. Please describe the primary solutions to that problem that have been proposed by others, and the disadvantages of those solutions (in particular, those disadvantages which your invention does not have or overcomes). If you are aware of any articles, technical papers, or patents that describe these other solutions, please attach copies to this form.**

There is only one other solution that I am aware of, and that is the DELI architecture, but the DELI solution will suffer from not validating the RDF document, thus using fixed vocabularies (will not apply to the real world, since most device manufacturers will submit their own parameters).

Papers on DELI can be found at: <http://www-uk.hpl.hp.com/people/marbut/>

**5. Please attach a detailed description of your invention. If possible, please include (where applicable) a top-level block diagram showing the primary functional units of the invention and the associated inputs and outputs for each block, flowcharts showing the primary data processing routines, circuit schematics, and a written description of how the functional blocks or components operate to produce the inventive result. If the invention has not yet been implemented, please provide sufficient information to enable a senior engineer to determine that it could in fact be implemented without encountering obstacles requiring significant engineering and/or other innovations to overcome.**

See attached document.

The solution offers the following benefits to the system:

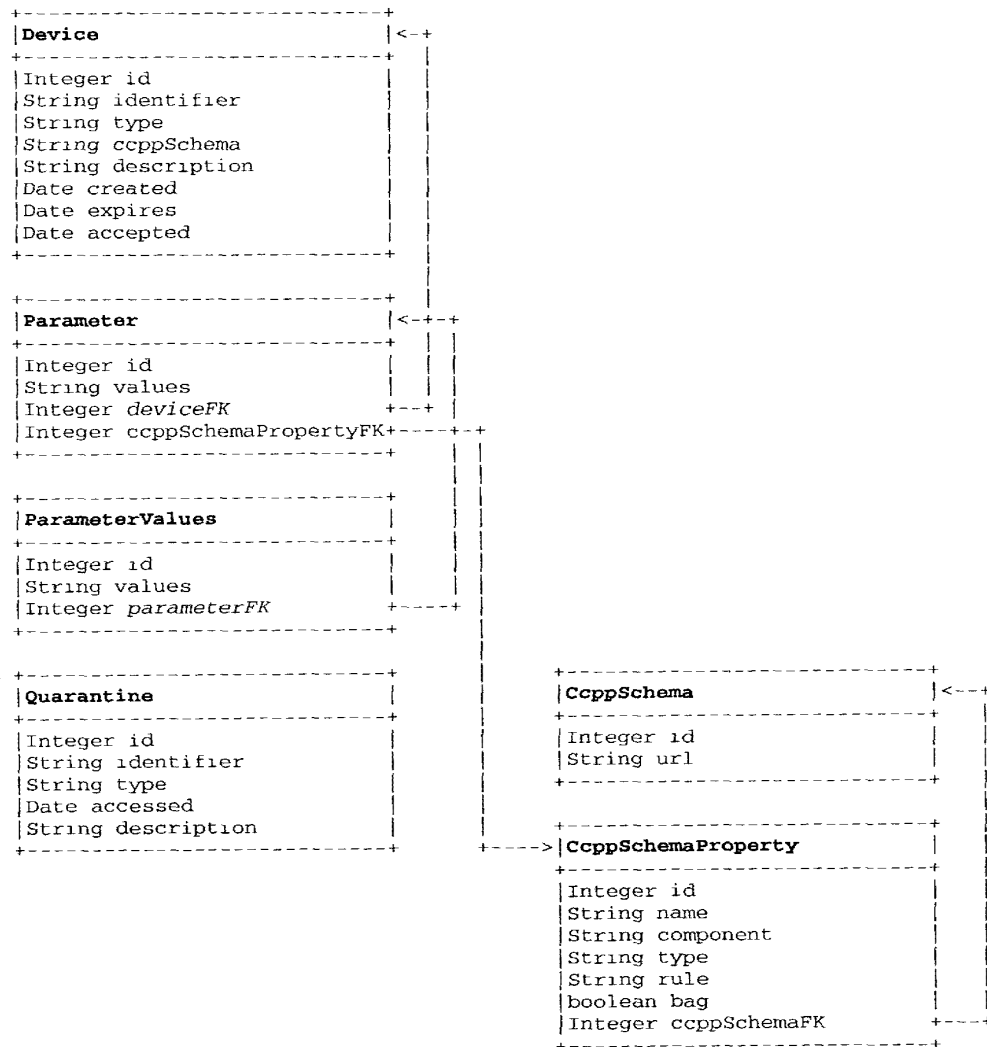
- **Performance control.** A centrally located repository and information gathering algorithm provides a controllable environment for managing performance. Although a distributed model normally would scale better, the algorithm implies network queries, which, in a distributed environment, would amount to  $[n * m]$  network transactions rather than just  $[n]$  transactions (where  $n$  is the number of queries made for missing information and  $m$  the number of resulting queries required to assemble the information).
- **Minimize dependencies.** Since information gathering depends on additional external resources, every such external request can potentially fail. In this solution, even if the initial request fails, it can start a process where the administrator is alerted that the external resource is unavailable. It can then periodically query the external resource and then quickly assemble the information as soon as it becomes available. This actually speeds up the remedy process.
- **Synchronization.** The system implements a lazy synchronization scheme, where each distributed resource is updated if/when the system requests information. The central repository initiates one information gathering activity the first time a new information is requested and provides that information every to all other clients every time they request it thereafter.

- **Stateless management of stateful function.** Since each distributed resource will query the central repository if it is unable to find the information in its local cache, it will not need to persist its state. If the resource fails and is restarted, it will recreate its state over time, using lazy synchronization as described above.

## Device Manager 2.2

This is a proposal for a architecture better suited for the current iMAP system. It requires a complete rewrite to meet the requirements stated in this document. The device manager will use the adapter framework for device recognition. This way we can have an exported API for device recognition that developers can use to create new device adapters. There will also be possible to arrange the adapters in order to more effective identify devices accessing the system. The system uses an more exact RDF parser that will allow even extended UAProf parameters. Validation of RDF document against the CC/PP schema will be included in the parser. Caching of accessed devices is done in the device manager and sub-caching of CC/PP schemas can be done in the adapters. See the section "Centralized capability repository" for an extension of the device manager for easy management.

## Database schema



## Device

### \* identifier

The request header or other identification for this device (x-wap-profile or user-agent most frequent, but the request path is used in some cases).

### \* type

What type of device is detected ("CC/PP" | "User-Agent" or whatever. Depends on the

adapter discovered the device).

- \* **created**  
The time of database creation. Done after the discover adapter has returned the device.
- \* **expires**  
May be set or ignored (treated as no expire if ignored). This will cause the device to be removed from the system if a device accesses the profile after the time has expired.  
The profile will be reloaded from the source.
- \* **accepted**  
Is set when the admin has manually accepted the device as a legal device. The device profile will be used meanwhile until the admin accepts or rejects the device. This can be used to create a view for the admin to show last incoming devices not known to the system.

#### Parameter

##### \* values

The values of the parameter as a comma-separated string.

#### ParameterValues

##### \* values

The values of the parameter as a comma-separated string. The values here override the values stated in the parameter table (*non CC/PP behaviour, iMAP specific*).

#### Quarantine

##### \* identifier

The request header or other identification for this device (x-wap-profile or user-agent most frequent, but the request path is used in some cases).

##### \* type

What type of device is detected ("CC/PP" | "User-Agent" or whatever. Depends on the adapter discovered the device).

##### \* accessed

Date of device request

#### CcppSchema

##### \* url

URL to the Ccpp schema.

#### CcppSchemaProperty

##### \* name

The name of the parameter (based on the UAProf specification. Could be extended with device specific parameters).

##### \* component

The name of the component where the parameter was found ("HardwarePlatform", "SoftwarePlatform" etc.).

##### \* type

Tells the value type of the parameter values ("Literal", "Number", etc.).

##### \* rule

The resolution rule used to override/append parameters ("Override", "Append", "Locked").

##### \* bag

Gives an indication if there may be more than one value in the parameter.

#### Features

- \* Exact UAProf parsing, validating against the schema included in the RDF document.
- \* Validating against a default vocabulary (URL-pointer) for devices with expired or wrong schema.
- \* Device discovery based on the adapter framework.
- \* Possibility to ban devices by using a ban-adapter.
- \* Caching of accessed devices in the manager and sub-caching in the adapters.

## Adapters

- \* **UAPProf** adapter  
For discovery of UAPProf enabled devices. Configuration of this adapter is not needed.
- \* **UserAgent** adapter  
For discovery of non UAPProf enabled devices. Configuration is done by giving the adapter a list of regular expressions for user.agent matching and a URLs defining the parameters in RDF format. Parameters are set in the map as Expression/URL pairs.
- \* **SiemensSL45i** adapter (*Specific UserAgent adapter*)  
For discovery of the Siemens SL45i. Configuration is done by giving one regular expression and one URL.
- \* **MotorolaAccompli008** adapter  
For discovery of the Motorola Accompli 008
- \* **NokiaWapToolkit** adapter (*Specific UserAgent adapter*)  
For discovery of the Nokia WAP Toolkit. Configuration is done by giving one regular expression and one URL.
- \* **Netscape** adapter (*Specific UserAgent adapter*)  
For discovery of the Netscape browser. Configuration is done by giving one regular expression and one URL.
- \* **Msie** adapter (*Specific UserAgent adapter*)  
For discovery of the Internet Explorer browser. Configuration is done by giving one regular expression and one URL.
- \* **Oprah** adapter (*Specific UserAgent adapter*)  
For discovery of the Oprah browser. Configuration is done by giving one regular expression and one URL.
- \* **OpenwaveWapToolkit** adapter (*Specific UserAgent adapter*)  
For discovery of the Openwave WAP Toolkit. Configuration is done by giving one regular expression and one URL.
- \* **KBrowser** adapter (*Specific UserAgent adapter*)  
For discovery of the KBrowser. Configuration is done by giving one regular expression and one URL.
- \* **Ban** adapter  
For banning devices from entering the system (i.e. reject bots etc.). Configuration is done by giving the adapter a list of regular expressions for header matching and names defining the headers to match against. Parameters are set in the map as Expression/Header-Name pairs (the header-name could be some other adapter specific name like "request-path" for the adapter to match against the request path).
- \* **CC/R** adapter  
To let the device manager make use of the Centralized Capability Repository (CC/R) for profile resolution.

## Admin functionality

- \* Save device
- \* Remove device
- \* List all devices
- \* List devices based on some search criteria (list all "User-Agent" or "CC/PP" devices etc.).
- \* Retrieve identified device
- \* Save parameter
- \* Update parameter (override)
- \* Remove parameter (reset overridden parameter)
- \* List all parameter for a device
- \* Retrieve identified parameter
- \* Retrieve CC/PP schema properties

## User functionality

- \* Retrieve a device based on the device request parameters

## API

```

IDeviceManager
{
    /**
     * @param identifier is the header (or else) value identifying the device
     * @param params is a map of <names> against <IParameters>
     */
    IDevice saveDevice(String identifier, Map params)
}

```

```

throws ImapException;

/**
 * Allows a client (e.g. the UI) to manually add a new device to the
 * system. By filling out the map with significant values, the device
 * manager will be able to construct a device by asking the adapters.
 * Example:
 *   {{ "x-wap-profile", "http://host/file.ext, 1-jHrGMjJ"},
 *     { "x-wap-profile-diff", ["1-<?xml.>"] }}
 * or
 *   { "User-Agent", " Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)" }
 * @param headers is a map of name/value(s) pair that defines the device
 */
IDevice saveDevice(Map headers)
throws ImapException;

/**
 * Removes the device and forces the device parameters to fetch again.
 */
IDevice updateDevice(Integer id)
throws ImapException;

void removeDevice(Integer id)
throws ImapException;

/**
 * Retrieves all devices except the quarantined ones.
 */
IDevice[] getDevices()
throws ImapException;

IDevice getDevice(Integer id)
throws ImapException;

/**
 * DeviceContext("accepted")
 * DeviceContext("latest")
 * DeviceContext("type", { "User-Agent" | "CC/PP" | ... })
 * DeviceContext("parameter", <name>, <value(s)>)
 * DeviceContext("quarantined")
 */
IDevice[] getDevices(ISearchContext ctx)
throws ImapException;

IParameter saveParameter(String name, String values, String component, String type,
                        String rule, boolean bag)
throws ImapException;

/**
 * Overrides the parameter (non CC/PP behaviour, iMAP specific)
 */
IParameter updateParameter(Integer id, String values)
throws ImapException;

/**
 * Resets the overridden parameter (non CC/PP behaviour, iMAP specific)
 */
void removeParameter(Integer id)
throws ImapException;

IParameter[] getParameters(Integer device)
throws ImapException;

IParameter getParameter(Integer id)
throws ImapException;

ICcppSchemaProperty[] getCcppScema(String ccppSchema)

/**
 * Retrieves a device by asking the adapters to identify the device
 * via the request parameters. Devices not found in the system will
 * automatically be stored and unidentified devices will be stored in
 * a quarantine table for the admin to view and manually handle.
 */
IDevice getDevice(HttpServletRequest req)
throws ImapException;

```

}

where *IDevice* and *IParameter* is interfaced value objects showing only getter methods.

*IDevice*

```
{
    Integer getId();
    String getIdentifier();
    Map getParameters();
    String getCcppSchema();
    String getName();
}
```

*IParameter* extends *ICcppSchemaProperty*

```
{
    Integer getId();
    String[] getValues();
    String getValueString();
}
```

*ICcppSchemaProperty*

```
{
    String getName();
    String getComponent();
    String getRule();
    String getType();
    boolean isBag();
}
```

Device discovery adapters are using the following interface:

*IDeviceAdapter*

```
{
    IProfile getProfile(HttpServletRequest req)
        throws Exception;

    String getIdentifier(HttpServletRequest req)
        throws Exception;
}
```

*IProfile*

```
{
    String getIdentifier(); // {<x-wap-profile> | <User-Agent> | ...}
    String getType(); // {"User-Agent" | "CC/PP" | ...}
    Map getParameters(); // map(name, IParameter)
    String[] getURIs(); // {URLs | x-wap-profile-diffs}
    Map getParameters(String uri);
    String getCcppSchema();
    String getDescription();
}
```

### Centralized capability repository

To easy capability management for the iMAP, a specialized device manager may be placed on a central server hosted by Openwave/Ellipsus that has a repository (CC/R) of all devices discovered and verified on the market. The distributed device managers (those incorporated in the sold product) may be configured to look for device profiles on the central server before trying to connect to the URIs provided by the phone. This way we can ensure correctness and availability for distributed device managers. By using the expire flag described above, we can timeout cached profiles and then allow the device manager to reload the profile from the central storage.

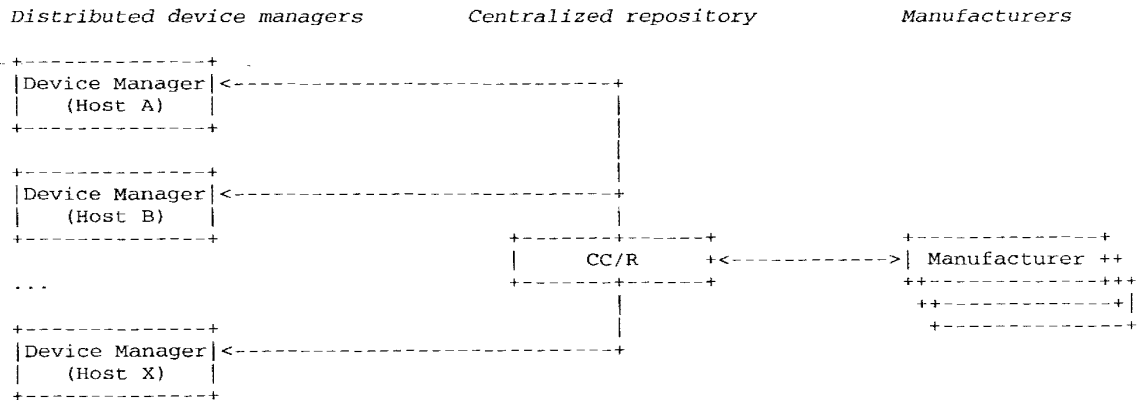
1. The device manager lookup schema could look something like this for an unidentified device:

```
    Check CC/R for a suitable profile;
    * if found store that as a cached profile locally.
    * if not found the CC/R will try to resolve the profile from the manufacturers
server
    (if CC/PP) or if a legacy device, return "Unknown device" that will be put in the
    quarantine. If the CC/R can not connect to the manufacturer server and resolve
the
    profile (connection problems etc), the local device manager will try to connect
to
    the manufacturer server to resolve the profile.
```

2. The schema could look like this for an already identified device:

- Check the cached profile's expire flag for validity;
- \* if valid, use the found profile
- \* if not valid perform the schema in (1) above.

Below is a graphical view of the relationships between the different components in the centralized capability repository model.



---

# Ellipsus Systems

---

## **infiniteMAP Feature Specification Device Capability Manager**

**Version 1.1**

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

## Revision History

| Date | Version | Description                                                                                                                              | Author           |
|------|---------|------------------------------------------------------------------------------------------------------------------------------------------|------------------|
|      | 0.1     | Draft                                                                                                                                    | Rikard Kjellberg |
|      | 1.0     | Added new feature set Resolving a profile and changed feature set Receiving a UAProf/CC-PP Header to Supporting CC/PP exchange protocol. | Anders Skagne    |
|      | 1.0     | Changed description of the first feature in feature set Supporting CC/PP ....                                                            | Anders Skagne    |
|      | 1.0     | Removed feature set: Checking a Product Implementation                                                                                   | Tomas Andersson  |
|      | 1.0     | Removed feature set: Checking a Product Implementation Set                                                                               | Tomas Andersson  |
|      | 1.0     | Removed feature set: Resolving a Profile                                                                                                 | Tomas Andersson  |
|      | 1.0     | Added feature set: Match a set of capability requirement to a set of device capabilities                                                 | Tomas Andersson  |
|      | 1.1     | Removing feature set: Configuration of the capabilities and Match a set of capability requirements to a set of device capabilities.      | Johan Apelqvist  |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

## Table of Contents

|                                                     |    |
|-----------------------------------------------------|----|
| 1. Introduction                                     | 4  |
| 1.1 Purpose                                         | 4  |
| 1.2 Scope                                           | 4  |
| 1.3 Definitions, Acronyms and Abbreviations         | 4  |
| 1.3.1 Definitions                                   | 4  |
| 1.3.2 Acronyms                                      | 4  |
| 1.3.3 Abbreviations                                 | 4  |
| 1.4 References                                      | 5  |
| 1.5 Overview                                        | 5  |
| 2. System Description                               | 6  |
| 2.1 Overview of the device capability system        | 6  |
| 2.1.1 Overview of the device capability manager     | 6  |
| 2.1.2 Overview of the device manager GUI            | 7  |
| 2.2 Use case view for the device capability manager | 7  |
| 3. Features                                         | 9  |
| 3.1 Requirements                                    | 9  |
| 3.2 Feature Set: Recognize a Device                 | 11 |
| 3.3 Feature Set: Auto-configuring a Device          | 11 |
| 3.4 Feature Set: Configure an Unknown Device        | 11 |
| 3.5 Feature Set: Configure a Quarantined User-Agent | 11 |
| 4. Architecturally Significant Design Packages      | 11 |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

## Detailed Design Specification

### 1. Introduction

#### 1.1 Purpose

This document describes the detailed features for the Device Capability Manager, which is called a *major feature set* (MFS). It uses the requirements from the functional requirements specifications as *feature sets* (FS) and details them into features.

#### 1.2 Scope

#### 1.3 Definitions, Acronyms and Abbreviations

The definitions, acronyms and abbreviations from [A.1] apply.

##### 1.3.1 Definitions

| Term | Description |
|------|-------------|
|      |             |

##### 1.3.2 Acronyms

| Acronym | Description                                 |
|---------|---------------------------------------------|
| MFS     | Major Feature Set                           |
| FS      | Feature Set                                 |
| CC/PP   | Composite Capabilities / Preference Profile |
| UAPROF  | User Agent Profile                          |
| CPI     | Capability and Preference Information       |
| RDF     | Resource Description Format                 |
| CRUD    | Change, Read, Update and Delete             |

##### 1.3.3 Abbreviations

| Abbreviation | Description |
|--------------|-------------|
|              |             |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

#### 1.4 References

| Id    | Description (title, version, date, publisher, source)                  |
|-------|------------------------------------------------------------------------|
| [R.1] | Functional Requirement Specification, MAP-002-RS-01, Ellipsus Systems. |
| [R.2] | WAP-248-UAPROF-20011020-a                                              |
|       |                                                                        |

#### 1.5 Overview

This SD is a detailed design document below [R.1], which is the overall functional requirement specification of the MAP.

Each feature is assigned a priority (pri). This priority can take the following values:

| Value | Priority       | Meaning                                                                                                                                                                           |
|-------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L     | Low            | This feature can be deferred to later in the project or even to a later release.                                                                                                  |
| M     | Medium         | This feature can be deferred to later in the project but should be implemented prior to first release.                                                                            |
| H     | High           | This feature should be implemented as soon as possible to verify that it can be realized, or to be able to implement other aspects of the system, which rely on this requirement. |
| N/A   | Not Applicable | This feature is not subject to prioritization since it is of general character and affects the overall development project.                                                       |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

## 2. System Description

### 2.1 Overview of the device capability system

The device capability systems is responsible for dealing with capabilities and support other modules in their decision of how to present content, and which products should be discovered. To be able to do so, the system have a manager which exposes the operations for managing existing capabilities, detecting current connected devices and support filtering on device requirements.

#### 2.1.1 Overview of the device capability manager

The device capability manager (DCM) is responsible for managing device capabilities and providing a way to match a connected device to a set of capabilities. The DCM assumes that the primary channel for communication with a system, is the hyper text transfer protocol (HTTP) and will therefore make use of all available information in the HTTP headers to find a set of capabilities matching the information in the headers. The DCM is capability centric, since many devices are only identified via the applications they use when communicating with external systems. The DCM works with a capability tree, where capabilities can be organized with great freedom. The tree allows the DCM to associate device applications with a set of capabilities whenever possible.

The DCM is also capable of searching for a match between a device identifier and a set of capability requirements, which are associated with a product identifier or similar. This function provides the DCM client with a real-time mechanism to discriminate capability requirements sets, which aren't a subset of the devices capabilities. The filtering is done using comparison rules depending on the type of capability.

- Use UAProf in the CC/PP framework as defined by the World Wide Web Consortium (<http://www.w3.org/Mobile/CCPP/>).
- Supports HTTP uaprof header match between device and capabilities, as defined in [R.2]
- Supports HTTP user-agent header match between device and capabilities.
- Supports HTTP accept header match between device and capabilities.
- Uses a best-fit approach to match previously unknown device types with capabilities.
- User interface for device capability management.
- Provides quarantine for previously unknown device types (i.e. devices without UAProf header or with unknown user-agent header), so that the administrator can be alerted to new device types and perform complete configuration of these.
- Persistent cache for all device capabilities, including UAProf.
- Auto-configures capabilities provided via UAProf.
- Capability-centric device management.

#### 2.1.1.1 Composite Capabilities / Preference Profiles and User Agent Profile brief description

Work is ongoing in the World-Wide Web Consortium (W3C) to define mechanisms for describing and transmitting information about the capabilities of Web clients and the display preferences of Web users. The Composite Capabilities/Preferences Profile (CC/PP) specification defines a high-level structured framework for describing this information using the Resource Description Framework (RDF). CC/PP profiles are structured into named "components," each containing a collection of attribute-value pairs, or properties. Each component may optionally provide a default description block containing either a set of default values for attributes of that component or a URI that refers to a document containing those default values. Any attributes explicitly provided in the component description therefore override the default values provided in the default description block or through that URI. The CC/PP specification does not mandate a particular set of components or attributes,

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

choosing instead to defer that definition to other standards bodies. The mechanism by which the profile is transported between the mobile terminal, WAP Gateway and origin server is defined in this specification

The User Agent Profile (UAProf) specification extends WAP 2.0 to enable the end-to-end flow of a User Agent Profile (UAProf), also referred to as **Capability and Preference Information (CPI)**, between the WAP client, the intermediate network points, and the origin server. It seeks to interoperate seamlessly with the emerging standards for Composite Capability/Preference Profile (CC/PP) distribution over the Internet. It uses the CC/PP model to define a robust, extensible framework for describing and transmitting CPI about the client, user, and network that will be processing the content contained in a WSP/HTTP response. The specification defines a set of components and attributes that WAP-enabled devices may convey within the CPI. This CPI may include, but is not limited to, hardware characteristics (screen size, color capabilities, image capabilities, manufacturer, etc.), software characteristics (operating system vendor and version, support for MExE, list of audio and video encoders, etc.), application/user preferences (browser manufacturer and version, markup languages and versions supported, scripting languages supported, etc.), WAP characteristics (WML script libraries, WAP version, WML deck size, etc.), and network characteristics (bearer characteristics such as latency and reliability, etc.). This specification seeks to minimize wireless bandwidth consumption by using a binary encoding for the CPI and by supporting efficient transmission and caching over WSP in a manner that allows easy interoperability with HTTP.

As a request travels over the network from the client device to the origin server, each network element may optionally add additional profile information to the transmitted CPI. These additions may provide information available solely to that particular network element. Alternatively, this information may override the capabilities exposed by the client, particularly in cases where that network element is capable of performing in-band content transformations to meet the capability requirements of the requesting client device.

Origin servers, gateways, and proxies can use the CPI to ensure that the user receives content that is particularly tailored for the environment in which it will be presented. Moreover, this specification permits the origin server to select and deliver services that are appropriate to the capabilities of the requesting client. Finally, it is expected that this specification will be used to enhance content personalization based on user preferences, and other factors, as specified by the Platform for Privacy Preferences Project (P3P).

*You can read more about the UAPROF and it's attributes in[R.2].*

#### 2.1.1.2 Module decoupling

The DCM will consist of two major parts, the web layer and the server layer. The web interface is a way to provide an administration view of the capabilities and the grouping of the same. The server interface is used by the web layer to fulfill the administration of the DCM. The server layer also provides method to determine the capabilities for a device during run-time, through use of the HTTP request headers.

Due to it's independencies the DCM server layer design must aim for a loose coupling to the rest of the system., in other word the DCM server layer should be designed so it can reside stand-alone in any J2EE compliant environment.

### 2.1.2 Overview of the device manager GUI

The device manager GUI should be able to present the current capability grouping in the system, and enable graphical views of CRUD operations, on several components (capability, device, device group etc).

## 2.2 Use case view for the device capability manager

- o **Recognize a device**  
The DCM client ask the device manager to recognize capabilities for a connected device based on the HTTP request headers
- o **CRUD on manager components**  
The administrator of the system can perform CRUD operations on the components which are part of the device manager system

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Feature Specification                     | Date:        |
| MAP-002-FS-01 – Device Capability Manager |              |

- o **Auto-configure device**  
When the device isn't present in the system and a UAPROF header is available the DCM should go and fetch the content describing the UAPROF and then store it internal.
- o **Configure an unknown device**  
When the device isn't present in the system the DCM should try to make a best fit based on the ACCEPT header
- o **Configure a quarantined user-agent**  
Administrator should be able to bind a quarantined user-agent to any of the existing capabilities set in system.
- o **Match a capability set to a set of device requirements**  
The DCM client ask the device manager to determine whether a set of requirements is supported by a device or not.

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Detailed Design Specification             | Date:        |
| MAP-002-SD-04 – Device Capability Manager |              |

### 3. Features

#### 3.1 Requirements

The following requirements have been extracted from [R.1] and form the *feature sets* for this module. One requirement can result in several Feature Sets. Several requirements can also fit in one single feature set.

| Req # | Requirement              | Pri | Description                                                                                                                                        | Notes                                                                                                                                                                                                                                                                | Owner | Feature Set |
|-------|--------------------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------|
| 14    | Device recognition       | H   | It shall be possible to accommodate any device, provided it conforms to the requirements in this section                                           | This implies that the device type can be unknown as long as the provisioning requirements are met                                                                                                                                                                    |       | 3.2         |
| 15    | Device capabilities      | H   | The MAP shall detect device type and capabilities when a user connects to the <i>consumer view</i> .                                               |                                                                                                                                                                                                                                                                      |       | 3.2         |
| 16    | Device functionality     | L   | The MAP shall be able to handle any functionality announced by the device, provided it conforms to the provisioning requirements in this document. |                                                                                                                                                                                                                                                                      |       | 3.2, 3.3    |
| 75    | Best fit                 | H   | The menu shall be based on a "best fit" where the products are matched against execution capabilities.                                             | <i>Note-</i> . There might be two version of an application, one provisions a URL, the other an applet. If the device is capable of running an applet as well as a URL, the applet is the preferred product to download and consequently also to present in the menu |       | 3.7         |
| 77    | Auto-configuration       | M   | It shall be possible to connect to the MAP and auto-configure device and device applications                                                       |                                                                                                                                                                                                                                                                      |       | 3.3<br>3.5  |
| 78    | Transport capabilities   | H   | It shall be possible to enter a number of transport capabilities into a database                                                                   | <i>Note-</i> . Examples of transport capabilities are WAP, TCP/IP/HTTP, SMS, enhanced SMS, OTA a s o.                                                                                                                                                                |       |             |
| 79    | Execution capabilities   | H   | It shall be possible to enter a number of execution capabilities into a database.                                                                  | <i>Note-</i> . Examples are J2ME/MIDP, J2SE, applets, web browser, wap browser, SMS, native OS platform                                                                                                                                                              |       |             |
| 80    | Device type              | H   | It shall be possible to configure device make and model                                                                                            |                                                                                                                                                                                                                                                                      |       |             |
| 81    | Physical characteristics | H   | It shall be possible to configure physical characteristics of the device                                                                           | <i>Note-</i> . This could be screen size, color enabled/disabled, input type (keyboard, keypad, graffitu, pencil, voice), processing power, typical memory size a s o                                                                                                |       |             |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Detailed Design Specification             | Date:        |
| MAP-002-SD-04 – Device Capability Manager |              |

|    |                                  |   |                                                                                                                                                                       |                                                                                                                                                                                                                                |       |          |
|----|----------------------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------|
| 83 | Capability                       | H | It shall be possible to configure a device type to support several capabilities in parallel (see capabilities in 3.3.1.1)                                             | <i>Note-</i> This could mean that a device supports J2ME/MIDP while also being able to send SMS, browse via WAP as o                                                                                                           |       |          |
| 85 | Authentication                   | H | The authentication mechanism supported by the device shall be specified                                                                                               | <i>Note-</i> An example is support for http status code 401.                                                                                                                                                                   |       |          |
| 86 | Content type                     | H | The markup language used to communicate with the application shall be specified                                                                                       |                                                                                                                                                                                                                                |       |          |
| 87 | Descriptor format                | H | It shall be possible to specify which descriptor format the application uses to describe the downloaded product (download applications only)                          |                                                                                                                                                                                                                                |       |          |
|    | UAPROF/CC-PP                     | H |                                                                                                                                                                       | Use CC/PP as framework to support UAPROF capabilities<br>Make it pluggable enough to support other standard document formats. Perhaps Ellipsus should define it's own capabilities in a schema to be used together with UAPROF | johap | 3.2, 3.3 |
|    | Capability categories            |   | Each capability shall be categorized as discovery (content types), Static physical (keypad etc), Dynamic physical (RAM size etc), Runtime (program execution related) |                                                                                                                                                                                                                                |       |          |
|    | Device type insertion            |   | The device shall automatically be inserted into the device capability tree based on its capabilities.                                                                 |                                                                                                                                                                                                                                |       | 3.3      |
|    | Best fit presentation capability |   | The DCM should be able to determine the best fitted content type for an unknown device                                                                                |                                                                                                                                                                                                                                |       | 3.4      |

|                                           |              |
|-------------------------------------------|--------------|
| InfiniteMAP                               | Version: 0.1 |
| Detailed Design Specification             | Date:        |
| MAP-002-SD-04 – Device Capability Manager |              |

### 3.2 Feature Set: Recognize a Device

The manager SHALL be able to determine any devices capabilities, provided that the device can communicate information in a format that is digestible by the manager, such as HTTP headers like ACCEPT, USER-AGENT and UAPROF. **If UAPROF is present in the header it SHOULD always be used in favor of other headers.** Should the device type be available in the http request header information, it may be used when matching it with capabilities.

When a user connects to the system, the receiver of the request MUST forward the http request header information to the device manager

The manager SHALL promptly respond with the appropriate identifier. It MAY also indicate device, if it is possible to detect that.

| # | Feature                                                           | Notes | Pri | Owner |
|---|-------------------------------------------------------------------|-------|-----|-------|
| 1 | Get the capability set identifier based on the UAPROF HTTP header |       |     |       |
| 2 | Get the capability set identifier based on the USER-AGENT header  |       |     |       |
| 3 | Get the capability set identifier based on the ACCEPT header      |       |     |       |

### 3.3 Feature Set: Auto-configuring a Device

In those cases where an unknown device is detected, and a UAPROF header is available, the system shall allow it to perform an auto-configuration procedure. The header(s) is provided as an URI which points at a RDF-based document describing the device's capabilities. To increase performance the document is parsed and then stored internally, and the next time the same URI is encountered the DCM doesn't have to retrieve the information again.

| # | Feature                                                                                                  | Notes | Pri | Owner |
|---|----------------------------------------------------------------------------------------------------------|-------|-----|-------|
| 1 | Get the capability set identifier based on the UAPROF HTTP header pointing at a previously unvisited URI |       |     |       |

### 3.4 Feature Set: Configure an Unknown Device

The manager will always perform a best fit and thereby at least be able to inform the user of the manager of some capability that the device supports.

The header information retrieved from the unknown device is put in a quarantine storage, so that the administrator can set up the device capabilities properly later.

| # | Feature                                                            | Notes | Pri | Owner |
|---|--------------------------------------------------------------------|-------|-----|-------|
| 1 | Get the best fitted capability set for an non-deterministic device |       |     |       |

### 3.5 Feature Set: Configure a Quarantined User-Agent

A user shall be able to connect to the manager and list quarantined devices.

It shall be possible to further configure capabilities manually and position the device in the capability tree.

| # | Feature                                                | Notes | Pri | Owner |
|---|--------------------------------------------------------|-------|-----|-------|
| 1 | Associate a quarantine device to a set of capabilities |       |     |       |

## 4. Architecturally Significant Design Packages

The device manager related classes are located in the com.ellipsus.imap.admin.device package.