

Wireless Internet Platform for Interoperability 2.0.1

Supplement

September 2004

**Korea Wireless Internet
Standardization Forum**

1.	Criteria for the Mixed Use of API Between MSF/MSP and CLDC/MIDP	3
2.	Security Policy for MSF/MSP and CLDC/MIDP	5
3.	Return Value for Basic Error	6
4.	Usage Example	7
4.1.	Usage Example of DLL	7
4.1.1.	C API	7
4.1.2.	Java API	10
4.2.	Example of Terminal Performance Measurement	11
5.	wCard	14
5.1.	Related Specifications	14
5.2.	Overview of vCard 3.0 Specifications	14
5.3.	vCard 3.0 Specifications: Examples	15
5.4.	wCard Specifications	17
5.4.1.	Add <Types>:	17
5.4.2.	Add <Type Parameter>:	19
5.4.3.	Usage of <Value>	19
5.4.4.	Other Settings	21
5.5.	Example of wCard Specifications	23
5.6.	Examples of wCard	24

1. Criteria for the Mixed Use of API Between MSF/MSP and CLDC/MIDP

In WIPI 2.0, there are APIs with identical functions in MSF/MSP and CLDC/MIDP. Since the mixed use of APIs poses problems depending on the implemented platform, the platform should ensure interoperability in compliance with the following:

To define the criteria for the use of CLDC/MIDP API in WIPI 2.0, related packages are classified into three types.

For APIs with identical package names, the platform should provide identical functions.

Table 1. Identical packages.

Package Group	MSF/MSP Package	CLDC/MIDP Package
IO	java.io	
Language	java.lang	
Utility	java.util	

Although possessing identical functions, certain APIs should be used independently since the content of implementation is different. In this case, the platform should ensure the interoperability of MSF/MSP and CLDC/MIDP packages in identical package groups (including lower-level packages) without being mixed up when they are used independently.

For example, when using `org.kwis.msp.lcdiui`, which means selecting a graphics package of MSF/MSP, `javax.microedition.midlet` and `javax.microedition.lcdiui` cannot be used. Nonetheless, other package groups are still available for specification and use. In such cases, the platform should ensure interoperability.

In other words, `org.kwis.msp.lcdiui`, `javax.microedition.rms`, `org.kwis.msp.media`, and `javax.microedition.io` can be used simultaneously since they do not involve the mixed use of packages from identical package groups. In contrast, `org.kwis.msp.lcuio`, `javax.microedition.lcdiui`, `org.kwis.msp.io`, and `javax.microedition.io` cannot be used simultaneously since they involve the mixed use of packages from identical package groups. Moreover, interoperability is not maintained.

Table 2. Corresponding packages between MSF/MSP and CLDC/MIDP by identical group.

Package Group	MSF/MSP Package	CLDC/MIDP Package
---------------	-----------------	-------------------

Package Group	MSF/MSP Package	CLDC/MIDP Package
Graphics	org.kwis.msp.lcdi (jlet) org.kwis.msp.lwc	javax.microedition.midlet javax.microedition.lcdi
DB	org.kwis.msp.db	javax.microedition.rms
Media	org.kwis.msp.media	javax.microedition.media javax.microedition.mediacontrol
High level IO	org.kwis.msp.io	javax.microedition.io

In addition, it is often necessary to call on other profiles for the required functions since a certain function is available only on one profile.

In this case, the platform should ensure interoperability for cross-reference and use since there are no corresponding packages.

Table 3. Unique packages of MSP/MSF and CLDC/MIDP.

Package Group	MSF/MSP Package	CLDC/MIDP Package
Low level IO	org.kwis.msf.io	None
Kernel	org.kwisf.core	None
Devices	org.kwis.msp.handset	None
Address book	org.kwis.msp.address org.kwis.msp.addressbook	None

When adopting the mixed use of packages as described earlier, additional limitations can be identified depending on the actual implementation of the platform since WIPI 2.0 does not provide an implementation method of the platform. As such, the content of this chapter can be revised through protocol revision.

2. Security Policy for MSF/MSP and CLDC/MIDP

- When an application starts with Jlet and uses both MSF/MSP and CLDC/MIDP, the platform should comply with WIPI's API security policy.
- When an application starts with Midlet and uses both MSF/MSP and CLDC/MIDP, the platform should comply with MIDP's security policy and ensure interoperability with WIPI's API security policy.

3. Return Value for Basic Error

Error Code	Value	Description	Remarks
M_E_ERROR	-1	Other error	
M_E_BADFD	-2	Wrong ID	
M_E_BADFILENAME	-3	Wrong filename	
M_E_BADSEEKPOS	-4	Wrong file location	
M_E_EXIST	-5	The resource already exists.	
M_E_BADFORMAT	-6	Wrong format	
M_E_INPROGRESS	-7	Operation in progress	
M_E_INUSE	-8	In use	
M_E_INVALID	-9	Invalid parameter	
M_E_ISCONN	-10	Connection already established	
M_E_LONGNAME	-11	Exceeds the allowed length	
M_E_NOENT	-12	No entry	
M_E_NOSPACE	-13	No space	
M_E_NOTCONN	-14	Connection not established	
M_E_NOTEMPTY	-15	Not empty	
M_E_NOTSUP	-16	Service unsupported	
M_E_NOMEMORY	-17	Insufficient memory	
M_E_SHORTBUF	-18	Buffer shortage	
M_E_WOULDBLOCK	-19	Would block generated	
M_E_TIMEOUT	-20	Timeout	
M_E_DATABIG	-21	Data too big	
M_E_BADRECID	-22	Wrong record ID	
M_E_EOF	-23	End of file	
M_E_ACCESS	-24	Access error	
M_E_NORESOURCE	-25	Insufficient resource	
M_E_NOTEXIST	-26	File/Resource does not exist.	
M_E_BADVERSION	-27	Platform version does not support the execution of the application.	
M_E_DEVCLOSE	-28	IO device closed	
M_E_OEMERROR	-29	Specific WIPI operation stopped due to OEM	
M_E_NOTSUPPORTTYPE	-30	Type not supported	
M_E_NOTSUPPORTLOCK	-31	Lock not supported	
M_E_NOTSUPPORTPLOCK	-32	Individual lock not supported	
M_E_NOTSUPPORTGLOCK	-33	Group lock not supported	
M_E_MAXCOUNT:	-34	More than the maximum value	

4. Usage Example

4.1. Usage Example of DLL

4.1.1. C API

In WIPI, functions that can add or override libraries dynamically should be provided in addition to the APIs that are established as standard and embedded in a terminal. Examples that can be used to develop dynamic linking libraries (DLL) in Clet and definition of the necessary APIs are presented.

A dynamic linking library has a route called interface that communicates with the external environment. An interface refers to a "unit that manages a group consisting of function and variables by assigning it a name and a version." When adding and updating APIs, an interface serves as the basic unit.

For example, the method of implementing and using DLLs on the part of a DLL developer or an application program developer providing 3D API is as follows:

- **Definition of DLL Export Function Header**

A 3D API DLL developer creates a head for the list group of functions to be provided to application program developers.

```
API3D.h <source1>
typedef struct _API3D
{
    M_Int32 (*draw3DLine)(M_Int32 x, M_Int32 y, M_Int32 x2,
M_Int32 y2);
    M_Int32 (*draw3DRect)(M_Int32 x, M_Int32 y, M_Int32 w, M_Int32
h);
    ...
}
API3D;
```

- **Implementing DLL Export Function**

A 3D API DLL developer creates a DLL that provides the "API3D.h" interface.

```
3D_DLL.c <source2>
#include "API3D.h"
#include "Demo_Interface.h"

M_Int32 dll_draw3DLine(M_Int32 x, M_Int32 y, M_Int32 x2, M_Int32
y2);
M_Int32 dll_draw3DRect(M_Int32 x, M_Int32 y, M_Int32 w, M_Int32
h);
```

```
API3D api3d = {
    dll_draw3DLine,
    dll_draw3DRect
};

DemoInf demoinf = {
    ...
};

MC_EXPORT_DLL_INTERFACE_START(module3D)
MC_DLL_INTERFACE(api3d, "Fast3D", 1, 0)
    MC_DLL_INTERFACE(demoinf, "TestDLL", 1, 0)
MC_EXPORT_DLL_INTERFACE_END

MC_EXPORT_DLL_START(module3D)
DLL_INIT(dll_init)
DLL_EXIT(dll_exit)
MC_EXPORT_DLL_END

M_Int32 dll_init() {
    return(0);
}

void dll_exit() {
}

M_Int32 dll_draw3DLine(M_Int32 x, M_Int32 y, M_Int32 x2, M_Int32
y2) {
    MC_knlPrintk("draw3d line\n");
    return(0);
}

M_Int32 dll_draw3DRect(M_Int32 x, M_Int32 y, M_Int32 w, M_Int32 h)
{
    MC_knlPrintk("draw3d rect\n");
    return(0);
}
```

A DLL developer should export a DLL interface. The exported DLL interface is exposed to application program developers through MC_knlGetDLLInterface() and is made available.

- **Using DLL Function**

A 3D API application program developer uses the DLL function as follows:

Assuming that 3D_DLL.c is compiled as a DLL option and is registered in the server under the program name = "3d_library," version = "1.1," and vendor = "testsoft," it can be used as:

```
3D_USE.c <source3>
#include "API3D.h"
...
API3D* inf3d;

int startClet(int argc, char* argc[])
{
    char buf[256];
    ...
    rtn = MC_knlGetExecNames("3d_library", NULL, NULL, buf,
        sizeof(buf));
    rtn = MC_knlLoad(buf, 0);

    inf3d = MC_knlGetDLLInterface("Fast3D", -1, -1, NULL, NULL);

    inf3d->draw3DLine (0, 0, 10, 10);
    rtn = inf3d->draw3DRect (10, 10), 50, 60);
    ...
}
```

- **Example of Adding/Overriding API Scenario**

When the "API3D.h" interface of <source1> is embedded as an example of overriding for embedded APIs, there is no change in the code of <source3>. When downloading the "API3D.h" DLL/interface, however, the "API3D.h" interface can be overridden.

In addition, when executing a program using a specific DLL API that is not embedded, the DLL should be downloaded first before it can be executed.

4.1.2. Java API

Since Java has functions that load and link all functions and variables dynamically as part of the characteristics of the language, dynamic linking libraries are implemented based on the linguistic characteristics of Java.

For example, the method of implementing and using DLLs on the part of a DLL developer or an application program developer providing 3D API is as follows:

- **Creating 3D_DLL.Jar to Provide for Application Program Developers**

A 3D API DLL developer creates 3D DLLs to be provided to application program developers.

```
Engine3D.java <source1>
package api3d;

public class Engine3D {
    public int draw3DLine(int x, int y, int x2, int y2) {
        ...
    }
    public int draw3DRect(int x, int y, int x2, int y2) {
        ...
    }
}
```

In Java, the 3D_DLL.jar library itself acts as a C header to application program developers.

- **Using 3D DLL Function**

A 3D API application program developer uses the 3D DLL function as follows:

Assuming that the DLL is registered in the server under the program name = "java_3d_library," version = "1.1," and vendor = "testsoft," the function is used as follows:

```
USE_3D.java <source2>
```

```

import api3d.Engine3D;

public class USE_3D {
    ...
    public int dLine(int x, int y, int x2, int y2) {
        String[] exeName = Kernel.getExecNames("java_3d_llibrary",
"1.1", "testsoft");
        Kernel.load(exeName[0], null);
        try {
            Class c = Class.forName("api3d.Engine3D");
            Engine3D obj3d = (Engine3D) c.newInstance();    ← Caution 1
            obj3d.draw3DLine(x, y, x2, y2);
        } catch(Error r) {
            System.out.println("3D library is not loaded");
        }
    }
    ...
}

```

In Precaution 1 of <source2>, unless the corresponding library is loaded using Kernel.load() before using 3D library API, "java/lang/Error" occurs.

4.2. Example of Terminal Performance

Measurement

Given the available diverse terminal devices and varying hardware in terms of processing speed, there is a need for developers who have to develop one application on a number of terminals to provide various versions of packages based on different screen sizes and terminal performance. To address difficulties encountered by developers, a tuning guide on performance is provided.

Before carrying out tuning, terminal performance should first be measured. The developer can prepare measurement items and source required for the measurement, although the measured data can be shared utilizing a tool used by many people. Among the performance benchmark tools that are easy to use, a benchmark tool called TaylorBench (<http://www.pogit.com/midp/bench/>) was created using Java MIDlet. It is available to anyone for use and correction for any purpose since no licensing is required, provided the developer's name is specified in "About."

The TaylorBench program consists of the low-level graphics of an MIDP-based device as well as simple items that measure VM/CPU performance. For applications requiring extensive operation, the program places weighted value on items of arithmetic operation. On the other hand, for applications with extensive screen movement, the program can

be run using the result values for test items.

The following is a brief discussion on each item:

● **Low-level Graphics**

Lines: Draws a line between two random points

Rectangles (outline and filled): Draws an outline and a filled rectangle of random size at random location

Ellipses (outline and filled): Draws an outline and a filled ellipse of random size at random location

Arcs (outline and filled): Draws an outline and a filled arc of random size at random location

Images (small, medium, large): Draws images having three different sizes at random location

Fonts (small, medium, large): Writes strings in three different fonts at random location

● **RMS**

Record creation: Creates records

Record reading using enumeration: Reads records in sequence

Record reading by randomly accessing records: Reads records from random locations

Record deletion by randomly deleting records: Deletes records

● **CPU/VM**

System array copies: As a native method that is not related to the Java implementation, system array copies are used to test CPU/VM performance.

VM test (multiply, divide, add): Displays the processing speed of arithmetic operation

Random ints.: Shows in a graph whether the numbers created by a random number generator are evenly distributed in the area

● **COMM**

Reads files (or dynamically generated number bytes) from an HTTP server

Writes files or specific number of bytes to an HTTP server

Reads local files: Reads files from among TaylorBench MIDlet jar files

※ Control item displays the amount of time taken by a loop for a set number of repeats.

First, an example source that applies frame conversion speed for a fixed screen without measured values on performance is presented.

The following is a brief description on each item:

```
Game.java <source1>
while (running) {
    long time = System.currentTimeMillis();

    moveSprites();
    checkCollision();
    repaint();
    serviceRepaints();

    time = System.currentTimeMillis();
    try {
        if (time < DELAY)
            Thread.sleep(DELAY - (int) time);
    } catch (Exception ex) {
    }
}
※ Reference: "UI Guidelines & Efficient MIDP Java Programming" by
Chiam Poh Guan (Forum Nokia)
```

5. wCard

Based on vCard 3.0, wCard has additional specifications that are necessary for the terminal environment even though they are not included in the vCard 3.0 specifications.

This specification provides a detailed explanation on the vCard 3.0 specifications and a definition of the wCard specifications.

5.1. Related Specifications

wCard specifications are based on the following specifications (in case parts of the following documents are not consistent with the requirements of this specification, the requirements of this specification shall take precedence; in principle, items that are not stipulated in the requirements of this specification will comply with the following specifications and related specifications):

- **RFC 2425 (MIME Content-Type for Directory Information)**
- **RFC 2426 (vCard MIME Directory Profile)**

5.2. Overview of vCard 3.0 Specifications

vCard format consists of strings. One field consists of a pair of <Type> and <Value> in a line, and each field is separated from other fields by a line change (CRLF). Denoting the data type, <Type> is used in the same way as N (name), NICKNAME (alias), and TEL (telephone number). Corresponding to <Type>, <Value> is used in the same manner as "N:Kim Cheol-soo" (when the name is Kim Cheol-soo), "NICKNAME:Wipi" (when the alias is Wipi), and "TEL:0200001111" (when the telephone number is 02-0000-1111).

- **<Type>**

<Type> consists of <Type Name> and <Type Parameter>; it is separated by ";"

Ex 1) "TEL;TYPE=home:0200001111" (when the home telephone number is 02-0000-1111)

Ex 2) "TEL;TYPE=work:0233334444" (when the company telephone number is 02-3333-4444)

- **<Type Parameter>**

One <Type Name> may have no <Type Parameter> or multiple parameters. In case of multiple <Type Parameters>, each <Type Parameter> is separated by “,.” Before each <Type Parameter> is “TYPE=.”

Ex) “TEL;TYPE=work,cell:01100001111” (when the company mobile phone number is 011-0000-1111)

- **<Value>**

<Value> denotes data for <Type>; multiple <Values> are possible for one <Type>. In case of multiple <Values>, each <Value> is separated by “,.”

Ex 1) “ORG:Korea Telecom;Jeju Research Institute” (if the workplace is Jeju Research Institute, Korea Telecom)

In case of multiple <Values> for one <Type>, <Values> may be deleted but not the separator “,.” For the last <Value>, however, the separator “,.” may be deleted.

Ex 2) “ORG::Jeju Research Institute” (when specifying only the Jeju Research Institute, Korea Telecom as the workplace, the “,.” in front of the “Jeju Research Institute” should not be deleted)

Ex 3) “ORG:'Korea Telecom” (when specifying only the Korea Telecom if the workplace is Jeju Research Institute, Korea Telecom - there will be no “,.” following “Korea Telecom”)

5.3. vCard 3.0 Specifications: Examples

- **<Type Name>**

The following are some of the <Type Names> defined in RFC2426:

Type Name	Description
N	Name
NICKNAME	Alias
TEL	Telephone number
E-MAIL	E-mail

URL	Homepage
ADR	Address
NOTE	Memo
ROLE	Occupation
ORG	Workplace
PHOTO	Photo
REV	Revised information

● **<Type Parameter>**

The following are some of the <Type Parameters> defined in RFC2426:

<Type Name>	<Type Parameter>
N	
NICKNAME	
TEL	work (workplace), home (home), cell (cellular phone), fax (fax), pager (pager)
E-MAIL	
URL	work (workplace)
ADR	home (home), work (workplace)
NOTE	
ROLE	
ORG	
PHOTO	
REV	
All Type Names	pref (preferred)

Ex 1) "TEL;TYPE=pager:01212345678" (When the pager number is 012-1234-5678)

Ex 2) "TEL;TYPE=fax:0298765432" (When the fax number is 02-9876-5432)

5.4. wCard Specifications

Based on vCard3.0, wCard specifications has the following items that are not defined in the vCard 3.0 specifications:

5.4.1. Add <Types>

<Type Name>	Description														
X-MDAY	<p>Record anniversaries (birthday, wedding anniversary, meeting, memorial day)</p> <p>Follow the "<Type Name>;<Type Parameter>-1,<Type Parameter>-2:<Value>" format without space between characters.</p> <table border="1" data-bbox="578 873 1357 1262"> <thead> <tr> <th data-bbox="578 873 967 903"><Type Parameter></th> <th data-bbox="967 873 1357 903">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="578 903 967 963">birthday</td> <td data-bbox="967 903 1357 963">Birthday</td> </tr> <tr> <td data-bbox="578 963 967 1024">wedding</td> <td data-bbox="967 963 1357 1024">Wedding</td> </tr> <tr> <td data-bbox="578 1024 967 1085">meeting</td> <td data-bbox="967 1024 1357 1085">Meeting</td> </tr> <tr> <td data-bbox="578 1085 967 1146">memorial</td> <td data-bbox="967 1085 1357 1146">Memorial day</td> </tr> <tr> <td data-bbox="578 1146 967 1207">sun</td> <td data-bbox="967 1146 1357 1207">Julian calendar</td> </tr> <tr> <td data-bbox="578 1207 967 1262">moon</td> <td data-bbox="967 1207 1357 1262">Lunar calendar</td> </tr> </tbody> </table> <p><Type Parameter>-1 is either the birthday, wedding, meeting, or memorial day.</p> <p><Type Parameter>-2 is either the sun or moon.</p> <p>For example, record "memorial, sun" or "wedding, moon."</p> <p><Value> is the memorial day.</p> <p>For a terminal that supports year display for a memorial day (date), the entry consists of eight digits (YYYYMMDD), e.g., "20030101" and "20031231."</p> <p>For a terminal that does not support year display for a memorial</p>	<Type Parameter>	Description	birthday	Birthday	wedding	Wedding	meeting	Meeting	memorial	Memorial day	sun	Julian calendar	moon	Lunar calendar
<Type Parameter>	Description														
birthday	Birthday														
wedding	Wedding														
meeting	Meeting														
memorial	Memorial day														
sun	Julian calendar														
moon	Lunar calendar														

	<p>day (date), the entry consists of four digits (MMDD), e.g., "1231."</p> <p>When the <Type parameter> is moon, "L" added at the end of the month means leap month.</p> <p>Ex 1) "X-MDAY;TYPE=wedding,sun:19960415" (when a wedding anniversary falls on April 15, 1996 in the Julian calendar)</p> <p>Ex 2) "X-MDAY;TYPE=meeting,moon:20040213" (when the memorial day for a meeting falls on February 13, 2004 in the lunar calendar)</p> <p>Ex 3) "X-MDAY;TYPE=meeting,moon:20040213L" (when the memorial day for a meeting falls on February 13, 2004, a leap month)</p> <p>Ex 4) "X-MDAY;TYPE=meeting,moon:1102" (when the memorial day for a meeting falls on November 2 in the lunar calendar, in case the terminal does not support year display)</p>
X-GROUP	<p>Record the resource name of a phonebook group to which a phonebook individual belongs.</p> <p>Follow the "<Type Name>:<Value>" format without space between characters.</p> <p>Only one Value exists for one X-GROUP Type.</p> <p>Ex) "X-GROUP:Friend," "X-GROUP:School," "X-GROUP:Classmate"</p> <p>In the absence of X-GROUP, the default phonebook group of the terminal is specified (ex: a phonebook group such as "Not Specified").</p> <p>When a phonebook group is deleted, all phonebook individuals belonging to the phonebook group are specified as a default phonebook group (ex: a phonebook group such as "Not Specified").</p>

5.4.2. Add <Type Parameter>:

<Type Name>: PHOTO	
<Type Parameter>	Description
termres	<p>Specify a photo of the terminal resource to the phonebook individual.</p> <p>Follow “<Type Name>;<Type Parameter>:<Value>-1;<Value>-2” format without space between characters.</p> <p><Value>-1 is the resource group name of the photo; <Value>-2 is the resource name.</p> <p>Ex 1) “PHOTO;TYPE=termres:PICTUREMATE;mypicture” (when mypicture of PICTUREMATE is specified as the photo)</p> <p>Ex 2) “PHOTO;TYPE=termres:PHOTO;myphoto” (When myphoto of PHOTO is specified as the photo)</p>
<Type Name>: TEL	
<Type Parameter>	Description
etc	<p>Other telephone number</p> <p>Ex) Other telephone number: 02-1234-5678</p> <p>→ “TEL;TYPE=etc:0212345678”</p>

5.4.3. Usage of <Value>

<Type Name>	<Value>
N	<p>“Family name;Given name”</p> <p>Ex 1) “N:Kim;Cheol-soo” (when the name is Kim Cheol-soo)</p> <p>Ex 2) “N:Chae;Shi-ra” (when the name is Chae Shi-ra)</p> <p>Ex 3) “N:Kim;Hye-soo” (when the name is Kim Hye-soo)</p>

	<p>“Entire name” (Family name+Given name)</p> <p>Ex 1) “N:Kim Cheol-soo” (when the name is Kim Cheol-soo)</p> <p>Ex 2) “N:Chae Shi-ra” (when the name is Chae Shi-ra)</p> <p>Ex 3) “N: Kim Hye-soo” (when the name is Kim Hye-soo)</p>
ADR	<p>“;;Address;Gu/Gun/Shi;Shi/Do;Zip Code”</p> <p>Ex 1) When the home address is WIPI Research Institute, Bongcheon-Dong, Guanak-Gu, Seoul (zip code: 123-456):</p> <p>→ “ADR;TYPE=home;;Bongcheon-Dong WIPI Research Institute;Guanak-Gu;Seoul;123-456”</p> <p>Ex 2) When the work address is WIPI Research Institute, Bundang-Gu, Seongnam-Shi, Gyeonggi-Do (zip code: 000-111):</p> <p>→ “ADR;TYPE=work;;Bundang-Ku WIPI Research Institute;Seongnam-Shi;Gyeonggi-Do;000-111”</p> <p>→ “ADR;TYPE=work;;Seongnam-Shi Bundang-Ku WIPI Research Institute;;Gyeonggi-Do;000-111”</p> <p>→ “ADR;TYPE=work;;Gyeonggi-Do Seongnam-Shi Bundang-Ku WIPI Research Institute;;000-111”</p> <p>→ “ADR;TYPE=work;; Gyeonggi-Do Seongnam-Shi Bundang-Ku WIPI Research Institute” (when zip code is not used)</p>
ORG	<p>“Company;Department”</p> <p>Ex) When specifying Jeju Research Institute, Korea Telecom as the workplace:</p> <p>→ “ORG:Korea Telecom;Jeju Research Institute” (when both the company and department names are used)</p> <p>→ “ORG:Korea Telecom” (when only the company name is used)</p> <p>→ “ORG;; Jeju Research Institute” (when only the department</p>

	name is used)
PHOTO	<p>“Resource group name;Resource name” (<Type Parameter>: termres)</p> <p>Ex) When “myphoto” resource is specified in the PHOTO group</p> <p>→ “PHOTO;TYPE=termres:PHOTO;myphoto”</p>

5.4.4. Other Settings

- **Use of “pref”**

“pref” <Type parameter> can be used for all <Type Name>.

In case of several identical <Type Name>, “pref” <Type parameter> should be specified at least for one of them.

In case of several identical <Type Name>, “pref” <Type parameter> should be specified at least for one of them.

“pref” should be located before all <Type parameter>.

Ex 1) TEL;TYPE=pref,work:01188880000

TEL;TYPE=home:01199991111

TEL;TYPE=cell:01112345678

Ex 2) X-MDAY;TYPE=pref,birthday,moon:19700505

X-MDAY;TYPE=wedding,sun:20000101

- **Record the telephone number without the hyphen (“-”).**

Ex 1) 01100001111 (O)

Ex 2) 011-0000-1111 (X)

- **Always record the <Type Parameter> “work” at the URL <Type Name>**

Ex) URL;TYPE=work:http://www.nate.com

- **When using “:,” “;,” “\” for <Value> content, place them after the back slash (“\”). For “:” and “;” used as separator, however, a back slash (“\”) is not necessary.**

Ex 1) When the alias is “My nick;name”:

→ NICKNAME:My nick\;name

Ex 2) When the company is “Korea Telecom,” and the department is “Jeju Research:Institute”:

→ ORG:Korea Telecom;Jeju Research\Institute

Ex 3) When the company is “Korea Telecom,” and the department is “Jeju Research\Institute”:

→ ORG:ORG:Korea Telecom;Jeju Research\\Institute

- **When the <Type parameter> representing the type of X-MDAY assumes a variable format that can be entered directly by the user, and if “:” “;” “,” “\” are used as <Type parameters>, place them after a back slash (“\”). For “:” and “;” used as separator, however, a back slash (“\”) is not necessary.**

Ex 1) When the <Type parameter> is “Girl friend;the Day (date) that I met,” which falls on January 1 in the Julian calendar (for a terminal that does not support year: MMDD):

→ X-MDAY;Type=Girl friend\;the Day (date) that I met,sun:0101

Ex 2) When the <Type parameter> is “Girl friend:100 Days,” which falls on December 12, 2000 in the lunar calendar (for a terminal that supports year display YYYYMMDD)

→ X-MDAY;Type=Girl friend\:100 Days,moon:20001212

Ex 2) When the <Type parameter> is “Girl friend\Separated,” which falls on August 15, 2001 in the lunar calendar (for a terminal that supports year display YYYYMMDD)

→ X-MDAY;Type=Girl friend\\Separated,moon:20010815

Ex 3) When the <Type parameter> is “Girl friend,Met again,” which falls on August 15, 2001 in the lunar calendar (for a terminal that supports year display YYYYMMDD):

→ X-MDAY;Type=Girl friend\,Met again,moon:20010815

- **REV**

1. When generating or revising a phonebook individual on OEM, the time information should be recorded or amended on OEM.
2. When the WIPI address book application MC_termResRead or MH_termResRead requests for phonebook individual data, OEM should provide time information in wCard format and REV type.
3. When a phonebook individual is generated or modified in a WIPI address book application, OEM should record or modify the time information in wCard REV value.
4. When generating or modifying a phonebook individual in a WIPI address book application, if there is no REV value in wCard, OEM should record or modify the time information beginning with the time when MC_termResWrite or MH_termResWrite is called.

5.5. Example of wCard Specifications

- **<Type Name>**

The following are <Type Name> examples that can be used in wCard:

<Type Name>	Description
N	Name
NICKNAME	Alias
TEL	Telephone number
E-MAIL	E-mail
URL	Homepage
ADR	Address
NOTE	Memo
ROLE	Occupation
ORG	Workplace
X-MDAY	Memorial day (absent in vCard 3.0 but added in wCard)
X-GROUP	Phonebook group (absent in vCard 3.0 but added in wCard)
PHOTO	Photo

REV	Revised information
-----	---------------------

- **<Type Parameter>**

The following are examples of <Type Parameter> that can be used in wCard:

<Type Name>	<Type Parameter>
N	
NICKNAME	
TEL	work (workplace), home (home), cell (cellular phone), fax (fax), pager (pager), etc.
EMAIL	
URL	work (workplace)
ADR	home (home), work (workplace)
NOTE	
ROLE	
ORG	
X-MDAY	birthday (birthday), wedding (wedding), meeting (meeting), memorial (memorial day), sun (Julian calendar), moon (lunar calendar)
X-GROUP	
PHOTO	termres (terminal resource)
REV	
All Type Names	pref (preferred)

5.6. Examples of wCard

- **Example 1**

When MC_termResGetGroupInfo or MH_termResGetGroupInfo results in the following:

MC_termResGetGroupInfo MH_termResGetGroupInfo	
InfoType	InfoData

"TYPELIST"	"N\0NICKNAME\0TEL\0EMAIL\0URL\0ADR\0NOTE\0ROLE\0ORG\0X-MDAY\0X-GROUP\0PHOTO\0REV\0\0"
"TYPEINFO/N"	"10/1"
"TYPEINFO/NICKNAME"	"10/1"
"TYPEINFO/TEL"	"16/7/work:7/home:7/cell:7/fax:7/pager:7/etc:7"
"TYPEINFO/EMAIL"	"30/2"
"TYPEINFO/URL"	"50/1"
"TYPEINFO/ADR"	"50/2/home:2/work:2"
"TYPEINFO/NOTE"	"40/1"
"TYPEINFO/ROLE"	"20/1"
"TYPEINFO/ORG"	"30/1"
"TYPEINFO/X-MDAY"	"4/4/birthday:4/wedding:4/meeting:4/memorial:4/sun:4/moon:4"
"TYPEINFO/X-GROUP"	"10/3"
"TYPEINFO/PHOTO"	"20/1/termres:1"
"TYPEINFO/REV"	"16/1"
"X-MDAYINFO"	"MMDD/FIXED"

BEGIN:VCARD

N:Kim;Cheol-soo

```
NICKNAME:WIPI
TEL;TYPE=pref,work:0200001111
TEL;TYPE=work:0211110000
TEL;TYPE=home:0222223333
TEL;TYPE=cell:01112345678
TEL;TYPE=fax:0267896789
TEL;TYPE=pager:01203690369
TEL;TYPE=etc:0212345678
EMAIL;TYPE=pref:wipi@nate.com
EMAIL:wipi-apps@nate.com
URL;TYPE=work:http://www.nate.com
ADR;TYPE=pref,home;;;Bongcheon-Dong WIPI Research Institute;Gwanak-
Gu;Seoul;123-456
ADR;TYPE=work;;;Bundang-Ku WIPI Research Institute;Seongnam-Shi;Gyeonggi-
Do;000-111
NOTE:wCard example
ROLE:Programmer
ORG:Korea Telecom;Jeju Research Institute
X-MDAY;TYPE=pref,birthday,moon:0505
X-MDAY;TYPE=wedding,sun:0101
X-MDAY;TYPE=meeting,sun:1231
X-MDAY;TYPE=memorial,moon:0630
X-GROUP;TYPE=pref:friend
X-GROUP:school
X-GROUP:classmate
PHOTO;TYPE=termres:PHOTO;myphoto
REV:20031009T180135Z
END:VCARD
```

The following are the descriptions of the wCard examples presented above:

Name: Kim Cheol-soo (family name: Kim, given name: Cheon-soo)
Alias: WIPI
Tel (work, preferred): 02-0000-1111
Tel (work): 02-1111-0000
Tel (home): 02-2222-3333
Tel (HP): 011-1234-5678
Tel (Fax): 02-6789-6789
Tel (Pager): 012-0369-0369
Tel (other): 02-1234-5678
E-mail (preferred): wipi@nate.com
E-mail: wipi-apps@nate.com
Homepage: <http://www.nate.com>
Address (home, preferred): Bongcheon-Dong WIPI Research Institute, Gwanak-Ku, Seoul (zip code: 123-456)
Address (work): Bundang-Ku WIPI Research Institute, Seongnam-Shi, Gyeonggi-Do (zip code: 000-111)
Memo: wCard example
Occupation: Programmer
Workplace: Korea Telecom (company), Jeju Research Institute (department)
Memorial day (birthday, preferred): May 5 (lunar calendar)
Memorial day (wedding anniversary): January 1 (Julian calendar)
Memorial day (meeting): December 31 (Julian calendar)
Memorial day (memorial day): June 30 (lunar calendar)
Phonebook group (preferred): Friend
Phonebook group: School
Phonebook group: Classmate
Revised on: 18:01:35 October 9, 2003 (GMT) or 03:01:35 October 10, 2003 (GMT + 09:00)

Photo: myphoto of the terminal resource PHOTO group

● **Example 2**

When MC_termResGetGroupInfo or MH_termResGetGroupInfo results in the following:

MC_termResGetGroupInfo MH_termResGetGroupInfo	
infoType	infoData
"TYPELIST"	"N\0TEL\0EMAIL\0NOTE\0X-MDAY\0X-GROUP\0REV\0\0"
"TYPEINFO/N"	"10/1"
"TYPEINFO/TEL"	"16/4/work:1/home:1/fax:1/cell:1"
"TYPEINFO/EMAIL"	"30/1"
"TYPEINFO/NOTE"	"40/1"
"TYPEINFO/X-MDAY"	"8/2/sun:2/moon:2"
"TYPEINFO/X-GROUP"	"10/1"
"TYPEINFO/REV"	"16/1"
"X-MDAYINFO"	"YYYYMMDD/VARIABLE/10"

```

BEGIN:VCARD
N:Kim Cheol-soo
TEL;TYPE=pref,work:0200001111
TEL;TYPE=home:0211110000
TEL;TYPE=fax:0222223333
TEL;TYPE=cell:01112345678
EMAIL:wipi@nate.com

```

```

NOTE:wCard example

X-MDAY;TYPE=pref,birthday,moon:19700505

X-MDAY;TYPE=day of first meeting,sun:20000101

X-GROUP:friend

REV:20031009T180135Z

END:VCARD

```

The following are the descriptions of the wCard examples presented above:

```

Name: Kim Cheol-soo

Tel (work, preferred): 02-0000-1111

Tel (home): 02-1111-0000

Tel (fax): 02-2222-3333

Tel (HP): 011-1234-5678

E-mail: wipi@nate.com

Memo: wCard example

Memorial day (birthday, preferred): May 5, 1970 (lunar calendar)

Memorial day (first meeting): January 2, 2000 (Julian calendar)

Phonebook group: Friend

Revised on: 18:01:35 October 9, 2003 (GMT) or 03:01:35, October 10, 2003 (GMT +
09:00)

```

● Example 3

When MC_termResGetGroupInfo or MH_termResGetGroupInfo results in the following:

MC_termResGetGroupInfo MH_termResGetGroupInfo	
InfoType	InfoData
"TYPELIST"	"N\0TEL\0EMAIL\0NOTE\0X-MDAY\0X-GROUP\0REV\0\0"
"TYPEINFO/N"	"10/1"
"TYPEINFO/T EL"	"16/4/work:1/home:1/fax:1/cell:1"

"TYPEINFO/E MAIL"	"30/1"
"TYPEINFO/N OTE"	"40/1"
"TYPEINFO/X- MDAY"	"8/2/birthday:2/wedding:2/meeting:2/memorial:2/sun:2/moon:2 "
"TYPEINFO/X- GROUP"	"10/1"
"TYPEINFO/R EV"	"16/1"
"X- MDAYINFO"	"YYYYMMDD/FIXED"

```

BEGIN:VCARD
N:Kim Cheol-soo
TEL;TYPE=pref,work:0200001111
TEL;TYPE=home:0211110000
TEL;TYPE=fax:0222223333
TEL;TYPE=cell:01112345678
EMAIL:wipi@nate.com
NOTE:wCard example
X-MDAY;TYPE=pref,birthday,moon:19700505
X-MDAY;TYPE=wedding,sun:20000101
X-GROUP:friend
REV:20031009T180135Z
END:VCARD

```

The following are the descriptions of the wCard examples presented above:

```

Name: Kim Cheol-soo
Tel (work, preferred): 02-0000-1111

```

Tel (home): 02-1111-0000

Tel (fax): 02-2222-3333

Tel (HP): 011-1234-5678

E-mail: wipi@nate.com

Memo: wCard example

Memorial day (birthday, preferred): May 5, 1970 (Lunar calendar)

Memorial day (wedding anniversary): January 1, 2000 (Julian calendar)

Phonebook group: Friend

Revised on: 18:01:35 October 9, 2003 (GMT) or 03:01:35 October 10, 2003 (GMT + 09:00)