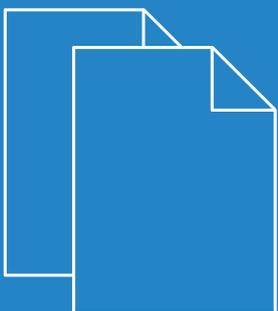


LANCOM Wireless ePaper Server

Integration MS Exchange



Inhalt

1 Zielsetzung und Voraussetzungen.....	4
1.1 Zielsetzung.....	4
1.2 Voraussetzungen.....	4
2 Anpassungen in Microsoft Exchange.....	5
3 Skript- und Konfigurationsdatei.....	7
3.1 Erläuterung des Skripts.....	7
3.2 Beispiel einer Konfigurationsdatei.....	7
4 Start des Update-Prozesses.....	10
5 Template für Raumbeschilderung.....	12
5.1 Beispiel eines Templates.....	12
5.2 Erläuterungen einzelner Codeabschnitte.....	13
5.3 Interaktion von Template, XML-Informationen und Wireless ePaper Displays.....	14

Copyright

© 2019 LANCOM Systems GmbH, Würselen (Germany). Alle Rechte vorbehalten.

Alle Angaben in dieser Dokumentation sind nach sorgfältiger Prüfung zusammengestellt worden, gelten jedoch nicht als Zusicherung von Produkteigenschaften. LANCOM Systems haftet ausschließlich in dem Umfang, der in den Verkaufs- und Lieferbedingungen festgelegt ist.

Weitergabe und Vervielfältigung der zu diesem Produkt gehörenden Dokumentation und Software und die Verwendung ihres Inhalts sind nur mit schriftlicher Erlaubnis von LANCOM Systems gestattet. Änderungen, die dem technischen Fortschritt dienen, bleiben vorbehalten.

Windows® und Microsoft® sind eingetragene Marken von Microsoft, Corp.

LANCOM, LANCOM Systems, LCOS, LANcommunity und Hyper Integration sind eingetragene Marken. Alle anderen verwendeten Namen und Bezeichnungen können Marken oder eingetragene Marken ihrer jeweiligen Eigentümer sein. Dieses Dokument enthält zukunftsbezogene Aussagen zu Produkten und Produkteigenschaften. LANCOM Systems behält sich vor, diese jederzeit ohne Angaben von Gründen zu ändern. Keine Gewähr für technische Ungenauigkeiten und / oder Auslassungen.

Das Produkt enthält separate Komponenten, die als sogenannte Open Source Software eigenen Lizenzen, insbesondere der General Public License (GPL), unterliegen. Die Lizenzinformationen zur Geräte-Firmware (LCOS) finden Sie auf der WEBconfig des Geräts unter dem Menüpunkt „Extras > Lizenzinformationen“. Sofern die jeweilige Lizenz dies verlangt, werden Quelldateien zu den betroffenen Software-Komponenten auf Anfrage über einen Download-Server bereitgestellt.

Produkte von LANCOM Systems enthalten Software, die vom „OpenSSL Project“ für die Verwendung im „OpenSSL Toolkit“ entwickelt wurde (www.openssl.org).

Produkte von LANCOM Systems enthalten kryptographische Software, die von Eric Young (eay@cryptsoft.com) geschrieben wurde.

Produkte von LANCOM Systems enthalten Software, die von der NetBSD Foundation, Inc. und ihren Mitarbeitern entwickelt wurde.

Produkte von LANCOM Systems enthalten das LZMA SDK, das von Igor Pavlov entwickelt wurde.

LANCOM Systems GmbH

Adenauerstr. 20/B2

52146 Würselen

Deutschland

www.lancom-systems.de

1 Zielsetzung und Voraussetzungen



Die hier aufgeführten Anleitungen wenden sich ausschließlich an Experten! Sie beschreiben zudem Systeme einer bestimmten Software-Version und dienen daher ausschließlich zu Illustrationszwecken.

1.1 Zielsetzung

Das Ziel dieses Dokumentes ist eine Möglichkeit aufzuzeigen, wie eine erfolgreiche Integration der LANCOM Wireless ePaper Solutions in einem Netzwerk mit Microsoft Exchange umgesetzt werden kann. Dies erfolgt anhand einer IronPython-Skriptdatei, einer zugehörigen Konfigurationsdatei und einem XSL-Template.

Bei dem Code von Skriptdatei, Konfigurationsdatei und Template handelt es sich um Beispiele, welche die Erstellung eigener Dateien vereinfachen soll. Soll der Code produktiv eingesetzt werden, müssen Angaben in der Konfigurationsdatei, entsprechend geändert werden. Weitere Templates können erstellt, beziehungsweise genutzt werden, um eine optimale Anpassung an die geplante Nutzung zu erreichen. Des Weiteren kann die Skriptdatei umgeschrieben oder neu geschrieben werden, wenn die bereits vorhandene nicht ausreichen sollte.

1.2 Voraussetzungen

Die erfolgreiche Integration der LANCOM Wireless ePaper Solutions in einer Microsoft Exchange-Umgebung erfordert gewisse Voraussetzungen an vorhandener Software und deren Anpassung. Alle drei Komponenten werden separat aufgeführt, müssen aber nicht zwingend auf unterschiedlichen Netzwerkreisourcen installiert werden. Der Wireless ePaper Server und die Access Points der E-Serie sollten auf Grund von Latenz allerdings über das gleiche lokale Netzwerk verbunden sein.

1. Skript

Folgende Software ist auf dem Rechner, der das Skript ausführt, erforderlich:

- > Microsoft Windows 7, Windows 8, Windows 8.1, Windows Server 2008 R2, Windows Server 2012
- > IronPython Version 2.7.4
- > Exchange Web Services Managed API Version 2.2
- > .NET Framework 3.5 oder höher
- > Es muss in der Umgebungsvariablen IRONPYTHONPATH der Pfad zur installierten DLL der Microsoft Web Services Managed API angegeben werden. Sollte die Umgebungsvariable noch nicht existieren, muss diese angelegt werden.

2. Microsoft Exchange

Microsoft Exchange muss bereits zur Ressourcenreservierung (Meetingräume) eingerichtet sein. Unterstützt werden Microsoft Exchange 2007 SP1, 2010, 2010 SP1, 2010 SP2, 2013 und 2013 SP1

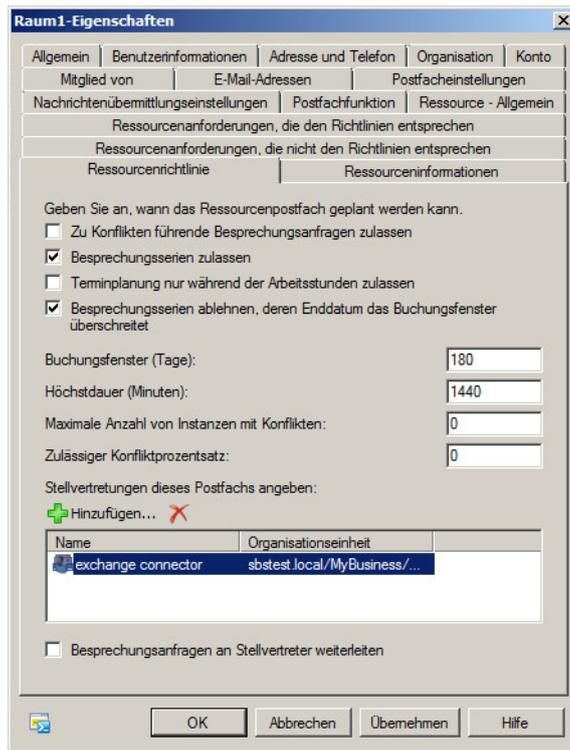
3. LANCOM Wireless ePaper Server

Folgende Voraussetzungen müssen bei dem LANCOM Wireless ePaper Server gegeben sein:

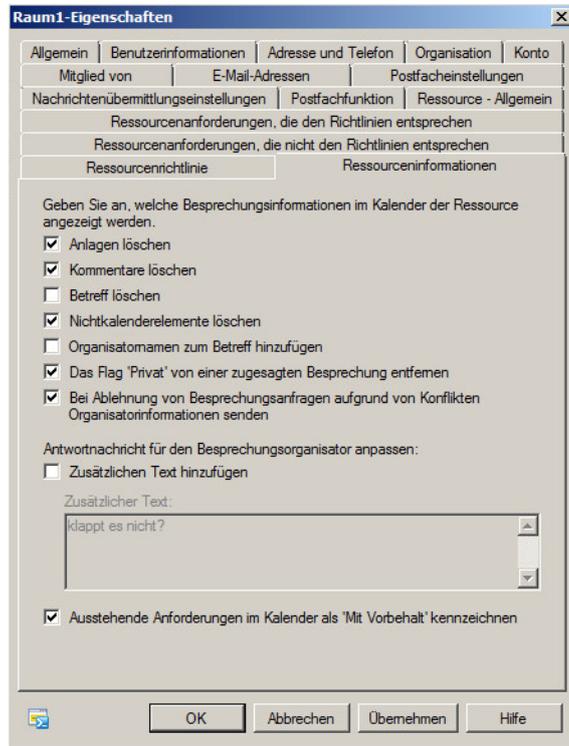
- > Das zu nutzende Template muss unter `\\<Installationsverzeichnis>\data\templates\` abgelegt werden.
- > Bilder, auf die im Template verwiesen wird, müssen unter `\\<Installationsverzeichnis>\data\images\` abgelegt werden.

2 Anpassungen in Microsoft Exchange

Meetingräume haben in Microsoft Exchange eigene Postfächer, in denen die Einladungen zu Meetings eingehen. Damit das Skript auf diese zugreifen kann, muss auf dem Exchange Server ein Benutzer eingerichtet werden, der Zugriff auf diese Postfächer hat. Dieser kann unter der Registerkarte **Ressourcenrichtlinie** als Stellvertreter in den einzelnen Postfächern eingetragen werden.



Zusätzlich müssen unter der Registerkarte **Ressourceninformationen** die Optionen **Betreff löschen** und **Organistornamen zum Betreff hinzufügen** entfernt werden, da das Skript die Reservierung sonst nicht oder nicht korrekt abfragen kann.



! In Microsoft Exchange 2013 fehlen die Optionen **Betreff löschen** und **Organistornamen zum Betreff hinzufügen** in der grafischen Benutzeroberfläche. Zum Ändern dieser Einstellungen führen Sie am Beispiel der Identität **boardroom** auf der Kommandozeilenebene des Exchange Servers folgenden Befehl aus:
`Set-CalendarProcessing -identity boardroom -AddOrganizerToSubject $false -DeleteSubject $false.`

3 Skript- und Konfigurationsdatei

Skript- und Konfigurationsdatei sind eine logische Einheit, bei der die Skriptdatei den Programm-Code enthält und die Konfigurationsdatei die Parameter definiert, die von Installation zu Installation unterschiedlich sind.

3.1 Erläuterung des Skripts

Die Skriptdatei `exchange_display_updater.py` enthält den Code, der die relevanten Informationen des Exchange Servers abfragt, verarbeitet und bei Bedarf ein Update an den LANCOM Wireless ePaper Server sendet. Die Skriptdatei finden Sie auf der [LANCOM Homepage](#) im Produktbereich der LANCOM Wireless ePaper Displays zum Download. Auf eine detaillierte Erläuterung des Codes wird an dieser Stelle verzichtet, da alle notwendigen Änderungen für die Integration über die Konfigurationsdatei durchführbar sind.

3.2 Beispiel einer Konfigurationsdatei

Die Konfigurationsdatei `config.json` (es kann ein beliebiger Dateiname gewählt werden) enthält die Informationen über den LANCOM Wireless ePaper Server, den Microsoft Exchange Server, das genutzte Template und die eingesetzten Displays. Sie enthält Code in JSON-Schreibweise und muss in UTF-8 ohne BOM formatiert sein. Anbei ein Beispiel einer Konfigurationsdatei, anhand derer die einzelnen Einträge erläutert werden.

```
{
  "wireless_display_server": {
    "address": "wds.sbstest.local",
    "port": 8001
  },
  "exchange_server": {
    "version": "Exchange2010_SP1",
    "default_credentials": false,
    "user": "exchangeconnector",
    "password": "secret2011!",
    "domain": "sbstest.local",
    "autodiscover_url_from_email": true,
    "autodiscover_email": "exchangeconnector@sbstest.local",
    "url": "https://sbs2011.sbstest.local/EWS/Exchange.asmx"
  },
  "conference_label": {
    "template": "lcsconference_landscape.xsl",
    "no_new_data_message": "Keine Reservierungen"
  },
  "displays" : [
    {
      "exchange_room_mailbox": "Raum1@sbstest.local",
      "display_id": "D1002DE2",
      "display_name": "Meetingraum 1"
    },
    {
      "exchange_room_mailbox": "Raum2@sbstest.local",
      "display_id": "D1002FED",
      "display_name": "Meetingraum 2"
    }
  ]
}
```

```
]
}
```

wireless_display_server

Bestimmt den LANCOM Wireless ePaper Server. Es wird normalerweise nur die Adresse benötigt bzw. localhost, wenn es sich um das lokale System handelt.

exchange_server

Bestimmt den Microsoft Exchange Server.

version

Bestimmt die Version, welche für die API wichtig ist. Zur Auswahl stehen:

- > Exchange2007_SP1
- > Exchange2010
- > Exchange2010_SP1
- > Exchange2010_SP2
- > Exchange2013
- > Exchange2013_SP1

default_credentials

Besitzt dieser Eintrag den Wert `true`, so werden die Anmeldedaten des aktuellen Windows-Benutzers zum Zugriff auf die Raum-Mailboxen verwendet. Bei `false` werden dagegen die Werte von `user`, `password` und `domain` verwendet.



Es wird im Beispielcode davon ausgegangen, dass die Kommunikation mit dem Server in einem vertrauenswürdigen Netz bzw. lokal stattfindet. Die Netzwerkkommunikation/Zugangsdaten werden nicht gesondert geschützt.

autodiscover_url_from_email

Besitzt dieser Eintrag `true`, so kann eine E-Mail-Adresse aus der Domäne verwendet werden, um den Exchange-Server zu finden. Die E-Mail-Adresse wird über `autodiscover_email` spezifiziert. Bei `false` muss dagegen der Exchange-Service genau per URL spezifiziert werden.

conference_label

Bestimmt die Basis-Eigenschaften der Konferenzraumbeschilderung.

template

Gibt das XSL-Template aus dem Template-Ordner an.

no_new_data_message

Erscheint, wenn für den aktuellen Tag keine Reservierungen mehr vorliegen.

displays

Spezifiziert die eingesetzten Wireless ePaper Displays. Diese werden als Liste von einzelnen Display-Objekten angegeben. Sie haben die folgenden Eigenschaften:

exchange_room_mailbox

Für die E-Mail-Adresse des Meetingraum-Postfachs

display_id

Für die Display ID (in der Regel 8-stellig hexadezimal)

display_name

Für den Raumnamen. Dieser ist fest und wird nicht durch die Raum-Reservierungen beeinflusst.

4 Start des Update-Prozesses

Um den Update-Prozess zu starten, muss das Skript auf der Kommandozeile, per Batch oder per Dienst gestartet werden. Die Syntax hierfür lautet:

```
ipy.exe exchange_display_updater.py [-h] [-i <interval>] [-f] [-u (always|newdata|required)]  
[-d (debug|info|warning|error|critical)] <configfile>
```

Bedeutung der einzelnen Argumente:

-h

Anzeige einer Hilfe zum Skript und dessen Argumenten.

-i <interval>

Angabe des Abstandes zwischen den einzelnen Prüfungen in Minuten.

-f

Die erste Prüfung findet zur nächsten vollen Stunde statt.

-u (always | newdata | required)

Angabe, wie die Updates durchgeführt werden:

always

Bei jeder Prüfung wird ein Update an den Wireless ePaper Server zum Update der Displays gesendet. Dies sollte nur zu Tests oder bei sehr langen Intervallen verwendet werden, um die Batterien der Displays nicht unnötig zu belasten.

newdata

Es wird nur dann ein Update des Displays durch den Wireless ePaper veranlasst, wenn sich die neuen Daten von den bereits dargestellten unterscheiden.

required

Update erfolgt wie bei `newdata` mit dem Unterschied, dass auch dann ein Update veranlasst wird, wenn das letzte Update nicht in den Status `SUCCESSFUL` gelangt ist. Diese Einstellung ist zu empfehlen, wenn sich Displays im Grenzbereich der AP-Abdeckung befinden.

-d (debug | info | warning | error | critical)

Angabe des Debug-Levels für das interne Logging. Der Default ist `warning`: Es werden keine Ausgaben aus dem internen Logging angezeigt, sondern lediglich die Update-Aktivität.

-q

Deaktiviert die die Konsolenausgabe (Quiet-Modus).

<configfile>

Angabe der Konfigurationsdatei (UTF-8 ohne BOM).

Beispiel:

```
ipy.exe exchange_display_updater.py config_example.json -i 5 -u required
```

Wenn das IronPython-Skript gestartet wird, gibt es zunächst den Exchange Server aus. Dann werden in einer Schleife immer wieder die Meetingraum-Informationen bei dem Exchange Server angefragt, daraus die Display-Inhalte generiert und dies je nach Update-Verfahren an den Wireless ePaper Server gesendet.

Mit STRG + C werden die Updates abgebrochen und es wird eine Statistik auf der Konsole ausgegeben. Folgend ein Beispiel mit der Erläuterung der einzelnen Punkte:

```
updates (new) : 2
updates (repeated) : 3
updates finished successfully : 2
skipped : 11
```

updates (new)

An den Wireless ePaper Server gesendete Updates mit neuen Information.

updates (repeated)

An den Wireless ePaper Server gesendete Wiederholungen von Updates nachdem das vorherige Update nicht erfolgreich war (nur bei gesetztem Argument `-u required`).

updates finished successfully

Als erfolgreich bestätigte Updates (nur bei gesetztem Argument `-u required`).

skipped

Wegen Mangels neuer Informationen oder noch laufendem Update im Status WAITING wurde das Update unterdrückt.

5 Template für Raumbeschilderung

Das Template ist als XSL-Datei auf dem Wireless ePaper Server abgelegt. Es gibt das Format vor, wie die von der Skriptbibliothek übermittelten Updates vom Wireless ePaper Server gerendert werden. Das resultierende Bild wird dann an die entsprechenden Wireless ePaper Displays übermittelt.

5.1 Beispiel eines Templates

Dieses XSL-Template zeigt eine mögliche Beschilderung von Meetingräumen am Beispiel der LANCOM Systems GmbH. Die Zeilennummerierung ist nicht Bestandteil des Codes und dient lediglich zur Verbesserung der Übersichtlichkeit.

```

001 <?xml version="1.0" encoding="UTF-8"?>
002 <!--Das Template ist für 7,4" Displays vorgesehen, die in horizontaler Ausrichtung
angebracht sind.-->
003 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
004
005 <xsl:template match="Record">
006
007 <!-- Render-Informationen für 7,4" Displays -->
008 <image height="480" width="800" rotation="90" font-family="Verdana">
009
010 <!-- Room -->
011 <field height="108" width="780" x="10" y="20">
012 <text align="center" font-size="40" font-weight="bold">
013 <utils method="toUpperCase">
014 <xsl:value-of select="room/@roomName"/>
015 </utils>
016 </text>
017
018 <!-- Date -->
019 <text align="center" font-size="35" font-weight="bold" padding-top="10">
020 <xsl:value-of select="room/field[@key='date']/@value"/>
021 </text>
022 </field>
023
024 <line thickness="2" x-from="0" x-to="800" y-from="130" y-to="130"/>
025
026 <!-- Time1 -->
027 <field height="50" width="780" x="20" y="150">
028 <text align="left" font-weight="bold" font-size="40">
029 <xsl:value-of select="room/field[@key='time1']/@value"/>
030 </text>
031 </field>
032
033 <!-- Purpose1 -->
034 <field height="50" width="780" x="20" y="200">
035 <text align="left" font-size="40" condense="1, 0.8, 0.6, 0.5">
036 <xsl:value-of select="room/field[@key='purpose1']/@value"/>
037 </text>
038 </field>
039
040 <!-- Chair1 -->
041 <field height="40" width="770" x="20" y="250">
042 <text align="left" font-weight="bold" font-size="30">
043 <xsl:value-of select="room/field[@key='chair1']/@value"/>
044 </text>

```

```

045 </field>
046
047 <!-- Time2 -->
048 <field height="35" width="780" x="20" y="320">
049 <text align="left" font-size="28">
050 <xsl:value-of select="room/field[@key='time2']/@value"/>
051 </text>
052 </field>
053
054 <!-- Purpose2 -->
055 <field height="35" width="780" x="20" y="355">
056 <text align="left" font-size="28" condense="1, 0.8, 0.6, 0.5">
057 <xsl:value-of select="room/field[@key='purpose2']/@value"/>
058 </text>
059 </field>
060
061 <!-- Chair2 -->
062 <field height="30" width="770" x="20" y="390">
063 <text align="left" font-size="20">
064 <xsl:value-of select="room/field[@key='chair2']/@value"/>
065 </text>
066 </field>
067
068 <!-- LANCOM Logo -->
069 <field align="right" height="60" width="780" x="10" y="410">
070 </img>
071 </field>
072 </image>
073 </xsl:template>
074 </xsl:stylesheet>

```

5.2 Erläuterungen einzelner Codeabschnitte

In diesem Kapitel werden einzelne Code-Abschnitte des gezeigten XSL-Templates erläutert.

Display-Informationen

```

007 <!-- Render-Informationen für 7,4" Displays -->
008 <image height="480" width="800" rotation="90" font-family="Verdana">

```

Hier wird definiert, für welches Wireless ePaper Display der darauf folgende Code-Abschnitt gültig ist. In diesem Fall handelt es sich um ein 7,4"-Display mit einer Auflösung von 800 x 480 Pixeln.

`height` bestimmt die Höhe des Bildes in Pixeln.

`width` bestimmt die Breite des Bildes in Pixeln.

`rotation` bestimmt die Rotation des Bildes in Grad. 0 entspricht der vertikalen Montierung und 90 der horizontalen Montierung.

`font-family` bestimmt die genutzte Schriftart.

Definition von Textfeldern

```

010 <!-- Room -->
011 <field height="108" width="780" x="10" y="20">
012 <text align="center" font-size="40" font-weight="bold">
013 <utils method="toUpperCase">
014 <xsl:value-of select="room/@roomName"/>
015 </utils>
016 </text>

```

5 Template für Raumbeschilderung

```

017
018 <!-- Date -->
019 <text align="center" font-size="35" font-weight="bold" padding-top="10">
020 <xsl:value-of select="room/field[@key='date']/@value"/>
021 </text>
022 </field>

```

In diesem Bereich wird die Positionierung der Raumbezeichnung und des Datums definiert.

Hierzu wird zuerst ein Feld mit Höhe, Breite und Positionierung auf dem Display definiert:

```
<field height="108" width="780" x="10" y="20">.
```

`height` bestimmt die Höhe des Feldes in Pixeln.

`width` bestimmt die Breite des Feldes in Pixeln.

`x` bestimmt den Abstand zum linken Rand des Displays in Pixeln.

`y` bestimmt den Abstand zum oberen Rand des Displays in Pixeln.

Innerhalb dieses Feldes wird das Layout der verschiedenen Texte definiert:

```
<text align="center" font-size="40" font-weight="bold">
```

`align` bestimmt, wie der Text angeordnet ist. (z.B. center, left, right).

`font-size` bestimmt die Schriftgröße.

`font-weight` bestimmt den Schriftschnitt (z.B. bold, italic).

```
<utils method="toUpperCase">
```

Der Aufruf dieser Methode erzwingt die Darstellung des Textes in Großbuchstaben.

```
<xsl:value-of select="room/@roomName"/>
```

Hier wird auf die von der Skriptbibliothek übermittelten XML-Informationen zugegriffen und der Text an der entsprechenden Stelle eingesetzt. Die bisherigen Definitionen haben nur das Layout vorgegeben.

Daraufhin wird nach gleichem Prinzip der Text definiert, in dem das übermittelte Datum angezeigt wird. Weitere Felder enthalten das Layout für die restlichen übermittelten Informationen für die einzelnen Meetings.

Einbindung von Grafiken

```

068 <!-- LANCOM Logo -->
069 <field align="right" height="60" width="780" x="10" y="410">
070 </img>
071 </field>

```

Zusätzlich zu Textfeldern können auch Grafiken eingebunden werden. Hier wird wie bereits oben zunächst ein Feld definiert, um die Position der Grafik festzulegen.

```
</img>
```

Zusätzlich wird nun hiermit der Speicherort der Grafik und deren Dateiname angegeben.

5.3 Interaktion von Template, XML-Informationen und Wireless ePaper Displays

Die Interaktion von Template, XML-Informationen und der finalen Darstellung der Wireless ePaper Displays wird in diesem Kapitel exemplarisch gezeigt.

Schritt 1:

Wird durch den Notes-Agenten ein Update ausgelöst, enthält der Wireless ePaper Server XML-Informationen mit folgendem Inhalt: (Die Zeilennummerierung ist nicht Bestandteil des Codes und dient lediglich zur Verbesserung der Übersichtlichkeit.)

```
001 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
002 <TaskOrder title="Refresh D1001BF6 for Aachen B2.100">
003   <TemplateTask labelId="D1001BF6" externalId="4711"
004     template="lcsconference_landscape.xsl">
005     <room roomName="Aachen B2.100">
006       <field key="date" value="16.09.2014"/>
007       <field key="time1" value="10:00 - 11:30"/>
008       <field key="purpose1" value="Projektkoordination Marketing"/>
009       <field key="chair1" value="Helmut Müller"/>
010       <field key="time2" value="11:30 - 13.00"/>
011       <field key="purpose2" value="Team-Besprechung Controlling"/>
012       <field key="chair2" value="Karin Klein"/>
013     </room>
014   </TemplateTask>
015 </TaskOrder>
```

Schritt 2:

Der Wireless ePaper Server wertet die erhaltenen Informationen Zeile für Zeile aus. So werden das betroffene Display und das benötigte Template bestimmt. Anhand des Templates erstellt der Server nun den Datensatz, aus dem das spätere Bild gerendert wird. Das Layout wird durch das Template bestimmt und um die Informationen aus dem XML ergänzt.

Zu Illustrationszwecken werden zwei Inhaltselemente erläutert: die Leitung des Folgemeetings und das Firmenlogo.

Die Leitung des Folgemeetings wird laut Template wie folgt formatiert:

```
061 <!-- Chair2 -->
062 <field height="30" width="770" x="20" y="390">
063   <text align="left" font-size="20">
064     <xsl:value-of select="room/field[@key='chair2']/@value"/>
065   </text>
066 </field>
```

Die notwendige Information für die Darstellung findet sich im XML unter:

```
011 <field key="chair2" value="Karin Klein"/>
```

Für die im Template hinterlegte Grafik muss nicht auf die XML-Informationen zugegriffen werden. Stattdessen wird die Pfadangabe ausgewertet.

```
068 <!-- LANCOM Logo -->
069 <field align="right" height="60" width="780" x="10" y="410">
070   </img>
071 </field>
```

Schritt 3:

Rendern des Bildes und anschließendes Senden an das Wireless ePaper Display. Das Resultat ist die folgende Darstellung, farblich hervorgehoben sind die Leitung des Folgemeetings (1) und die Grafik (2).

