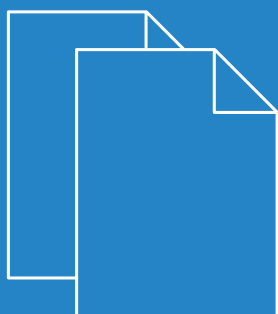


# LANCOM Wireless ePaper Server

## Integration IBM Notes



# Inhalt

<b>1 Zielsetzung und Voraussetzungen.....</b>	<b>4</b>
1.1 Zielsetzung.....	4
1.2 Voraussetzungen.....	4
<b>2 Skriptbibliothek IBM Notes.....</b>	<b>6</b>
2.1 Beispiel einer Skriptbibliothek IBM Notes.....	6
2.2 Erläuterungen einzelner Code-Abschnitte.....	10
<b>3 Template für Raumbeschilderung.....</b>	<b>18</b>
3.1 Beispiel eines Templates.....	18
3.2 Erläuterungen einzelner Codeabschnitte.....	19
3.3 Interaktion von Template, XML-Informationen und Wireless ePaper Displays.....	20

## 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](http://www.openssl.org)).

Produkte von LANCOM Systems enthalten kryptographische Software, die von Eric Young ([eay@cryptsoft.com](mailto: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](http://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 Implementierung der LANCOM Wireless ePaper Solutions in IBM Notes umgesetzt werden kann. Dies erfolgt anhand einer IBM Notes Skriptbibliothek und eines XSL-Templates für Raumbeschilderung. Bei den Codes der Skriptbibliothek und des Templates handelt es sich um Beispiele, welche die Erstellung eigener Dateien vereinfachen soll. Bei der Übernahme des Codes müssen der Einsatzumgebung entsprechend Anpassungen vorgenommen werden.

## 1.2 Voraussetzungen

Die erfolgreiche Implementierung der LANCOM Wireless ePaper Solutions in einer IBM Notes-Umgebung erfordert gewisse Voraussetzungen an vorhandener Software und deren Anpassung.

### 1. Software

Folgende Software ist erforderlich:

- > LANCOM Wireless ePaper Server
- > IBM Notes 8.5 oder höher
- > IBM Notes Designer 8.53 oder höher

### 2. IBM Notes

Notes muss bereits zur Ressourcenreservierung (Meetingräume) eingerichtet sein. Zusätzlich müssen folgende Punkte in der Notes-Datenbank zur Ressourcenreservierung erfüllt sein:

#### > Skriptbibliothek

Eine Skriptbibliothek, wie sie in diesem Dokument erklärt wird (siehe Kapitel 2).

#### > Maske

Eine Maske im Notes-Format „profile document“, welche die Konstante ROOMNR<Raum> aus der Skriptbibliothek für jeden Meetingraum in Form eines Textfeldes enthalten muss. Der Name muss in der Skriptbibliothek entsprechend angepasst werden (siehe Erläuterungen zur Skriptbibliothek in Kapitel 2).

#### > Ansicht

Pro Meetingraum eine Ansicht, die die aktuelle und alle noch kommenden Reservierungen des entsprechenden Raumes für den aktuellen Tag anzeigt (ROOMVIEW<Raum> aus Skriptbibliothek). Der Name der Ansicht muss in der Skriptbibliothek pro Meetingraum angepasst werden (siehe Erläuterungen zur Skriptbibliothek in Kapitel 2).

#### > Agent

Einen Agenten, der die Funktion CheckForUpdate aus der Skriptbibliothek aufruft. Ausführungszeitraum und -intervall sollten nach Bedarf definiert werden (siehe Erläuterungen zur Skriptbibliothek in Kapitel 2).

### 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.
- Adresse (Name oder IP) des Servers müssen in der Konstante UPDATEURLEPAPERSEVER in der Skriptbibliothek entsprechend eingetragen werden (siehe Erläuterungen zur Skriptbibliothek in Kapitel 2).
- Bilder, auf die im Template verwiesen wird, müssen unter \\<Installationsverzeichnis>\data\images\ abgelegt werden.

## 2 Skriptbibliothek IBM Notes

Die Skriptbibliothek enthält den Programm-Code, der die relevanten Informationen der Ressourcenverwaltung abfragt, verarbeitet und bei Bedarf ein Update an den LANCOM Wireless ePaper Server sendet.

### 2.1 Beispiel einer Skriptbibliothek IBM Notes

Für einen Überblick ist der vollständige Code der auf der LANCOM Website zum Download bereitgestellten Skriptdatei unten aufgeführt. Eine Erklärung einzelner Code-Abschnitte findet sich im Anschluss. Die Zeilennummerierung ist nicht Bestandteil des Codes und dient lediglich zur Verbesserung der Übersichtlichkeit.

```
001 %REM
002 Skriptbibliothek für die Ressourcenverwaltung in Notes
003 Created 08.10.2014 by LANCOM Systems GmbH
004 %END REM
005
006 Option Public
007
008 '# Deklaration von Konstanten
009 Private Const ROOMIDAACHEN = "D1001BF6"
010 Private Const ROOMNAMEAACHEN = "Aachen B2.100"
011 Private Const ROOMNRAACHEN = "B2100"
012 Private Const ROOMVIEWAACHEN = "CurResForB2100"
013 Private Const UPDATEURLEPAPERSERVER = "http://epaper-server:8001/service/task"
014 Private Const DEBUGEMAIL = "debugging@yourcompany.com"
015 Private Const PROFILEDOC = "yourProfileDoc"
016 Private Const NOMOREMEETINGS = "keine weiteren Reservierungen"
017 Private Const TEMPLATE = "lcsconference_landscape.xsl"
018
019
020 '# Erzeugen einer Unique-ID
021 Function createGuid()
022 Dim object
023 Set object = CreateObject("Scriptlet.TypeLib")
024 guid = object.Guid
025 Set object = Nothing
026
027 createGuid = Left(guid, Len(guid)-2)
028 End Function
029
030
031 '# Behandlung von XML-Steuerzeichen in Strings
032 Function EscapeXML(newStr As String) As String
033
034 Dim replaceFrom (1 To 5) As String
035 Dim replaceTo (1 To 5) As String
036 Dim escapedString As String
037
038 replaceFrom(1) = |&|
039 replaceFrom(2) = |"|
040 replaceFrom(3) = |'|
041 replaceFrom(4) = |<|
042 replaceFrom(5) = |>|
043
044 replaceTo(1) = |&amp;|
045 replaceTo(2) = |&quot;|
```

```
046 replaceTo(3) = |&apos;|
047 replaceTo(4) = |&lt;|
048 replaceTo(5) = |&gt;|
049
050 EscapeXML = Replace(newStr,replaceFrom,replaceTo)
051
052 End Function
053
054
055 '# Funktion zum Display Update
056 Function UpdateDisplay(meetingUNID As String, meetingUNID2 As String, displayID As
String, resourceName As String, roomNr As String) As String
057
058 Dim session As New NotesSession
059 Dim db As NotesDatabase
060 Set db = session.CurrentDatabase
061
062 Dim profDoc As NotesDocument
063 Set profDoc = db.Getprofiledocument (PROFILEDOC)
064 Dim mailDoc As NotesDocument
065 Dim rtItem As NotesRichTextItem
066
067 Dim objHttp As Variant
068 Dim reqHttp As String
069 Dim url As String
070 Dim response As String
071
072 Dim meetingDoc1 As NotesDocument
073 Dim meetingDoc2 As NotesDocument
074 Dim meetingChair1 As NotesName
075 Dim meetingChair2 As NotesName
076 Dim roomName As NotesName
077
078 Dim dispDate As String
079 Dim dispRoom As String
080 Dim dispTime1 As String
081 Dim dispPurpose1 As String
082 Dim dispChair1 As String
083 Dim dispTime2 As String
084 Dim dispPurpose2 As String
085 Dim dispChair2 As String
086
087 DebugMode = False
088
089 Set roomName = New NotesName(resourceName)
090 dispRoom = roomName.Common
091 dispDate = Format$(Today,"dd.mm.yyyy")
092
093 '# Abfrage der Meetings
094 If meetingUNID <> "" Then
095     Set meetingDoc1 = db.Getdocumentbyunid(meetingUNID)
096     Set meetingChair1 = New NotesName(meetingDoc1.Chair(0))
097     dispTime1 = Format$(meetingDoc1.StartDateTime(0),"hh:mm")+ " - " +
Format$(meetingDoc1.EndDateTime(0),"hh:mm")
098     If meetingDoc1.Hasitem("txtPurpose") Then
099         dispPurpose1 = EscapeXML(FullTrim(meetingDoc1.txtPurpose(0)))
100     Else
101         dispPurpose1 = EscapeXML(Trim(meetingDoc1.Purpose(0)))
102     End If
103
104     dispChair1 = meetingChair1.Common
105
106     If meetingUNID2 <> "" Then
107         Set meetingDoc2 = db.Getdocumentbyunid(meetingUNID2)
```

```

108 Set meetingChair2 = New NotesName(meetingDoc2.Chair(0))
109     dispTime2 = Format$(meetingDoc2.StartDateTime(0),"hh:mm")+ " - " +
Format$(meetingDoc2.EndDateTime(0),"hh:mm")
110 If meetingDoc2.Hasitem("txtPurpose") Then
111     dispPurpose2 = EscapeXML(FullTrim(meetingDoc2.txtPurpose(0)))
112 Else
113     dispPurpose2 = EscapeXML(Trim(meetingDoc2.Purpose(0)))
114 End If
115
116     dispChair2 = meetingChair2.Common
117 End If
118 Else
119     dispPurpose1 = NOMOREMEETINGS
120 End If
121
122 Set mailDoc = db.CreateDocument
123 mailDoc.Form = "Memo"
124 mailDoc.Subject = "Report UpdateDisplay for " + resourceName + " in Ressource.nsf vom
" + Format$(Now)
125 mailDoc.SendTo = DEBUGEMAIL
126 Set rtItem = mailDoc.CreateRichTextItem("Body")
127
128 '# XML-Erstellung
129 reqHttp = |<?xml version="1.0" encoding="UTF-8" standalone="yes"?>|+_
130     |<TaskOrder title="Refresh label |+displayID+| for room |+dispRoom+|">|+_
131     |<TemplateTask labelId=" |+displayID+|" externalId="4711" template=" |+TEMPLATE|>|+_
132     |<room roomName=" |+dispRoom+|">|+_
133         |<field key="date" value=" |+dispDate+|" />|+_
134         |<field key="time1" value=" |+dispTime1+|" />|+_
135         |<field key="purpose1" value=" |+dispPurpose1+|" />|+_
136         |<field key="chair1" value=" |+dispChair1+|" />|+_
137         |<field key="time2" value=" |+dispTime2+|" />|+_
138         |<field key="purpose2" value=" |+dispPurpose2+|" />|+_
139         |<field key="chair2" value=" |+dispChair2+|" />|+_
140     |</room>|+_
141     |</TemplateTask>|+_
142     |</TaskOrder>|
143
144
145 Dim oldVal As Variant
146 oldval = profDoc.GetItemvalue(roomNr)
147
148 '# Prüfung, ob Update notwendig ist.
149 If StrComp(reqHttp,oldVal(0)) Then
150
151     Call profdoc.Replaceitemvalue(roomNr, Nothing)
152     Call profDoc.Replaceitemvalue(roomNr, reqHttp)
153     Call profDoc.Save(True, False)
154
155     Set objHttp = CreateObject("Microsoft.XMLHTTP")
156     url = UPDATEURLEPAPERSEVER
157     objHttp.Open "POST", url, False, "", ""
158     objHttp.setRequestHeader "Content-Type:", "application/xml"
159     objHttp.setRequestHeader "Content-Lenght:", Len(reqHttp)
160     objHttp.setRequestHeader "User-Agent:", "LCS_NTS_AGNT"
161     objHttp.Send(reqHttp)
162
163     If (objHttp.readyState <> 4) Or (objHttp.status <> 200) Then
164         Call rtItem.AppendText ("Der Request konnte nicht ausgeführt
werden!" + Chr(13) + Chr(10) +_
165         "für Meeting UNID 1: " + meetingUNID + Chr(13) + Chr(10) +_
166         "für Meeting UNID 2: " + meetingUNID2 + Chr(13) + Chr(10) +_
167         "readyState: " + Format$(objHttp.readyState) + Chr(13) + Chr(10) +_
168         "HTTP Status: " + Format$(objHttp.status) + Chr(13) + Chr(10) +_

```



```
169 "Request: " +Chr(13)+Chr(10)+ reqHttp +Chr(13)+Chr(10)+Chr(13)+Chr(10)+_
170 "Response: " +Chr(13)+Chr(10)+ objHttp.responseText)
171 Call mailDoc.Send(False)
172 Else
173 Call rtItem.AppendText ("works fine! "+Chr(13)+Chr(10)+_
174 "für Meeting UNID 1: " +meetingUNID +Chr(13)+Chr(10)+_
175 "für Meeting UNID 2: " +meetingUNID2 +Chr(13)+Chr(10)+_
176 "readyState: " +Format$(objHttp.readyState)+Chr(13)+Chr(10)+_
177 "HTTP Status: " +Format$(objHttp.status)+Chr(13)+Chr(10)+_
178 "Request: " +Chr(13)+Chr(10)+ reqHttp +Chr(13)+Chr(10)+Chr(13)+Chr(10)+_
179 "Response: " +Chr(13)+Chr(10)+ objHttp.responseText)
180 Call mailDoc.Send(False)
181 End If
182
183 response = objHttp.responseText
184 Set objHttp=Nothing
185
186 ElseIf DebugMode Then
187 Call rtItem.AppendText ("same Request. No Update! "+Chr(13)+Chr(10) +"Request: " +
reqHttp)
188 Call mailDoc.Send(False)
189 MsgBox("same as last")
190 End If
191
192 UpdateDisplay = response
193
194 End Function
195
196
197 '# Funktion zum Auslösen des Update-Prozesses
198 Sub CheckForUpdate
199
200 Dim session As New NotesSession
201 Dim db As NotesDatabase
202 Set db = session.CurrentDatabase
203
204 Dim meetingView As NotesView
205
206 Dim response As String
207 Dim meetingUNID1 As String
208 Dim meetingUNID2 As String
209
210
211 '# Beispiel für Raum AACHEN, pro Raum notwendig
212 Set meetingView = db.GetView(ROOMVIEWAACHEN)
213
214 If meetingView.Allentries.Count > 1 Then
215 meetingUNID1 = meetingView.Getfirstdocument().Universalid
216 meetingUNID2 = meetingView.Getnthdocument(2).Universalid
217 ElseIf meetingView.Allentries.Count = 1 Then
218 meetingUNID1 = meetingView.Getfirstdocument().Universalid
219 meetingUNID2 = ""
220 Else
221 meetingUNID1 = ""
222 meetingUNID2 = ""
223 End If
224
225 response = UpdateDisplay(meetingUNID1, meetingUNID2,ROOMIDAACHEN, ROOMNAMEAACHEN,
ROOMNRAACHEN)
226
227 End Sub
```

## 2.2 Erläuterungen einzelner Code-Abschnitte

In diesem Kapitel werden einzelne Code-Abschnitte erläutert, um den Umgang mit der Skriptbibliothek zu vereinfachen.

### Deklaration von Konstanten

In diesem Code-Abschnitt werden Konstanten deklariert, um sie einfacher ändern zu können und um einen besseren Überblick zu haben.

```
008 '# Deklaration von Konstanten
009 Private Const ROOMIDAACHEN = "D1001BF6"
010 Private Const ROOMNAMEAACHEN = "Aachen B2.100"
011 Private Const ROOMNRAACHEN = "B2100"
012 Private Const ROOMVIEWAACHEN = "CurResForB2100"
013 Private Const UPDATEURLEPAPERSERVER = "http://epaper-server:8001/service/task"
014 Private Const DEBUGEMAIL = "debugging@yourcompany.com"
015 Private Const PROFILEDOC = "yourProfileDoc"
016 Private Const NOMOREMEETINGS = "keine weiteren Reservierungen"
017 Private Const TEMPLATE = "lcsconference_landscape.xsl"
```

```
Private Const ROOMIDAACHEN
```

Die ID des Wireless ePaper Displays.

```
Private Const ROOMNAMEAACHEN
```

Die Bezeichnung des Wireless ePaper Displays.

```
Die Private Const ROOMNRAACHEN
```

ID des Raumes. Sie wird im Profile-Dokument genutzt, um das Textfeld zu benennen, in dem der zuletzt übertragene Datensatz abgelegt wird.

```
Private Const ROOMVIEWAACHEN
```

Der Name der Notes-Ansicht, in der die aktuelle und noch kommenden Reservierungen des Raumes angezeigt werden.

```
Private Const UPDATEURLEPAPERSERVER
```

Die Update-URL über die der LANCOM Wireless ePaper Server angesprochen wird. Schreibweise: "http://<Name oder IP>:<Port>/service/task"

```
Private Const DEBUGEMAIL
```

Eine Email-Adresse, die für Debugging genutzt wird (durchgeführte Updates, etc.).

```
Private Const PROFILEDOC
```

Der Name des Profile-Dokumentes, in dem die zuletzt übertragenen Datensätze für alle Räume abgelegt werden.

```
Private Const NOMOREMEETINGS
```

Der auf dem Display darzustellende Text, wenn keine weiteren Reservierungen für den Tag vorhanden sind.

```
Private Const TEMPLATE
```

Der Dateiname des Templates für die Darstellung der Reservierungen auf den Wireless ePaper Displays.

### Funktion: Erzeugen einer Unique-ID

```
020 '# Erzeugen einer Unique-ID
021 Function createGuid()
022 Dim object
023 Set object = CreateObject("Scriptlet.TypeLib")
024 guid = object.Guid
025 Set object = Nothing
026
```

```
027 createGuid = Left(guid, Len(guid)-2)
028 End Function
```

Mit dieser Funktion wird eine Unique-ID erzeugt, die im Verlauf des Prozesses benötigt wird.

### Funktion: Behandlung von XML-Steuerzeichen in Strings

```
031 '# Behandlung von XML-Steuerzeichen in Strings
032 Function EscapeXML(newStr As String) As String
033
034 Dim replaceFrom (1 To 5) As String
035 Dim replaceTo (1 To 5) As String
036 Dim escapedString As String
037
038 replaceFrom(1) = |&|
039 replaceFrom(2) = |"|
040 replaceFrom(3) = |'|
041 replaceFrom(4) = |<|
042 replaceFrom(5) = |>|
043
044 replaceTo(1) = |&amp;|
045 replaceTo(2) = |&quot;|
046 replaceTo(3) = |&apos;|
047 replaceTo(4) = |&lt;|
048 replaceTo(5) = |&gt;|
049
050 EscapeXML = Replace(newStr,replaceFrom,replaceTo)
051
052 End Function
```

Diese Funktion wird genutzt, um bei Strings (zum Beispiel dem Betreff der Reservierung), die auf dem Display angezeigt werden sollen die enthaltenen XML-Steuerzeichen so umzusetzen, dass sie korrekt auf dem Display dargestellt werden und keine Probleme im XML verursachen.

### Funktion: UpdateDisplay

Auf Grund des Umfangs dieser Funktion wird diese hier in ihrer Gesamtheit dargestellt und nachfolgend einzeln erläutert.

```
055 '# Funktion zum Display Update
056 Function UpdateDisplay(meetingUNID As String, meetingUNID2 As String, displayID As
String, resourceName As String, roomNr As String) As String
057
058 Dim session As New NotesSession
059 Dim db As NotesDatabase
060 Set db = session.CurrentDatabase
061
062 Dim profDoc As NotesDocument
063 Set profDoc = db.Getprofiledocument(PROFILEDOC)
064 Dim mailDoc As NotesDocument
065 Dim rtItem As NotesRichTextItem
066
067 Dim objHttp As Variant
068 Dim reqHttp As String
069 Dim url As String
070 Dim response As String
071
072 Dim meetingDoc1 As NotesDocument
073 Dim meetingDoc2 As NotesDocument
074 Dim meetingChair1 As NotesName
075 Dim meetingChair2 As NotesName
076 Dim roomName As NotesName
077
078 Dim dispDate As String
079 Dim dispRoom As String
080 Dim dispTime1 As String
```

```

081 Dim dispPurpose1 As String
082 Dim dispChair1 As String
083 Dim dispTime2 As String
084 Dim dispPurpose2 As String
085 Dim dispChair2 As String
086
087 DebugMode = False
088
089 Set roomName = New NotesName(resourceName)
090 dispRoom = roomName.Common
091 dispDate = Format$(Today,"dd.mm.yyyy")
092
093 '# Abfrage der Meetings
094 If meetingUNID <> "" Then
095     Set meetingDoc1 = db.Getdocumentbyunid(meetingUNID)
096     Set meetingChair1 = New NotesName(meetingDoc1.Chair(0))
097     dispTime1 = Format$(meetingDoc1.StartDateTime(0),"hh:mm")+ " - " +
Format$(meetingDoc1.EndDateTime(0),"hh:mm")
098     If meetingDoc1.Hasitem("txtPurpose") Then
099         dispPurpose1 = EscapeXML(FullTrim(meetingDoc1.txtPurpose(0)))
100     Else
101         dispPurpose1 = EscapeXML(Trim(meetingDoc1.Purpose(0)))
102     End If
103
104     dispChair1 = meetingChair1.Common
105
106     If meetingUNID2 <> "" Then
107         Set meetingDoc2 = db.Getdocumentbyunid(meetingUNID2)
108         Set meetingChair2 = New NotesName(meetingDoc2.Chair(0))
109         dispTime2 = Format$(meetingDoc2.StartDateTime(0),"hh:mm")+ " - " +
Format$(meetingDoc2.EndDateTime(0),"hh:mm")
110         If meetingDoc2.Hasitem("txtPurpose") Then
111             dispPurpose2 = EscapeXML(FullTrim(meetingDoc2.txtPurpose(0)))
112         Else
113             dispPurpose2 = EscapeXML(Trim(meetingDoc2.Purpose(0)))
114         End If
115
116         dispChair2 = meetingChair2.Common
117     End If
118 Else
119     dispPurpose1 = NOMOREMEETINGS
120 End If
121
122 Set mailDoc = db.CreateDocument
123 mailDoc.Form = "Memo"
124 mailDoc.Subject = "Report UpdateDisplay for "+ resourceName +" in Ressource.nsf vom
" + Format$(Now)
125 mailDoc.SendTo = DEBUGEMAIL
126 Set rtItem = mailDoc.CreateRichTextItem("Body")
127
128 '# XML-Erstellung
129 reqHttp = |<?xml version="1.0" encoding="UTF-8" standalone="yes"?>|+_
130 |<TaskOrder title="Refresh label |+displayID +| for room |+dispRoom+|">|+_
131 |<TemplateTask labelId="|+displayID +|" externalId="4711" template=|+TEMPLATE|>|+_
132 |<room roomName="|+dispRoom+|">|+_
133 |<field key="date" value="|+dispDate+|"/>|+_
134 |<field key="time1" value="|+dispTime1 +|"/>|+_
135 |<field key="purpose1" value="|+dispPurpose1 +|"/>|+_
136 |<field key="chair1" value="|+dispChair1 +|"/>|+_
137 |<field key="time2" value="|+dispTime2 +|"/>|+_
138 |<field key="purpose2" value="|+dispPurpose2 +|"/>|+_
139 |<field key="chair2" value="|+dispChair2 +|"/>|+_
140 |</room>|+_
141 |</TemplateTask>|+_

```

```

142     |</TaskOrder>|
143
144
145 Dim oldVal As Variant
146 oldval = profDoc.GetItemvalue(roomNr)
147
148 '# Prüfung, ob Update notwendig ist.
149 If StrComp(reqHttp,oldVal(0)) Then
150
151     Call profdoc.Replaceitemvalue(roomNr, Nothing)
152     Call profDoc.Replaceitemvalue(roomNr, reqHttp)
153     Call profDoc.Save(True, False)
154
155     Set objHttp = CreateObject("Microsoft.XMLHTTP")
156     url = UPDATEURLEPAPERSEVER
157     objHttp.Open "POST", url, False, "", ""
158     objHttp.setRequestHeader "Content-Type:", "application/xml"
159     objHttp.setRequestHeader "Content-Lenght:", Len(reqHttp)
160     objHttp.setRequestHeader "User-Agent:", "LCS_NTS_AGNT"
161     objHttp.Send(reqHttp)
162
163     If (objHttp.readyState <> 4) Or (objHttp.status <> 200) Then
164         Call rtItem.AppendText ("Der Request konnte nicht ausgeführt
werden!" + Chr(13) + Chr(10) + _
165     "für Meeting UNID 1: " + meetingUNID + Chr(13) + Chr(10) + _
166     "für Meeting UNID 2: " + meetingUNID2 + Chr(13) + Chr(10) + _
167     "readyState: " + Format$(objHttp.readyState) + Chr(13) + Chr(10) + _
168     "HTTP Status: " + Format$(objHttp.status) + Chr(13) + Chr(10) + _
169     "Request: " + Chr(13) + Chr(10) + reqHttp + Chr(13) + Chr(10) + Chr(13) + Chr(10) + _
170     "Response: " + Chr(13) + Chr(10) + objHttp.responseText)
171     Call mailDoc.Send(False)
172     Else
173     Call rtItem.AppendText ("works fine! " + Chr(13) + Chr(10) + _
174     "für Meeting UNID 1: " + meetingUNID + Chr(13) + Chr(10) + _
175     "für Meeting UNID 2: " + meetingUNID2 + Chr(13) + Chr(10) + _
176     "readyState: " + Format$(objHttp.readyState) + Chr(13) + Chr(10) + _
177     "HTTP Status: " + Format$(objHttp.status) + Chr(13) + Chr(10) + _
178     "Request: " + Chr(13) + Chr(10) + reqHttp + Chr(13) + Chr(10) + Chr(13) + Chr(10) + _
179     "Response: " + Chr(13) + Chr(10) + objHttp.responseText)
180     Call mailDoc.Send(False)
181     End If
182
183     response = objHttp.responseText
184     Set objHttp=Nothing
185
186 ElseIf DebugMode Then
187     Call rtItem.AppendText ("same Request. No Update! " + Chr(13) + Chr(10) + "Request: " +
reqHttp)
188     Call mailDoc.Send(False)
189     MsgBox("same as last")
190 End If
191
192 UpdateDisplay = response
193
194 End Function

```

### Einzelne Code-Abschnitte der Funktion: UpdateDisplay

#### Deklaration von Variablen

```

058 Dim session As New NotesSession
059 Dim db As NotesDatabase
060 Set db = session.CurrentDatabase

```

```

061
062 Dim profDoc As NotesDocument
063 Set profDoc = db.Getprofiledocument (PROFILEDOC)
064 Dim mailDoc As NotesDocument
065 Dim rtItem As NotesRichTextItem
066
067 Dim objHttp As Variant
068 Dim reqHttp As String
069 Dim url As String
070 Dim response As String
071
072 Dim meetingDoc1 As NotesDocument
073 Dim meetingDoc2 As NotesDocument
074 Dim meetingChair1 As NotesName
075 Dim meetingChair2 As NotesName
076 Dim roomName As NotesName
077
078 Dim dispDate As String
079 Dim dispRoom As String
080 Dim dispTime1 As String
081 Dim dispPurpose1 As String
082 Dim dispChair1 As String
083 Dim dispTime2 As String
084 Dim dispPurpose2 As String
085 Dim dispChair2 As String

```

In diesem Code-Abschnitt werden die verschiedenen Variablen deklariert, die von der Funktion genutzt werden. Dies umfasst die Variablen, in denen die Informationen der einzelnen Meetings gespeichert werden, als auch das Profile-Dokument und diverser anderer Variablen die benötigt werden.

#### Abfrage der Reservierungen anhand von UNIDs

```

093 '# Abfrage der Meetings
094 If meetingUNID <> "" Then
095   Set meetingDoc1 = db.Getdocumentbyunid(meetingUNID)
096   Set meetingChair1 = New NotesName(meetingDoc1.Chair(0))
097   dispTime1 = Format$(meetingDoc1.StartDateTime(0), "hh:mm") + " - " +
Format$(meetingDoc1.EndDateTime(0), "hh:mm")
098   If meetingDoc1.Hasitem("txtPurpose") Then
099     dispPurpose1 = EscapeXML(FullTrim(meetingDoc1.txtPurpose(0)))
100   Else
101     dispPurpose1 = EscapeXML(Trim(meetingDoc1.Purpose(0)))
102   End If
103
104   dispChair1 = meetingChair1.Common
105
106   If meetingUNID2 <> "" Then
107     Set meetingDoc2 = db.Getdocumentbyunid(meetingUNID2)
108     Set meetingChair2 = New NotesName(meetingDoc2.Chair(0))
109     dispTime2 = Format$(meetingDoc2.StartDateTime(0), "hh:mm") + " - " +
Format$(meetingDoc2.EndDateTime(0), "hh:mm")
110     If meetingDoc2.Hasitem("txtPurpose") Then
111       dispPurpose2 = EscapeXML(FullTrim(meetingDoc2.txtPurpose(0)))
112     Else
113       dispPurpose2 = EscapeXML(Trim(meetingDoc2.Purpose(0)))
114     End If
115
116     dispChair2 = meetingChair2.Common
117   End If
118 Else
119   dispPurpose1 = NOMOREMEETINGS
120 End If

```

In diesem Code-Abschnitt werden die Informationen der Reservierungen vom Notes-Server abgerufen und die Variablen für die aktuelle und folgende Reservierung gesetzt. Ist keine Folgereservierung oder keine aktuelle Reservierung vorhanden, wird entsprechend der Text der Konstanten NOMOREMEETINGS gesetzt. Damit einhergehend werden die Texte, die als Topic angegeben werden mit der Funktion `EscapeXML` bearbeitet, um sicherzustellen, dass keine XML-Steuerzeichen in den entsprechenden Variablen abgelegt werden.

### E-Mail-Definition

```
122 Set mailDoc = db.CreateDocument
123 mailDoc.Form = "Memo"
124 mailDoc.Subject = "Report UpdateDisplay for "+ resourceName +" in Ressource.nsf vom
" + Format$(Now)
125 mailDoc.SendTo = DEBUGEMAIL
126 Set rtItem = mailDoc.CreateRichTextItem("Body")
```

Dieser Code-Abschnitt definiert den Aufbau einer Email, die bei einem Update oder beim Debugging erzeugt und verschickt werden kann.

### XML-Erstellung

```
128 '# XML-Erstellung
129 reqHttp = |<?xml version="1.0" encoding="UTF-8" standalone="yes"?>|+_
130 |<TaskOrder title="Refresh label |+displayID +| for room |+dispRoom+|">|+_
131 |<TemplateTask labelId="|+displayID +|" externalId="4711" template=|+TEMPLATE|>|+_
132 |<room roomName="|+dispRoom+|">|+_
133 |<field key="date" value="|+dispDate+|" />|+_
134 |<field key="time1" value="|+dispTime1 +|" />|+_
135 |<field key="purpose1" value="|+dispPurpose1 +|" />|+_
136 |<field key="chair1" value="|+dispChair1 +|" />|+_
137 |<field key="time2" value="|+dispTime2 +|" />|+_
138 |<field key="purpose2" value="|+dispPurpose2 +|" />|+_
139 |<field key="chair2" value="|+dispChair2 +|" />|+_
140 |</room>|+_
141 |</TemplateTask>|+_
142 |</TaskOrder>|
```

Hier wird der HTTP-Request erstellt, welcher an den Wireless ePaper Server geschickt wird und die Informationen für die einzelnen Felder des Templates enthält. Der Wert für `externalId`, in diesem Beispiel 4711 kann frei gewählt werden. Sie wird vom Wireless ePaper Server benötigt, um die Quelle des Updates zu identifizieren und in einem Log abzulegen.

### Vergleich des aktuellen mit dem zuletzt übertragenen Datensatz

```
145 Dim oldVal As Variant
146 oldval = profDoc.GetItemvalue(roomNr)
147
148 '# Prüfung, ob Update notwendig ist.
149 If StrComp(reqHttp,oldVal(0)) Then
150
151 Call profdoc.Replaceitemvalue(roomNr, Nothing)
152 Call profDoc.Replaceitemvalue(roomNr, reqHttp)
153 Call profDoc.Save(True, False)
154
155 Set objHttp = CreateObject("Microsoft.XMLHTTP")
156 url = UPDATEURLEPAPERSERVER
157 objHttp.Open "POST", url, False, "", ""
158 objHttp.setRequestHeader "Content-Type:", "application/xml"
159 objHttp.setRequestHeader "Content-Lenght:", Len(reqHttp)
160 objHttp.setRequestHeader "User-Agent:", "LCS_NTS_AGNT"
161 objHttp.Send(reqHttp)
162
163 If (objHttp.readyState <> 4) Or (objHttp.status <> 200) Then
164 Call rtItem.AppendText ("Der Request konnte nicht ausgeführt werden!" + Chr(13) + Chr(10) +
```

```

165 "für Meeting UNID 1: " +meetingUNID +Chr(13)+Chr(10)+_
166 "für Meeting UNID 2: " +meetingUNID2 +Chr(13)+Chr(10)+_
167 "readyState: " +Format$(objHttp.readyState)+Chr(13)+Chr(10)+_
168 "HTTP Status: " +Format$(objHttp.status)+Chr(13)+Chr(10)+_
169 "Request: " +Chr(13)+Chr(10)+ reqHttp +Chr(13)+Chr(10)+Chr(13)+Chr(10)+_
170 "Response: " +Chr(13)+Chr(10)+ objHttp.responseText)
171 Call mailDoc.Send(False)
172 Else
173 Call rtItem.AppendText ("works fine! "+Chr(13)+Chr(10)+_
174 "für Meeting UNID 1: " +meetingUNID +Chr(13)+Chr(10)+_
175 "für Meeting UNID 2: " +meetingUNID2 +Chr(13)+Chr(10)+_
176 "readyState: " +Format$(objHttp.readyState)+Chr(13)+Chr(10)+_
177 "HTTP Status: " +Format$(objHttp.status)+Chr(13)+Chr(10)+_
178 "Request: " +Chr(13)+Chr(10)+ reqHttp +Chr(13)+Chr(10)+Chr(13)+Chr(10)+_
179 "Response: " +Chr(13)+Chr(10)+ objHttp.responseText)
180 Call mailDoc.Send(False)
181 End If
182
183 response = objHttp.responseText
184 Set objHttp=Nothing
185
186 ElseIf DebugMode Then
187 Call rtItem.AppendText ("same Request. No Update! "+Chr(13)+Chr(10) +"Request: " +
reqHttp)
188 Call mailDoc.Send(False)
189 MsgBox("same as last")
190 End If
191
192 UpdateDisplay = response

```

Damit nur Updates mit neuen Informationen an den Wireless ePaper Server und somit an die Displays geschickt werden, wird hier der zuletzt gesendete Datensatz mit gerade erstellten verglichen. Gibt es Unterschiede, wird das Update an den Server geschickt, ist kein Update notwendig wird der Datensatz verworfen. Wird ein Update abgeschickt, wird die Antwort des Servers als Rückgabewert für die Funktion genutzt. Dieser kann dann für Monitoring und Troubleshooting verwendet werden.

### Funktion: Update-Prozess der Skriptbibliothek starten

```

197 '# Funktion zum Auslösen des Update-Prozesses
198 Sub CheckForUpdate
199
200 Dim session As New NotesSession
201 Dim db As NotesDatabase
202 Set db = session.CurrentDatabase
203
204 Dim meetingView As NotesView
205
206 Dim response As String
207 Dim meetingUNID1 As String
208 Dim meetingUNID2 As String
209
210
211 '# Beispiel für Raum AACHEN, pro Raum notwendig
212 Set meetingView = db.GetView(ROOMVIEWAACHEN)
213
214 If meetingView.Allentries.Count > 1 Then
215 meetingUNID1 = meetingView.Getfirstdocument().Universalid
216 meetingUNID2 = meetingView.Getnthdocument(2).Universalid
217 ElseIf meetingView.Allentries.Count = 1 Then
218 meetingUNID1 = meetingView.Getfirstdocument().Universalid
219 meetingUNID2 = ""
220 Else
221 meetingUNID1 = ""
222 meetingUNID2 = ""

```



```
223 End If
224
225 response = UpdateDisplay(meetingUNID1, meetingUNID2,ROOMIDAACHEN, ROOMNAMEAACHEN,
ROOMNRAACHEN)
226
227 End Sub
```

Diese Funktion wird durch den Notes-Agenten aufgerufen, der festlegt in welchen Zeiträumen und Zeitabständen geprüft werden soll, ob ein Update von Displays notwendig ist.

Sollen mehrere Meetingräume (Displays) aktualisiert werden, muss diese Funktion für jeden Meetingraum um den folgenden Code ergänzt werden. Es müssen dabei die Konstanten ROOMVIEWAACHEN, ROOMIDAACHEN, ROOMNAMEAACHEN und ROOMNRAACHEN durch die entsprechenden Konstanten des zusätzlichen Raumes ergänzt werden. Außerdem müssen diese Konstanten zu Beginn der Skriptbibliothek deklariert werden.

```
211 '# Beispiel für Raum AACHEN, pro Raum notwendig
212 Set meetingView = db.Getview(ROOMVIEWAACHEN)
213
214 If meetingView.Allentries.Count > 1 Then
215     meetingUNID1 = meetingView.Getfirstdocument().Universalid
216     meetingUNID2 = meetingView.Getnthdocument(2).Universalid
217 ElseIf meetingView.Allentries.Count = 1 Then
218     meetingUNID1 = meetingView.Getfirstdocument().Universalid
219     meetingUNID2 = ""
220 Else
221     meetingUNID1 = ""
222     meetingUNID2 = ""
223 End If
224
225 response = UpdateDisplay(meetingUNID1, meetingUNID2,ROOMIDAACHEN, ROOMNAMEAACHEN,
ROOMNRAACHEN)
```

## 3 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.

### 3.1 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>

```

## 3.2 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>

```

### 3 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.

## 3.3 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).

