



Tracking Gas Balloons With REXX

An example for rapid deployment of REXX
applications

Presented at the REXX Symposium 2004 by
Christian Michel



Agenda

- Introduction
- Project requirements
- Implemented functions
- Web presentation
- Summary
- Video

Coupe Aéronautique Gordon Bennett





Introduction

- The Coupe Aéronautique Gordon Bennett is a traditional long distance race for gas balloons (since 1906)
- Balloons are flying for up to 3 days and up to 2200 km (distance achieved in 1912!)
- Difficult to pass the suspense to the public
- Tracking would help to involve public

Impressions from previous events



© Christian Michel. Unauthorized copying or publishing prohibited.

Impressions from previous events



© Artur Zonabend. Unauthorized copying or publishing prohibited.

Impressions from previous events



Impressions from previous events



Impressions from previous events



© Jörg Adam. Unauthorized copying or publishing prohibited.

Impressions from previous events



© Jörg Adam. Unauthorized copying or publishing prohibited.

Impressions from previous events



© Jörg Adam. Unauthorized copying or publishing prohibited.

Impressions from previous events



Impressions from previous events



© Philippe De Cock. Unauthorized copying or publishing prohibited.

Impressions from previous events



© Philippe De Cock. Unauthorized copying or publishing prohibited.

Sample map (2002 race)



Sample map (2003 race)



Traditional tracking method

- Not suited for web representation ☺





Electronic tracking

- First test of live tracking using ham radios and APRS in 2001 was unsuccessful:
 - APRS not as widely used in Europe as in the U.S. (APRS is used for American Challenge)
- In 2002 the SatPro company offered their Inmarsat based equipment to be used for the race



Project Requirements

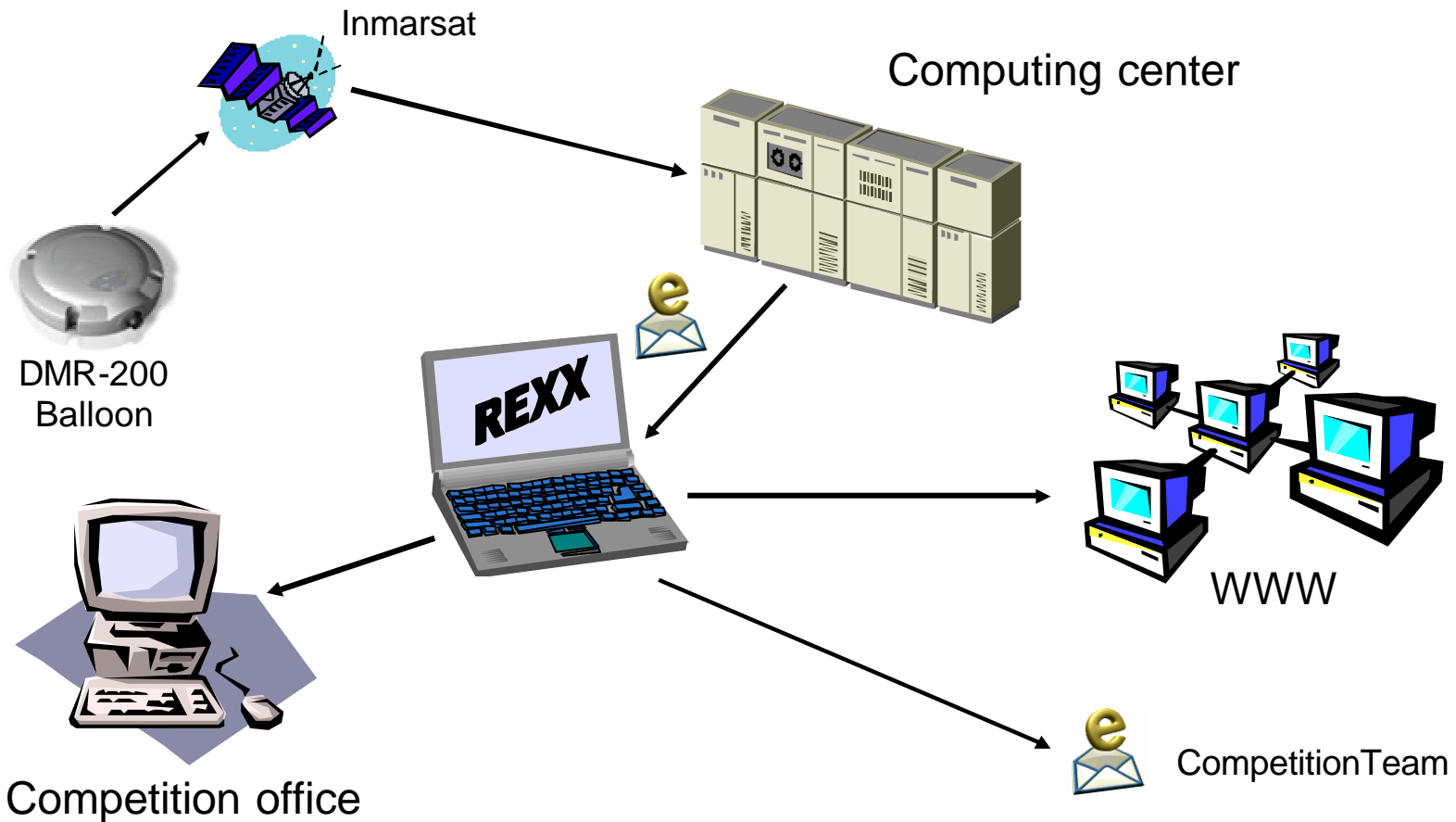
- Implement web tracking within three weeks for a test competition
- Show data with time delay to public (for reducing direct influence on competitors)
- Provide real time access to competition office
- Automate data collection/presentation



Project Fundamentals

- Data provided by SatPro company:
 - Four overview maps with all team positions
 - All position reports via e-mail
- Additional data available:
 - Manually created maps using local tracking software for each individual team

Tracking Infrastructure



Equipment

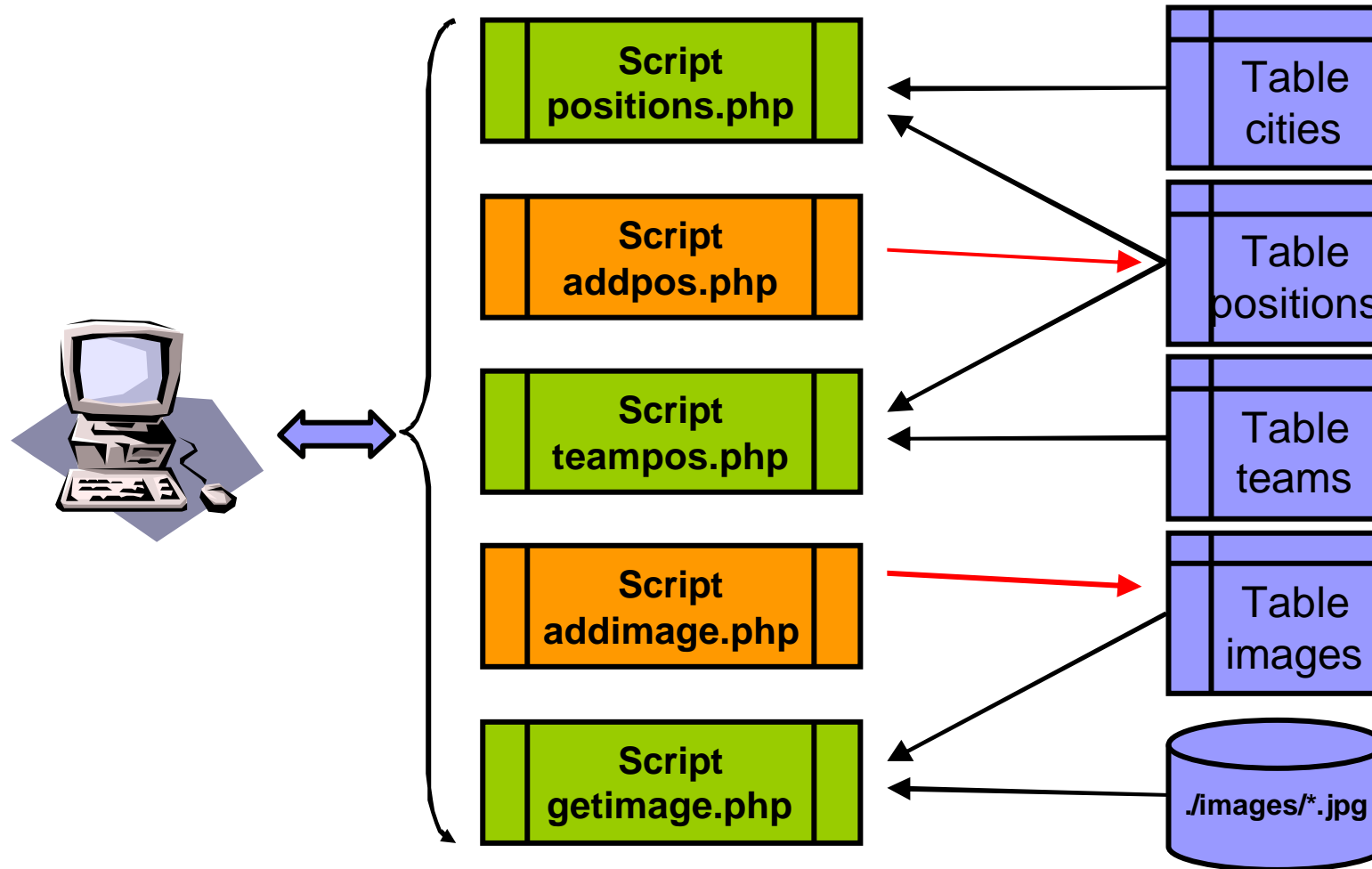




Project implementation

- Collect generated maps from ftp server or local directory and feed into webserver
- Read balloon positions from e-mail messages, calculate distances and feed into webserver
- All data is kept in MySQL database with timestamp for automatic delay handling

Data structure





REXX coming into the game

- Similar tasks already implemented in REXX for different presentations
- Combination of available REXX function packages and sample code
- Rapid development cycle for short term implementation




Project software details

- Pentium 166 MHz, 64 MB notebook with DSL
- Object REXX for Linux
- RxFTP package for file transfer
- RxSock package for http requests and POP3 e-mail retrieval
- RxMath for trigonometric calculations
- ImageMagick and NetPBM for image conversion tasks
- Tasks controlled with cron jobs, fully automatic



Collect map images

- Download from ftp server or copy from local directory
- Read timestamp from ftp server file
- Add timestamp to graphic file using file date in local directory or on ftp server
- Resize and recompress for web display
- Upload to web server, add entry to web database



Download images from ftp server (main program)

```
Call RxFuncAdd "FtpLoadFuncs", "rxftp", "FtpLoadFuncs"  
Call FtpLoadFuncs
```

```
Call FtpSetUser "www.myftpserver.com", "userid", "password"  
Call FtpChDir "gb2003"  
Call GetFTPFiles "*.jpg"  
Call FtpLogOff
```

```
Call FtpSetUser "www.dfsv.de", "userid", "password"  
Call FtpChDir "/dfsv/tracking/gb2003/files"  
URLPrefix = "http://www.dfsv.de/tracking/gb2003/addimage.php?"  
Call UploadImages  
Call FtpLogOff  
Exit
```



FTP server files

```
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 u36049018 ftpusers   4096 Apr  9 23:39 .
drwx---r-t  11 u36049018 ftpusers   4096 Apr  9 23:38 ..
-rw-r--r--   1 u36049018 ftpusers  125956 Apr  9 23:39 gor1.jpg
-rw-r--r--   1 u36049018 ftpusers  155516 Apr  9 23:39 gor2.jpg
-rw-r--r--   1 u36049018 ftpusers  178989 Apr  9 23:39 gor3.jpg
-rw-r--r--   1 u36049018 ftpusers  119624 Apr  9 23:39 gor4.jpg
226 Transfer complete.
ftp> quit
221 Goodbye.
```



Get ftp server files (1)

GetFTPFiles:

Parse Arg FileSpec

Call FtpDir FileSpec, "Files."

Do i = 1 To Files.0

Parse Var Files.i FileSize FileMonth FileDay FileTime FileName .

If Pos(":", FileTime) \= 0 Then /* young files with time */

FileYear = Left(Date("S"), 4) /* assume current year (not correct Jan-Apr) */

Else

Do /* old files with date and year only, assume noon as time */

FileYear = FileTime

FileTime = "12:00"

End



Get ftp server files (2)

```
FileDate = Date("S", FileDay FileMonth FileYear, "N")
FileTime = FileTime || ":00"
Call GetFileNames LocalPath, FileName, FileDate, FileTime, -7200 /* 2 hrs */
If Stream(LocalFileName, "C", "QUERY EXISTS") = "" Then
  Do
    /* local file does not exist, download it */
    Call FtpGet '/tmp/' || FileName, FileName, "binary"
    Call ProcessFile
  End
End

Return
```



Create local filenames (1)

GetFileNames:

Parse Arg LocalPath, FileName, FileDate, FileTime, UTCOffset

/* correct file time by DST + UTC offset */

FileTime = Time("S", FileTime, "N") + UTCOffset

If FileTime < 0 Then

Do

FileTime = FileTime + 86400

FileDate = Date("S", Date("B", FileDate, "S") - 1, "B")

End

FileTimeStr = Time("N", FileTime, "S")

FileTime = Left(FileTimeStr, 2) || SubStr(FileTimeStr, 4, 2) || ,
Right(FileTimeStr, 2)



Create local filenames (2)

```
/* Create output filenames, input filename may be any extension. Output filenames: */
/* LocalFileName: filename with timestamp and extension .png as backup of original file */
/* ThumbFileName: filename with timestamp and extension .jpg with reduced size */
/* StampedFileName: filename with timestamp and extension .jpg at full size */
DotPos = LastPos(".", FileName)
BaseName = LocalPath || Left(FileName, DotPos - 1) || "_" || FileDate || "_" || FileTime

LocalFileName = BaseName || "_bak.png"
ThumbFileName = BaseName || "_thumb.jpg"
StampedFileName = BaseName || ".jpg"
ImageTimeStamp = FileDate || FileTime
TimeStampText = "Date:" Left(FileDate, 4) || "-" || SubStr(FileDate, 5, 2) ||,
                "-" || Right(FileDate, 2) Left(FileTimeStr, 5) || " UTC"
Return
```

Images with timestamp

- Current file date is added as timestamp in lower right corner of image:





Convert images, add timestamp (1)

ProcessFile: /* uses ImageMagick and NetPBM package */

```
'convert /tmp/' || FileName LocalFileName /* PNG backup copy of original file */
```

/* create image with timestamp */

```
'convert -font /home/chm/gb2003/arial.ttf -pointsize 12 -fill black -draw "text 2,21 "" ||,  
    TimeStampText || "" /home/chm/gb2003/whitespace.png /tmp/copy1.ppm'
```

```
'pnmcrop /tmp/copy1.ppm >/tmp/copy2.ppm'
```

```
'convert -bordercolor white -border 2x2 /tmp/copy2.ppm /tmp/copy3.ppm'
```

```
'composite -quality 75 -gravity SouthEast -compose over /tmp/copy3.ppm' ,  
    LocalFileName StampedFileName
```

/* create thumbnail image */

```
'convert' LocalFileName ThumbFileName
```

```
'mogrify -quality 85 -geometry 200x150' ThumbFileName
```



Convert images, add timestamp (2)

```
/* determine image dimensions */
```

```
Dimension = GetImageDimension(StampedFileName)
```

```
Parse Var Dimension FullWidth FullHeight
```

```
Dimension = GetImageDimension(ThumbFileName)
```

```
Parse Var Dimension ThumbWidth ThumbHeight
```

```
/* remove temp files */
```

```
'rm /tmp/copy1.ppm'
```

```
'rm /tmp/copy2.ppm'
```

```
'rm /tmp/copy3.ppm'
```

```
'rm /tmp/' || FileName
```

```
Call AddUpload LocalFileName, StampedFileName, ThumbFileName,,  
ImageTimeStamp, FullWidth,FullHeight, ThumbWidth, ThumbHeight
```

```
Return
```



Get image dimensions

GetImageDimension:

Parse Arg FN

output_file = '/tmp/getimagedimensiongb2003'

'identify -format "%w %h" FN '>' || output_file

Do While Lines(output_file) > 0

 IDLine = LineIn(output_file)

 If Length(IDLine) > 2 Then

 Dimensions = IDLine

End

Call Stream output_file, "C", "CLOSE"

Return Dimensions



Add images to upload list

AddUpload:

Parse Arg LFN, SFN, TFN, TS, FW, FH, TW, TH

idx = UploadFiles.0 + 1

UploadFiles.idx.Local = LFN

UploadFiles.idx.Stamped = SFN

UploadFiles.idx.Thumb = TFN

UploadFiles.idx.Timestamp = TS

UploadFiles.idx.FWidth = FW

UploadFiles.idx.FHeight = FH

UploadFiles.idx.TWidth = TW

UploadFiles.idx.THeight = TH

UploadFiles.0 = idx

Return



Upload image files

UploadImages:

```
Do i = 1 To UploadFiles.0
```

```
  FN = UploadFiles.i.Stamped /* upload stamped image */
```

```
  File_BaseName = FileSpec("N", FN)
```

```
  Call FtpPut FN, File_BaseName, "binary"
```

```
  FN = UploadFiles.i.Thumb /* upload thumbnail image */
```

```
  Thumb_BaseName = FileSpec("N", FN)
```

```
  Call FtpPut FN, Thumb_BaseName, "binary"
```

```
  /* from the basename determine the imagetype */
```

```
  Type = 20 /* Overview picture */
```

```
  If Pos("gor0", File_BaseName) > 0 Then Type = 20
```

```
  ...
```



Add images to database

```
/* now add entry into image table */
```

```
URL = "imagetime=" || UploadFiles.i.Timestamp || "&imagetype=" || Type ||,  
      "&thumbnail=" || Thumb_BaseName || "&filename=" || File_BaseName ||,  
      "&fullwidth=" || UploadFiles.i.FWidth ||,  
      "&fullheight=" || UploadFiles.i.FHeight ||,  
      "&thumbwidth=" || UploadFiles.i.TWidth ||,  
      "&thumbheight=" || UploadFiles.i.THeight
```

```
URL1 = URLPrefix || URL
```

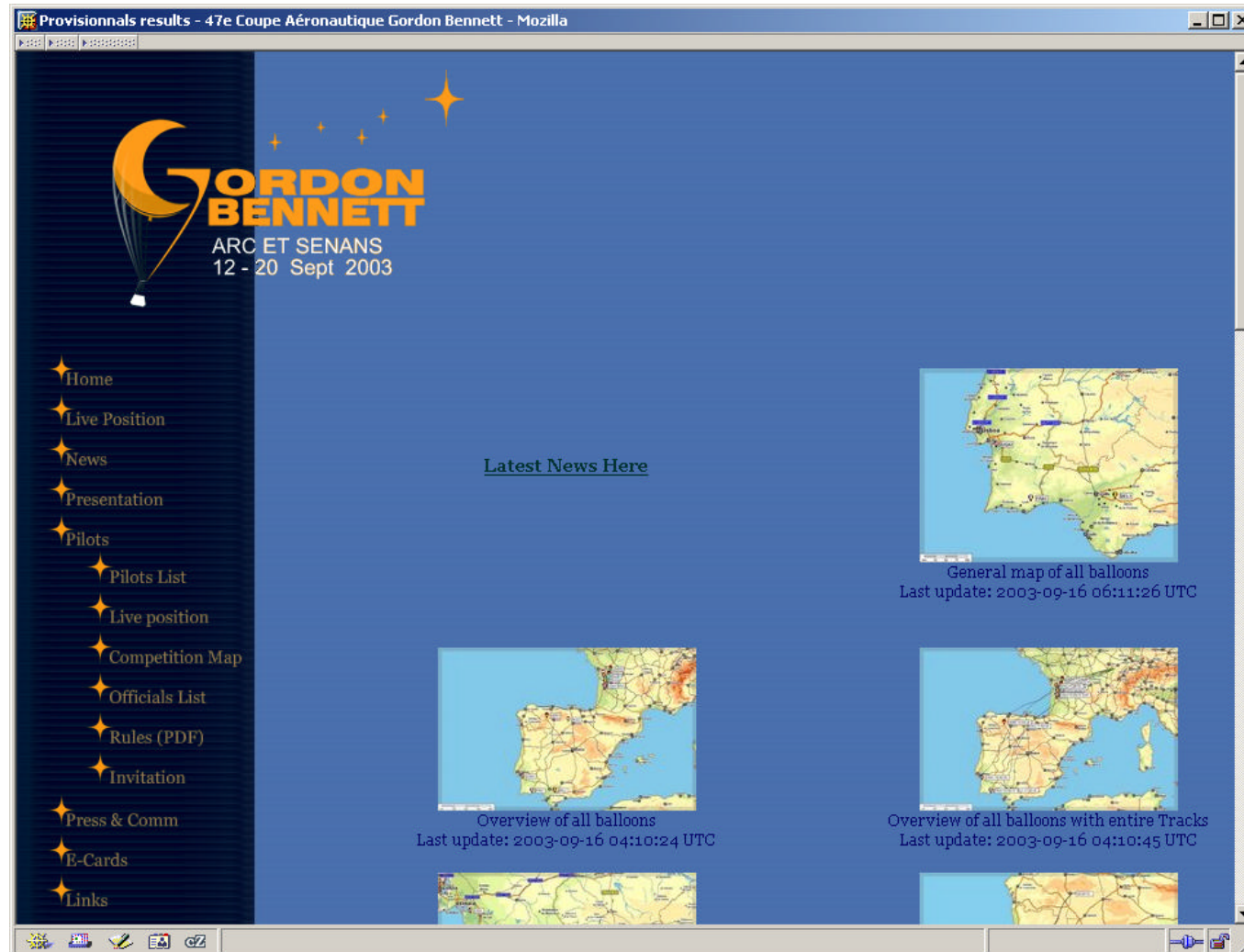
```
RetCode = GetDocument(URL1, "NOPROXY")
```

```
End
```

```
Return
```

```
http://www.dfsv.de/tracking/gb2003/addimage.php?imagetime=20030831141038&  
      imagetype=23&thumbnail=gor3_20030831_141038_thumb.jpg&  
      filename=gor3_20030831_141038.jpg&fullwidth=1016&fullheight=638&  
      thumbwidth=200&thumbheight=126
```


Web presentation



Web presentation

GB 2003 - Ballon Positionen - Mozilla



Deutscher Freiballonsport-
Verband e.V. im DAeC



Live Tracking realisiert mit Equipment der Firma **SatPro** in Altenholz.

Kartendarstellungen

Zum Vergrößern bitte die einzelnen Karten anklicken.

Übersichtskarte aller Ballone	Übersicht aller Ballone	Übersicht aller Ballone mit gesamten Tracks	Spitzenreiter mit Tracks vom aktuellen Tag
			
Letzte Aktualisierung: 2003-09-16 06:11:26 UTC	Letzte Aktualisierung: 2003-09-16 04:10:24 UTC	Letzte Aktualisierung: 2003-09-16 04:10:45 UTC	Letzte Aktualisierung: 2003-09-16 05:11:09 UTC
Vorherige Karte anzeigen: <input type="text" value="2003-09-16 06:11:26 UTC"/> ▾ <input type="button" value="Zeigen"/>	Vorherige Karte anzeigen: <input type="text" value="2003-09-16 04:10:24 UTC"/> ▾ <input type="button" value="Zeigen"/>	Vorherige Karte anzeigen: <input type="text" value="2003-09-16 04:10:45 UTC"/> ▾ <input type="button" value="Zeigen"/>	Vorherige Karte anzeigen: <input type="text" value="2003-09-16 05:11:09 UTC"/> ▾ <input type="button" value="Zeigen"/>

Windows taskbar icons: Start, Internet Explorer, Mozilla, Outlook, GZ, System tray with volume and network icons.



Position e-mail processing

- E-mails sent to competition center for each device as soon as position report is received by computing center
- E-mail contains current position, altitude, heading, speed and other information
- Position is added to tracking database together with distance to launch point



Get POP3 mails - main

Call RxFuncAdd "SockLoadFuncs", "rxsock", "SockLoadFuncs"

Call SockLoadFuncs "silent"

Call GetPositionMails "pop3.server.ip", "userid", "password"

Exit

GetPositionMails: Procedure Expose MailPath

Parse Arg POP3Server, POP3User, Password

mysock = Login(POP3Server, POP3User, Password)

...



Get all mails from POP3 server

```
If mysock \= -1 Then
  Do
    MailCount = GetMailCount(mysock)
    TimeStr = Time("N")
    FileName = MailPath || Date("S") || "_" || Left(TimeStr, 2) || SubStr(TimeStr,4,2) ||
      Right(TimeStr, 2) || "_"
    Do i = 1 To MailCount
      FileName = FileName || Right(i, 3, "0") || ".msg"
      MailText = GetMail(mysock, i)
      If Length(MailText) > 0 Then
        Do
          Call CharOut FileName, MailText
          Call Stream FileName, "C", "CLOSE"
          Parse Var MailText Header "0D0A0D0A"x Body
          If ProcessMail(Body) = 0 Then Call DeleteMail mysock, i
        End
      End
    End
    Call Logout mysock
  End
Return
```



Create socket to POP3 server

Login: Procedure

Parse Arg Server, Account, Password

rc = SockGetHostByName(Server, "host.!")

If rc = 0 Then

Do

Say "POP3 server name <" || Server || "> could not be resolved. Terminating!"

Return -1

End

host.!family = "AF_INET"

host.!port = 110

Socket = SockSocket("AF_INET", "SOCK_STREAM", 0)

If Socket = -1 Then

Do

Say "Could not create local socket. Terminating!"

Return -1

End

...



Connect to POP3 server

```
rc = SockConnect(Socket, "host!")  
If rc \= 0 Then  
  Do  
    Say "Could not connect to POP3 server <" || Server || ">. Terminating!"  
    Return -1  
  End
```

```
Answer = ReadReplyLine(Socket)  
If Word(Answer, 1) = "-ERR" Then  
  Do  
    Say "Error connecting to POP3 server <" || Server || ">."  
    Say "Error message was:" Answer  
    Return -1  
  End
```

...



Login to POP3 server

```
Answer = SendPOP3Command(Socket, "USER " || Account)
If Word(Answer, 1) = "-ERR" Then
  Do
    Say "Unknown POP3 user <" || Account || ">."
    Say "Error message was:" Answer
  Return -1
End
```

```
Answer = SendPOP3Command(Socket, "PASS " || Password)
If Word(Answer, 1) = "-ERR" Then
  Do
    Say "Invalid POP3 password."
    Say "Error message was:" Answer
  Return -1
End
Return Socket
```




Logout from POP3 server

Logout: Procedure

- Parse Arg Socket

- Call SendPOP3Command Socket, "QUIT"

- Call SockShutdown Socket, 2

- Call SockClose Socket

- Return

SendPOP3Command: Procedure

- Parse Arg Socket, Command

- CRLF = "0D0A"x

- Call SockSend Socket, Command || CRLF

- Return ReadReplyLine(Socket)



Get mail count from POP3 server

GetMailCount: Procedure

Parse Arg Socket

Data = SendPOP3Command(Socket, "STAT")

If Word(Data, 1) = "+OK" Then

Parse Value Data With "+OK" NoOfMails NoBytes

Else

Do

Say "Could not retrieve number of mails waiting."

Say "Error was:" Data

NoOfMails = 0

End

Return NoOfMails



Delete mail from POP3 server

DeleteMail: Procedure

Parse Arg Socket, MsgNumber

Command = "DELE" MsgNumber

Data = SendPOP3Command(Socket, Command)

If Word(Data, 1) = "+OK" Then

Result = 0

Else

Do

Say "Could not delete specified message number."

Say "Error was:" Data

Result = -1

End

Return Result



Retrieve mail from POP3 server

GetMail: Procedure

Parse Arg Socket, MsgNumber

Command = "RETR" MsgNumber

Call SockSend Socket, Command || "0D0A"x

Say "Reading message" MsgNumber || ":"

Message = ""

Data = ""

Do Forever

Bytes = SockRecv(Socket, "Data", 1024)

Message = Message || Data

If Bytes >= 3 & SubStr(Data, Bytes-2, 3) = "." || "0D0A"x Then

Leave

End

Return Message



Read reply line from POP3 server

ReadReplyLine: Procedure

Parse Arg Socket

Line = ""

Data = ""

Bytes = 1

Do While (Pos("0A"x, Line) = 0) & (Bytes \= 0)

 If SockRecv(Socket, "Data", 1024, "MSG_PEEK") = 0 Then Leave

 Bytes = SockRecv(Socket, "Data", 1024)

 Line = Line || Data

End

If Length(Line) > 0 Then

 Do

 If Length(SubStr(Line, Pos("0A"x, Line) + 1)) > 0 Then Say "Warning, multi line reply received!"

 Reply = Strip(Left(Line, Pos("0A"x, Line) - 1), "B", "0D"x)

 End

Else

 Reply = "-ERR - no answer from POP3 server"

Return Reply



Position e-mail body format

```
BoxID | PosDate | PosTime | Latitude |  
Longitude | Altitude | Temperature | Battery |  
Speed | Course | Rest
```

(| = tab character)

Example:

```
59404424 | 2003-09-15 | 19:59:26 | 37 Grad  
44 Min 46 Sek N | 6 Grad 22 Min 56 Sek W |  
1842.000000 | # | # | # | 269.000000 | # | #  
| # | # | # | 138 | #
```



Process position e-mails

ProcessMail: Procedure

Parse Arg MailBody

RetCode = -1

MailBody = ChangeStr("0D0A"x, MailBody, "0A"x)

Do While Length(MailBody) > 0

LineBreak = Pos("0A"x, MailBody)

If LineBreak > 0 Then

Do

Line = Strip(Left(MailBody, LineBreak - 1), "B")

MailBody = SubStr(MailBody, LineBreak + 1)

End

Else

Do

Line = Strip(MailBody, "B")

MailBody = ""

End

...



Parse position e-mail line

```
If Length(Line) > 2 Then
```

```
Do
```

```
/* replace tab characters with paragraphs */
```

```
Line = ChangeStr("09"x, Line, "$")
```

```
Line = ChangeStr("#", Line, "0")
```

```
/* Parse line into elements */
```

```
Parse Var Line BoxID "$" PosDate "$" PosTime "$" Latitude "$" Longitude "$",  
        Altitude "$" Temp "$" Batt "$" Speed "$" Course "$" .
```

```
Latitude = GetLatLon(Latitude)
```

```
Longitude = GetLatLon(Longitude)
```

```
/* strip unnecessary blanks from variables */
```

```
BoxID = Strip(BoxID, "B")
```

```
..
```

```
/* convert latitude/longitude to DMS format */
```

```
LatDMS = FmtLatLon(Latitude, 1)
```

```
LongDMS = FmtLatLon(Longitude, 0)
```

```
...
```




Prepare URL to add position

```
/* Get Distance from Chatellerault */
```

```
Distance = GetDistance(Latitude, Longitude)
```

```
/* now construct URL to pass this information into the database */
```

```
URL = "boxid=" || BoxId || "&reporttime=" || PosDate PosTime ||,  
      "&latitude=" || Latitude || "&longitude=" || Longitude ||,  
      "&latdms=" || LatDMS || "&longdms=" || LongDMS ||,  
      "&altitude=" || Altitude || "&distance=" || Distance ||,  
      "&temperature=" || Temp || "&battery=" || Batt ||,  
      "&currspeed=" || Speed || "&course=" || Course
```

```
...
```



Add position to web database

```
URL = ChangeStr(" ", URL, "%20")
URL = ChangeStr("°", URL, "%B0")
URL = ChangeStr("'", URL, "%27")
URL = ChangeStr("-", URL, "%2D")
URL = ChangeStr(':', URL, "%3A")
URL = "http://www.dfsv.de/tracking/gb2003/addpos.php?" || URL
```

```
If (Latitude \= 0.00000) & (Longitude \= 0.00000) & (Distance < 4000) Then
    RetCode = GetDocument(URL, "NOPROXY")
```


```
Else
```

```
    RetCode = 0
```

```
End
```

```
End
```

```
Return RetCode
```



Parse latitude/longitude value

GetLatLon: Procedure

Parse Arg LatLon

```
/* "54 Grad 24 Min 20 Sek N" */
```

```
Parse Var LatLon Degr "Grad" Mins "Min" Secs "Sek" Prefix
```

```
Degrees = Degr + Mins/60 + Secs/3600
```

```
If Prefix = "W" | Prefix = "S" Then
```

```
  Degrees = Degrees * -1.0
```

```
Return Degrees
```



Reformat latitude/longitude value

FmtLatLon: Procedure

Parse Arg LatLon, IsLatitude

If IsLatitude = 1 Then

Do

DegrLen = 2

If LatLon >= 0.0 Then Prefix = "N"

Else Prefix = "S"

End

Else

Do

DegrLen = 3

If LatLon >= 0.0 Then Prefix = "E"

Else Prefix = "W"

End

LatLon = Abs(LatLon)

...



Reformat latitude/longitude value

```
Degr = LatLon % 1
Temp = (LatLon - Degr) * 60
Min = Temp % 1
Temp = (Temp - Min) * 60
Secs = Format(Temp,,0)
If Secs >= 60 Then
  Do
    Secs = Secs - 60
    Min = Min + 1
  End
If Min >= 60 Then
  Do
    Min = Min - 60
    Degr = Degr + 1
  End
Coord = Prefix || Right(Degr, DegrLen, "0") || "°" || Right(Min, 2, "0") || "" || Right(Secs, 2, "0") || ""
Return Coord
```



Determine distance to launch point

- Given: coordinates lat1/lon1 & lat2/lon2 in degrees

$$\text{Distance} = \left(p/2 - \arcsin\left(\sin(\text{lat1}) * \sin(\text{lat2}) + \cos(\text{lat1}) * \cos(\text{lat2}) * \cos(\text{lon1} - \text{lon2}) \right) \right) * 6371$$

(arcsin needs to return radians value)



Distance calculation using RxMath

GetDistance: Procedure

Parse Arg Lat, Long

LatLaunch = 47.03305555 /* Arc-et-Senans */

LongLaunch = 5.7777777

DistArc = RxCalcPi(16)/2 - ,

RxCalcArcSin(RxCalcSin(Lat, 16, "D") * RxCalcSin(LatLaunch, 16, "D") +,
RxCalcCos(Lat, 16, "D") * RxCalcCos(LatLaunch, 16, "D") * ,
RxCalcCos(Long - LongLaunch, 16, "D"), 16, "R")

Distance = DistArc * 6371.0 /* FAI standard globe */

If Distance < 0.0001 Then Distance = 0.0

Return Distance

Web presentation

GB 2003 - Ballon Positionen - Mozilla

Positionsmeldungen

Sie können die Sortierreihenfolge durch Anklicken der Überschriften ändern!


Start-Nr.	Team Name	Pilot/ Co-Pilot	Ballon	Kennzeichen	Position	Koordinaten	Letzte Meldung (UTC)	Distanz zum Startort (km)	Status	Alle Positionen	Download
1	Switzerland I	Krebs, Max Vollenweider, Walter	Kandersteg	D-OCFT	34 km SW Besancon, France	N47°01'59" E005°46'40"	2003-09-13 18:00:00	0.00	wartet	Meldungen/ Detailkarten	Ozi Track Garmin Track
2	France II	Villey-Desmeserets, Thierry Buron Pilatre, Philippe	L'inconnu	F-GIRE	35 km WSW Angoulême, France	N45°34'04" W000°17'11"	2003-09-14 09:00:25	493.37	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
3	Belgium I	De Cock, Philippe Van Havere, Ronny	Belgica 2	OO-BCX	43 km O Sevilla, Spain	N37°21'41" W005°30'10"	2003-09-15 17:00:16	1418.35	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
4	Sweden	Akerstedt, Hans Balkedal, Jan	Münsterland	D-OWML	26 km W Angoulême, France	N45°39'23" W000°11'37"	2003-09-14 15:58:06	483.10	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
5	United States III	Sullivan, Mark Levin, David	Warsteiner	D-OWNT	22 km N Bordeaux, France	N45°02'45" W000°38'08"	2003-09-14 10:59:15	541.89	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
6	Great Britain	Folkes, Janet Butter, Colin	Motor Colombus	HB-BJS	35 km NNO Bordeaux, France	N45°07'22" W000°21'20"	2003-09-14 11:46:16	518.39	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
7	Austria I	Stürzlinger, Gerald Fürstner, Johann	Nautila Nautilaero	D-ORZL	74 km WNW Angoulême, France	N45°56'33" W000°44'02"	2003-09-14 15:50:26	512.89	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track
8	Germany I	Eimers, Wilhelm Landsmann, Bernd	Columbus V	D-OOWE	67 km WNW Leon, Spain	N42°56'28" W006°16'45"	2003-09-15 10:59:26	1050.21	gelandet	Meldungen/ Detailkarten	Ozi Track Garmin Track

Web presentation

GB 2003 - Team Positionsmeldungen - Mozilla

Team Nr. 4
Team Name Belgium I
Pilot De Cock, Philippe
Co-Pilot Van Havere, Ronny
Ballon Belgica 2
Kennzeichen OO-BCX
Status gelandet

Kartendarstellung
Zum Vergrößern bitte die Karte anklicken.



Vorherige Karte anzeigen:
2003-09-15 19:02:39 UTC Zeigen

Laden Sie sich die alle Positionen dieses Teams als Trackpunktdatei im [OziExplorer](#) oder [Garmin](#) Format herunter.

Meldungszeit (UTC)	Position	Koordinaten	Distanz zum Startort (km)
2003-09-15 17:00:16	43 km O Sevilla, Spain	N37°21'41" W005°30'10"	1418.35
2003-09-15 16:08:25	52 km O Sevilla, Spain	N37°21'30" W005°24'11"	1413.31
2003-09-15 15:01:56	56 km WSW Cordoba, Spain	N37°39'51" W005°19'35"	1382.15
2003-09-15 12:59:45	64 km NNW Cordoba, Spain	N38°23'05" W005°09'43"	1310.43
2003-09-15 08:00:46	94 km W Toledo, Spain	N39°43'13" W005°05'34"	1195.06
2003-09-14 23:59:45	64 km S Valladolid, Spain	N41°04'44" W004°43'43"	1067.64
2003-09-14 15:59:15	61 km NNO Burgos, Spain	N42°53'38" W003°28'57"	860.90



Summary

- Successful use of infrastructure for two events (2002 and 2003)
- 300,000 pageviews alone in the 2003 race on the German page during 3 days
- Competition teams used current position information to coordinate their weather briefings and strategy
- No serious problems encountered during usage
- Plan to use equipment in 2004 race in August



Links

- Coupe Aéronautique Gordon Bennett (general info):
<http://www.coupegordonbennett.org>
- Gordon Bennett race 2003:
<http://www.gordonbennett2003.org/>
<http://www.dfsv.de/tracking/gb2003/positionen.php> (German only)
- Gordon Bennett race 2004:
<http://www.gordonbennett2004.org>
Launch planned for August 28th, 2004
- SatPro company: <http://www.satpro.org> (German only)
- Skywave Mobile (DMR-200 Inmarsat tracking device):
<http://www.skywavemobile.com>



Contact

Christian Michel

cmichel@de.ibm.com (Business)

c.michel@dfsv.de (Webmaster German
balloon federation)

Video



- 5 min. video of 2002 Gordon Bennett race
- Launch from Chatellerault/France
- Winner flew 1282 km
- 400 km offshore on atlantic ocean
- 3D Studio Max animation of flight paths

**Produced by Astrid Gerhardt & Steffen Kutter
Astrid is current world record holder for gas balloons for altitude, distance & duration**