

## Konfiguration und Einstellungen des Servers für das Webprojekt Without Addresses, 1997 HP 9000 Serie 800 (Server) mit HP-UX Einführung

Interaktive Grafiken wurden mithilfe von Perl, Ghostscript und den Netpbm-Tools auf einem Apache-Webserver erstellt.

### 1. Apache-Webserver Version 1.2

Der Apache-Webserver war in den späten 1990er Jahren die dominierende Software für das Hosting von Websites.

Er erlaubte mithilfe des Common Gateway Interface (CGI) externe Programme oder Skripte auf einem Webserver auszuführen und mit einem Webbrowser zu interagieren.

Um CGI-Skripte auszuführen und Virtual Hosts zu konfigurieren, wurden bestimmte Einstellungen in der httpd.conf-Datei vorgenommen. Die Konfiguration:

```
LoadModule cgi_module modules/mod_cgi.so
```

```
<VirtualHost *:80>
```

```
ServerName example.com
```

```
DocumentRoot /opt/www/htdocs/wa
```

```
<Directory "/opt/www/htdocs/wa">
```

```
Options +FollowSymLinks +Indexes
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
<Directory "/opt/www/htdocs/wa/cgi-bin">
```

```
Options +ExecCGI
```

```
AddHandler cgi-script .cgi .pl
```

```
Require all granted
```

```
</Directory>
```

### 2. Perl Version 5.004

Perl wurde verwendet, um CGI-Skripte zu schreiben, die zur Verarbeitung von Formulardaten und zur Generierung dynamischer Inhalte erforderlich waren. Dabei wurde auch das cgi-lib.pl Modul verwendet. Das cgi-lib.pl-Modul ist eine Sammlung von Perl-Routinen, die oft verwendet werden, um CGI-Skripte zu schreiben. Es stellt Funktionen bereit, um HTTP-Formulardaten zu verarbeiten, Daten an Browser zurückzugeben und andere häufige Aufgaben bei der Erstellung von CGI-Skripten zu vereinfachen. Das Modul wird normalerweise in Perl-CGI-Skripten eingebunden, um den Umgang mit

Formular Daten und die Interaktion mit dem Webserver zu erleichtern. Es ist kein Standardmodul von Perl, sondern eine externe Bibliothek, die von verschiedenen Quellen zur Verfügung gestellt wird.

<https://cgi-lib.berkeley.edu/>

### 3. Ghostscript 5.03

Ghostscript wurde für die Erzeugung und Manipulation von Grafiken im PostScript-Format verwendet. Die Konfiguration von Ghostscript beinhaltet die Installation und die Bereitstellung von Berechtigungen für den Zugriff durch CGI-Skripte. Das Verzeichnis in dem Ghostscript installiert ist muss im PATH des Webserver-Benutzerkontos enthalten sein. Für das Projekt wurde eine spezielle Handschrift für Ghostscript erstellt.

### 4. Netpbm-Tools Version 10.0

Die Netpbm-Tools wurden verwendet, um PostScript-Grafiken in das GIF-Format zu konvertieren.

Das Verzeichnis in dem die Netpbm-Tools installiert sind muss im PATH des Webserver-Benutzerkontos enthalten sein. Ein Beispiel für die Verwendung der Netpbm-Tools in einem Perl-CGI-Skript:

```
print "Content-Type: image/gif\n\n";  
system("/usr/local/netpbm/pstogif /opt/www/htdocs/wa/input.ps >  
/var/www/htdocs/wa/output.gif");
```

### 5. Zusammenfassung

Die erfolgreiche Konfiguration eines Webprojekts in den späten 1990er Jahren auf einem HP-UX-System erforderte die sorgfältige Anpassung verschiedener Softwarekomponenten wie Apache, Perl, Ghostscript und den Netpbm-Tools. Die in diesem Bericht beschriebenen Konfigurationen und Beispiele sollen ein besseres Verständnis der erforderlichen Schritte vermitteln und die Grundlage für das Verständnis der Skripte, die für Without Addresses verwendet wurden.

Die Skripte:

1. Das Skript check erzeugt die HTML-Ausgabe, verarbeitet die Eingabe und erzeugt alle Bilder und HTML Dateien
2. Das Skript font.pl erzeugt eine benutzerdefinierte Schriftart namens "KHRand". Die Schriftart wurde für die Darstellung der Texte bei Without Addresses verwendet.

```
#!/usr/bin/perl

# Uncomment to set the Ghostscript library path
# $ENV{'GS_LIB'} = "./fonts";

use Socket;
require "../cgi-lib.pl";
require "./form.ph";
require "timelocal.pl";

# Initialize data and temp directories
$data = "data";
$temp = "temp";

# Get current time for unique ID
$id = time;

# Autoflush the output
$| = 1;

# Set the content type for HTTP response
print "Content-type: text/html\n\n";

# Run makefont script
system("./makefont");

# Parse form data
&ReadParse;

# Check if 'name' parameter is empty
if ($in{'name'} eq "") {
    print $form;
    # exit; # Exit is commented out
}

# Set search pattern and name
$pat = "\"$in{'name'}\"";
$name = $in{'name'};
$pat =~ s/\ /\\+/g; # Escape spaces for the search query
$name = "sex"; # Hardcoded name

# Change directory to 'files'
chdir("files");

# Perform web search using w3mir and extract URLs using getHREF.pl
open(F, "../w3mir -l http://www.altavista.com/cgi-bin/query?pg=q\\&sc=on\\&hl=on\\&q=$name\\&kl=XX\\&stype=stext | ../getHREF.pl |");
while (<F>) {
    push(@out, $_);
}
close(F);

# Get the first URL and shift the array
$where = $out[0];
shift(@out);

# Open a file to save the output
open(F, ">$id");

# Clean up the output data
map {s/\\//eg} @out;
map {s/\\(//eg} @out;
map {s/
//eg} @out;
map {s/\\&#(\\d{1,3})\\//eg} @out;
map {s/(\\&.*?\\)//gesxi} @out;
```

```
map {s/(<SCRIPT.*?SCRIPT>)/&gather($1)/gesxi} @out;
map {s/(<.*?>)/&gather($1)/gesxi} @out;
map {s/(\{.*?\})//gesxi} @out;
map {s/(\\".*?\\")//gesxi} @out;

# Filter out HTML tags and empty lines
@out2 = grep(!/<.*?/, @out);
@out = grep(!/.*?>/, @out2);
shift(@out);

# Write cleaned data to the file
print F @out;
close(F);

# Convert GIF to PNM and then to PS (PostScript)
open(IN, "/usr/X11R6/bin/giftoPNM $id.gif | /usr/X11R6/bin/pnmtops |");
while ($line = <IN>) {
    push(@image, $line) unless $line =~ /showpage/;
}
close(IN);

# PostScript header for drawing
$start = "
gsave
72 2.54 div dup scale
5 10 translate

/top 2 31 exp 1 sub def

0 0 .5 setrgbcolor
usertime srand
%rand top div 2 mul setlinewidth 1 setlinecap
.1 setlinewidth 1 setlinecap
1 1 15 { pop
    rand top div 15 mul
    rand top div 24 mul
    moveto 0 0 rlineto
} for
stroke

grestore

/LM 70 def
/MM 100 def
/lh {gsave newpath 0 0 moveto
    ({) true charpath flattenpath pathbbox
    4 1 roll
    pop pop pop grestore
} def

/nl {currentpoint
    lh 1.4 mul sub
    exch pop
    LM exch
    moveto
} def

gsave
/Hershey-Script-Complex findfont
24 scalefont
setfont
MM 600 moveto
270 rotate
1 1 0 setrgbcolor
( $where) show
```

```
grestore
%/Hershey-Script-Complex-Bold findfont
/KHRand findfont
  24 scalefont
  setfont
    LM 700 moveto
0 0 .5 setrgbcolor
  (          $name) show nl

/KHRand          findfont
  20 scalefont
  setfont
%          LM 700 moveto
0 0 .5 setrgbcolor
";

# PostScript footer
$end = "
  (...) show nl
showpage
";

# Read input file
open(F, "<$id");
@input = <F>;
close(F);

# Join input lines and replace spaces with newlines
$string = join(" ", @input);
$string =~ s/ /\n/g;

# Write formatted output to the file
open(F, ">$id");
select(F);

$~ = "OUTLINE";
write;
close(F);

select(STDOUT);

# Create PostScript lines from input file
open(T, "<$id");
while ($line = <T>) {
  chop($line);
  push(@ps, "($line) show nl", "\n") unless $line =~ /=|\.gif| if|function|return|
  applet|seed|length|\}|\/{/;
}
close(T);

# Create final PostScript file
$ps = "$id.ps";
open(F, ">$ps");
print F @image;
print F $start;
print F @ps;
print F $end;
close(F);

# Generate image using Ghostscript
open(F, "<$ps");
while (<F>) {
  $code .= $_;
}
close(F);
```



```
        $gif = $b;
    } else {
        $gif = $a;
    }
}
# push(@img, $gif, "\n");
}
return "";
}

sub image_pos {
    @all = @_;
    srand;
    local(@temp);
    push(@temp, splice(@all, rand(@all), 1)) while @all;
    @all = @temp;
    return @all;
}

sub rgb_r {
    @all = @_;
    srand;
    local(@temp);
    push(@temp, splice(@all, rand(@all), 1)) while @all;
    @all = @temp;
    return @all;
}

sub rgb_g {
    @all = @_;
    srand;
    local(@temp);
    push(@temp, splice(@all, rand(@all), 1)) while @all;
    @all = @temp;
    return @all;
}

sub rgb_b {
    @all = @_;
    srand;
    local(@temp);
    push(@temp, splice(@all, rand(@all), 1)) while @all;
    @all = @temp;
    return @all;
}
}
```

```

% Name der Schriftdatei
$font="KHRand.gsf";

% Anfang des PostScript-Skripts
$begin="
%!
% Allgemeine Definitionen für die KH-Schriften.
/top 2 31 exp 1 sub def % Berechnet den oberen Wert für die Zufallszahlengenerierung
usertime srand % Initialisiert den Zufallszahlengenerator mit der aktuellen
Zeit

/MakeKHFont where
{ pop 80 string
  { currentfile 1 index readline pop
    (%END) eq { exit } if
  } loop pop
} if

userdict begin
/KHBuildDict 10 dict def % Erstellt ein neues Wörterbuch für die Schriftartdefinition

/BuildKHChar {
  exch begin
    PenWidth setlinewidth % Setzt die Linienstärke
    Encoding exch get % Holt das aktuelle Zeichen aus der
Kodierung
    dup CharData exch known not { pop /.notdef } if
    CharData exch get % Holt die Zeichenpfad-Daten
  end
  /cd exch def

  /llx 99 def /lly 99 def /urx -99 def /ury -99 def % Initialisiert die
Begrenzungsrahmen

  true

  cd 2 get { % Verarbeitet die Pfaddaten
    exch {
      dup 32 eq { % Prüft auf Leerzeichen
        pop true
      } {
        82 sub cd 0 get sub % Berechnet den X-Wert des
Pfads

        dup llx lt { /llx exch def } if
        dup urx gt { /urx exch def } if
        pop false
      } ifelse
    } {
      neg 91 add % Berechnet den Y-Wert des Pfads

      dup lly lt { /lly exch def } if
      dup ury gt { /ury exch def } if
      pop true
    } ifelse
  } forall pop

  cd 1 get cd 0 get sub 0 llx 1 sub lly 1 sub urx 1 add ury rand top div 24 mul add
setcachedevice % Setzt die Cache-Geräteeinstellungen für
das Zeichen

  /domove true def
  true

  cd 2 get { % Zeichnet die Pfade des Zeichens
    exch {

```



```

        dup 32 eq {
            pop true
            /domove true def
        } {
            82 sub cd 0 get sub
            false
        } ifelse
    } {
        neg 91 add
        domove {
            moveto                                % Bewegt den Stift
            /domove false def
        } {
            lineto                                % Zeichnet eine Linie
        } ifelse
        true
    } ifelse
} forall pop

1 setmiterlimit 1 setlinejoin 1 setlinecap % Setzt Linienparameter
stroke
} bind def

/MakeKFont {
    10 dict begin
    /ItalicAngle exch def                        % Setzt den Schrägwinkel
    /PenWidth exch def                          % Setzt die Linienstärke
    /UniqueID exch def                          % Setzt die eindeutige ID der Schrift
    /FontName exch def                          % Setzt den Namen der Schrift
    /CharData exch def                          % Setzt die Zeichendaten

    /FontType 3 def                              % Setzt den Schrifttyp
    /FontMatrix
        [1 33 div
        0
        ItalicAngle neg dup sin exch cos div 33 div
        1 33 div
        0 0] def                                % Definiert die Schriftmatrix
    /Encoding StandardEncoding def              % Setzt die Standard-Kodierung
    /BuildChar {KHBUILDDict begin BUILDKHChar end} def % Definiert die BuildChar-
Prozedur
    /FontBBox {-10 -10 30 30} def              % Setzt die Begrenzungsrahmen der Schrift

    FontName currentdict end definefont pop
} bind def
end % userdict
%END

150 dict dup begin
";

% Ende des PostScript-Skripts
$end="
end
/KHrand 4295200 0.8 0 MakeKFont
";

% Definiert die Zeichendaten
$chars="
/hyphen [-6 6 (NRVR)] def
/bar [-4 4 (RBRb)] def
/periodcentered [-5 5 (RQRRSSRRQ)] def
/numbersign [-10 11 (SBLb YBRb LOZO KUYU)] def
/ampersand [-13 13 (\\0\\N[MZMYNXPVUTXRZP[L[JZIIYHWHUISJRQNRMSKSIRGPFNGMIMKNNPQUXWZY[[[\\Z\\
\\Y)] def
/currency [-11 11 (IIJKKOKUJYI[ [IZKYOYUZY[[ IIKJOKUKYJ[I I[KZOYUYZ[[])] def

```

```
/bullet [-2 2 (QPPQPSQTSTTSTQSPQP RQRRSSRRQ)] def
/slash [-7 7 (K^YF)] def
/backslash [-7 7 (KFY^)] def
/tilde [-8 8 (LTLRMPPOPUSWSXR LRMQOOUTWTRXP)] def
/bracketleft [-7 7 (OBOb PBbPb OBVB ObVb)] def
/bracketright [-7 7 (TBTb UBub NBUB NbUb)] def
/braceleft [-7 7 (TBRCQDPFPHQJRKSMOQQ RCQEQRISJTLTNSPORSTTVTXSZR[Q]Q_Ra QSSUSWRYQZP\
\^Q`RaTb)] def
/braceright [-7 7 (PBRCSDTFTHSJRKMQOSQ RCSESGRIQJPLNPQPURQTPVPXQZR[S]S_Ra SSQUQWRYSZT\
\^S`RaPb)] def
/less [-12 12 (ZIJRZ[)] def
/greater [-12 12 (JIZRJ[)] def
/asciitilde [-12 12 (IUISJPLONOPPTSVTXTZS[Q ISJQLPNNPQTTVUXUZT[Q(0)] def
/asciicircum [-11 11 (JTROZT JTRPZT)] def
/percent [-12 12 ([FI[ NFPHPJOLMMKMIKIIJGLFNFGSHVHYG[F WTUUTWTYV[X[ZZ[X[VYTW]] def
/at [-13 14 (WNVLTQKQOLNMPMSNUPVSVUUVS QKOMNPNOSUPV WKVSVUXVZV\T]Q]O\
\L[JYHWGTFQFNGLHJJILHOHRIUJWLYNZQ[T[WZYYZX XKWSWUXV)] def
/section [-8 8 (UITJUKVJVIUGSFQFOGNINKOMQOVR OMTPVrWTWVVXTZ
PNNPMRMTNVPXU[ NVSYU[V]V_UaSbQbOaN_N^O]P^O_) def
/dagger [-8 8 (RFQHRJSHRF RFRb RQTRbSTRQ LMNNPMNLLM LMXM TMVNXMVLTM)] def
/daggerdbl [-8 8 (RFQHRJSHRF RFRT RPQRSVRXQVSRRP RTRb R^Q`RbS`R^ LMNNPMNLLM LMXM
TMVNXMVLTM L[N\^P[NZL[ L[X[ T[V\^X[VZT[)] def
/A [-13 10 (XFVHTKQPOS LWIZG[E[DZDXEWFXYE XFVJUTT[ XFU[ T[TYSVRTPRNLQKRKTLWOZR[V[XZ]] def
/B [-12 12 (UGTHSJQOOUNWLZJ[ THSKQSPVOXMZJ[H[GZGXHWIXHY
OLNMMOKOJNJKLJMHOGRFXFZG[I[KZMXNTORO XFYGZIZKYMxN TOWPXQYSYVXYWZU[S[RZRXSU
TOVPWQXSXVWYU[)] def
/C [-10 11 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[)] def
/D [-12 11 (UGTHSJQOOUNWLZJ[ THSKQSPVOXMZJ[H[GZGXHWJWLXNZP[S[UZWXYTZOZLYIWGTFQE
THSKPUPXQZS[X[YZWwYSYKXIWHVHUJTJZKZNP)] def
/E [-12 11 (OJJJ[ TJIZ TJNKOLRL[N] RL[S] TLZL ]L^ LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN
XOZQ[S[UZWwX)] def
/F [-11 10 (OJJJ[ TJIZ TJNKOLRL[N] RL[S] TLZL ]L^ LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN
XOZQ[S[UZWwX)] def
/G [-11 12 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[ SXYX SYXX)] def
/H [-12 13 (OJZJ[ TJ\^J TMVM ]MMM ]MJZ M[J LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN XOZQ[S[UZWwX)]
def
/I [-7 8 (OJII[ TI\^I TJMJ ]MLL]M\^J LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN XOZQ[S[UZWwX)]
def
/J [-10 10 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[ SXYX SYXX)] def
/K [-11 10 (KJVJQQLTLVKXIWJVJVIIZIXJX JIJ[LXMXIXLM QLQMLTMVYK[J\^J\^JYM KJYLXMXIZIXLMX)]
def
/L [-9 9 (OHJHJ[ THIZ THNKOLRL[N] RL[S] TLZL ]L^ LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN
XOZQ[S[UZWwX)] def
/M [-13 12 (OJZJ[ TJ\^J TMVM ]MMM ]MJZ M[J LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN XOZQ[S[UZWwX)]
def
/N [-12 11 (OJZJ[ TJ\^J TMVM ]MMM ]MJZ M[J LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN XOZQ[S[UZWwX)]
def
/O [-11 12 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[)] def
/P [-11 11 (UGTHSJQOOUNWLZJ[ THSKQSPVOXMZJ[H[GZGXHWIXHY
OLNMMOKOJNJKLJMHOGRFXFZG[I[KZMXNTORO XFYGZIZKYMxN TOWPXQYSYVXYWZU[S[RZRXSU
TOVPWQXSXVWYU[)] def
/Q [-12 13 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[ SXYX SYXX)] def
/R [-11 12 (UGTHSJQOOUNWLZJ[ THSKQSPVOXMZJ[H[GZGXHWJWLXNZP[S[UZWXYTZOZLYIWGTFQE
THSKPUPXQZS[X[YZWwYSYKXIWHVHUJTJZKZNP)] def
/S [-10 10 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[)] def
/T [-12 12 (KJJJ[ TJIZ TJNKOLRL[N] RL[S] TLZL ]L^ LMNNMOKOJNJKLJMHOGRFXFZG[I[KZMYN
XOZQ[S[UZWwX)] def
/U [-12 12 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJPNNQMUMXNZP[)] def
/V [-12 12 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMLLULXMZP[R[UZWXXVXTWRURSSRU
```

```
WFUGRJP MNQMUMXNZP[]) def
/W [-12 13 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/X [-13 13 (XFVHTKQPOSLWIZG[E[DZDXEWFXY XFWJUTT[ XFU[ T[TYSVRTPRNQLQKRKTLWOZR[V[XZ]] def
/Y [-12 12 (XFVHTKQPOSLWIZG[E[DZDXEWFXY XFWJUTT[ XFU[ T[TYSVRTPRNQLQKRKTLWOZR[V[XZ]] def
/Z [-11 12 (OJZJ[ TJ\J TMVM ]MMM ]MJZ M[J LMNMOKOJNJLJKJMHOGRFXFZG[I[KZMYN XOZQ[S[UZWXX]]
def
/0 [-12 11 (OIHHJJKOLNNPOTNXLWIWJUJVKWLXPZQ[S[UZWXXVYXT WRJXJWKXMXOZQ[)] def
/1 [-10 11 (J[JZHIZXL] RJHJZHIZJ] J[JZHIZJ^ RJHJ[HZIZI[J^ RHL\H]) def
/2 [-11 11 (GHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/3 [-10 10 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/4 [-11 10 (OHJHJ[ THIZ THNKOLRL[N] RL[S] TLZL ]L^ LMNMOKOJNJLJKJMHOGRFXFZG[I[KZMYN
XOZQ[S[UZWXX]] def
/5 [-10 11 (GHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/6 [-10 11 (GHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/7 [-10 10 (JHL\H RJHJ[HZIZI[J^ RHL\H RJHJ[HZIZI[J^]) def
/8 [-10 10 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/9 [-10 10 (KHJJJLKNNOQOUNWMYKZIZGYFWFTGQJOMMQLULXMZP[R[UZWXXVXTWRURSSRU
WFUGRJP MNQMUMXNZP[]) def
/Agrave [-12 10 (XFVHTKQPOSLWIZG[E[DZDXEWFXY XFWJUTT[ XFU[ T[TYSVRTPRNQLQKRKTLWOZR[V[XZ
^U` T^Y^W]) def
/Acircumflex [-12 10 (XFVHTKQPOSLWIZG[E[DZDXEWFXY
XFWJUTT[ XFU[ T[TYSVRTPRNQLQKRKTLWOZR[V[XZ ^U^[\[\^W]) def
/Atilde [-12 10 (XFVHTKQPOSLWIZG[E[DZDXEWFXY XFWJUTT[ XFU[ T[TYSVRTPRNQLQKRKTLWOZR[V[XZ
^U_`^Y_`^V^)]) def
/Aogonek [-13 10 (XFVHTKQPOSLWIZG[E[DZDXEWFXY XFWJUTT[ XFU[
```