Automation made aasy
Egg, June 6, 2023: *On the occasion of a customer meeting, the question was discussed how the increasingly digitally incoming orders can be automatically equipped with a (virtual) barcode. This makes it unnecessary, for example, to print out the orders and manually affix a barcode sticker. Instead, a virtual barcode is affixed to the top right of the first page.*

The back story: printing, gluing, scanning...
More than a decade ago, it was not uncommon for orders to arrive by mail. The incoming receipts were barcoded and then scanned. The captured code on the receipts was transferred to the ERP software by means of a reading gun. From this point on, the keywording could be done automatically.

Over the years, receipts shifted from paper to the digital world. The majority of incoming orders are currently sent by e-mail or with correspondingly attached PDF files. As a result, more and more PDF files were printed out, only to have a barcode glued to them and fed to the scanner.

Automation with virtual barcodes
This manual process had to be optimized as far as possible. The options were discussed with the customer in a meeting that lasted about an hour. Two solutions emerged: firstly, a sticky note with a barcode would be attached to each incoming document, or secondly, the desired barcode would be "pixelated" above the first page.

The first option would have had the advantage that the documents would not have had to be "touched". However, in order to create sticky notes with barcodes, the appropriate barcode character set (Code39) would have had to be made available on all clients. The web server would have simply delivered the appropriate character set along with the web page. However, the sticky note has to be placed somewhere on the document and could possibly "cover" an area that contains central information.

In the second variant, the barcode is placed above the first page. This gives the first page of a file about 200 pixels more in height. However, this ensures that the barcode adhesive does not cover any other content on the documents at any time. Together with the customer, the second variant was chosen.

Implementation with script
The solution implemented for the customer is presented below:

```perl
#!/usr/bin/perl
use lib qw(/home/cvs/archivista/jobs);
use AVJobs;
use GD::Barcode::Code39;
my $start = 5000000;
my $field = "Titel";
my ($host,$db,$user,$pw,$doc) = @ARGV;
my $dbh = MySQLOpen($host,$db,$user,$pw);
if (HostIsSlave($dbh)==0) {
  my $sql = "use $db";
  $dbh->do($sql);
  $sql = "select Laufnummer,Seiten,Archiviert,ArchivArt from
archiv ".
    "where Laufnummer=$doc && ($field is null or $field=''
or $field=0)";
  my ($lnr,$seiten,$arch,$art) =
$dbh->selectrow_array($sql);
  my $ext = "png";
  $ext = "jpg" if $art==3;
  $ext = "tif" if $art==1;
  if ($lnr==$doc && $seiten>0 && $arch==0) {
    my $bc = getUnique($dbh,$start);
    if ($bc>=$start) {
      my $seite = ($doc*1000)+1;
      $sql = "select BildInput from archivbilder where
Seite=$seite";
      my ($pimg) = $dbh->selectrow_array($sql);
      my $fin = "/tmp/$db-$lnr-1.$ext";
      my $fbc1 = "/tmp/$db-$lnr-$bc-1.png";
      my $fbc2 = "/tmp/$db-$lnr-$bc-2.png";
      my $fbc3 = "/tmp/$db-$lnr-$bc-3.png";
      my $ftmp = "/tmp/$db-$lnr-2.$ext";
      my $fout = "/tmp/$db-$lnr-3.$ext";
      deleteFiles($fin,$fbc1,$fbc2,$fbc3,$ftmp,$fout);
      writeFile($fin,\$pimg);
      my $bco=GD::Barcode::Code39->new('*'.$bc.'*');
      my $img=$bco->plot();
```

```perl
        writeFile($fbc1,\$img->png());
        my $res=system("econvert -i $fbc1 --scale 2 -o
$fbc2");
        $res=system("convert $fbc2 -bordercolor White -border
50 $fbc3");
        $res=system("convert -append $fbc3 $fin $ftmp");
        $res=`edentify $fin`;
        $res =~ /([0-9]+)(x)([0-9]+)(dpi)/;
        if ($1>0 && $2 eq "x" && $3>0 && $4 eq "dpi") {
            $res=system("econvert -i $ftmp --resolution $1x$3 -o
$fout");
        } else {
            $res=system("mv $ftmp $fout");
        }
        my $cont="";
        readFile2($fout,\$cont);
        if ($cont ne "") {
            $sql = "update archivbilder set
BildInput=".$dbh->quote($cont).
            ",Bild='' where Seite=$seite";
            $dbh->do($sql);
            deleteFiles($fin,$fbc1,$fbc2,$fbc3,$ftmp,$fout);
            $sql = "update archiv set
$field=".$dbh->quote($bc)." where ".
            "Laufnummer=$doc";
            $dbh->do($sql);
            logit("stamp $bc sucessfully to $db and doc $doc
added!");
        }
    }
  }
}

sub deleteFiles {
  my @files = @_;
  foreach my $file (@files) {
    unlink $file if -e $file;
  }
}

sub getUnique {
  my ($dbh,$start) = @_;
  my $bc=0;
  my $plus="Name='BarcodeStamp' and Art='parameter' and
Tabelle='parameter'";
  my $sql0 = "select Inhalt from parameter where $plus";
  for (my $c=1;$c<=10;$c++) {
    ($bc) = $dbh->selectrow_array($sql0);
    if ($bc>0) {
      $bc++;
      my $bc2 = int($bc-(2*$bc));
      my $sql = "update parameter set
Inhalt=".$dbh->quote($bc2)." where ".
        "$plus and Inhalt>0";
      $dbh->do($sql);
      my ($bc3) = $dbh->selectrow_array($sql0);
      if ($bc3==$bc2) {
```

```
        my $sql = "update parameter set
Inhalt=".$dbh->quote($bc)." where ".
          "$plus and Inhalt<0";
        $dbh->do($sql);
        last;
      }
    } elsif ($bc==0) {
      my $sql = "insert parameter set
Inhalt=".$dbh->quote($start).",".
"Name='BarcodeStamp',Art='parameter',Tabelle='parameter'";
      $dbh->do($sql);
      my ($bc) = $dbh->selectrow_array($sql0);
      last $bc==$start;
    }
    sleep $c;
  }
  return $bc;
}
```

This script is to be placed in the following folder named 'bcadd':

**/home/data/archivista/cust/autofields**

In order for the script to be called after each captured page, a corresponding definition is needed in WebAdmin at 'Scan'. Concretely to our example a scan definition with the name 'bcadd' is to be created. Further a corresponding folder is needed in the processing folder:

cd /home/data/archivista/ftp/office/dbname
mkdir bcadd
chown -R ftp.users bcadd

This ensures that the scan definition 'bcadd' is always used when capturing documents via the folder 'bcadd'. So that the corresponding script is started during processing, the script 'bcadd.pl' must be entered in the scan definition for default fields. This activates the process.
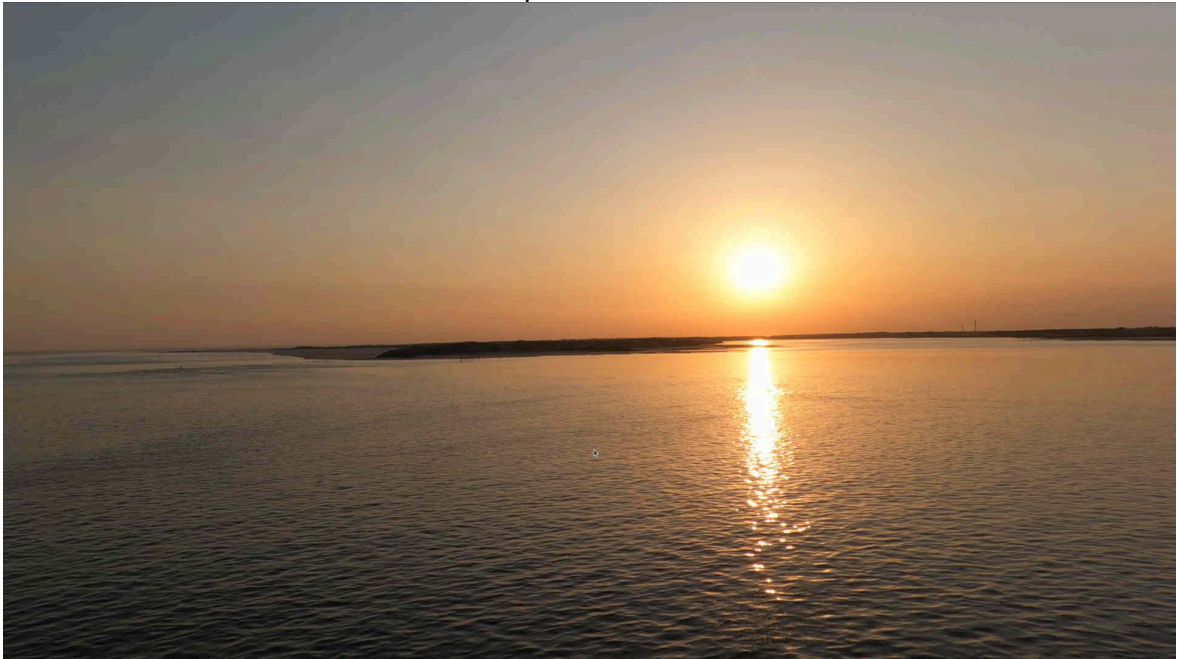
Conclusion: Little effort for high savings

With approx. 100,000 pages (50,000 orders with an average of 2 pages), which arrive at the customer's annually by mail, first of all the printing of these pages can be saved. At **druckkosten.de,** the savings per page are given somewhere between 6.3 and 13.8 cents (for the 50 cheapest models). On average, this

works out to about 10 cents (or centimes) per page. The printing costs per year thus amount to about 10,000 euros or francs.

In addition, there are the costs for the barcode stickers. A look at various web stores showed that minimum costs of approx. 3 cents/cents per label can be expected here. This results in further costs of 1500 francs/euro for the labels. The time required to print 100,000 pages can only be estimated very roughly. However, since the documents consist for the most part of a few pages, approximately 6 seconds per page can be expected. This results in 600,000 seconds or 166 working hours. For the application of the adhesive (including the search for a free space on the document), approx. 6 seconds per process can also be expected. This results in an additional 83 working hours. In addition, there is the time required for scanning. A good scanner should be able to process about 2000 pages per hour with a decent document scanner, which means another 50 hours for scanning.

In contrast, the time needed to move the PDF files from the mails to the share folder has to be calculated. A small side note: This process could be completely automated with the mail module, but this step has yet to be taken by the customer. The time required for each process is probably about the same as that required for applying a glue. Consequently, the time saved is 216 hours (166 hours for printing and 50 hours for scanning). At an estimated 25 francs/euro (including fixed costs), this results in 5400 francs/euro in personnel costs (216*25).

In total, the costs should amount to 16,900 francs/euro per year, which will no longer be incurred with the virtual barcodes. On the other hand, there are about 10 hours for the development and implementation time. This results in total costs of 2200 Euro/Franc if the project is handled by us at an hourly rate of 220.– Francs/Euro. This means that in the first year a reduction in costs of 14,700 francs/euro can be realized. Extrapolated to 10 years (1*14700+9*16900) this results in the proud sum of 166'800.– Francs/Euro.

Of course, the process can be further automated by reading the order number directly from the document and storing it in a corresponding field in ArchivistaDMS. This allows the customer's ERP solution to automatically assign and process the document. However, this is where the ERP provider is required, as the ERP solution must be adapted with this optimization. In the case described here, no adjustments to the ERP software are necessary and yet the corresponding savings result. Enjoy!

Further information: *The customer works with the ArchivistaBox Matterhorn. The customer's archive currently contains 7.7 million documents with a total of 11.4 million pages. On the hard disk, 1.1 TByte are required for the database.*