Accounts payable management with ArchivistaDMS

Egg, 3 December 2018: *The last few months have been marked by a lot of hard work for customers. There was not much time to expand the ArchivistaBox. Release 2018/XI has been available for a few days now. First and foremost, a new creditor management module is optionally available for ArchivistaDMS. Furthermore, the Fujitsu fi-7140 and fi-7240 are now supported. Last but not least, some bugs were solved with ArchivistaERP and ArchivistaDMS, this is also part of product maintenance. In the following, the creditor management is presented.*



Step 1: Activate script

As part of a new project, the aim was to enable all incoming vendor invoices to be processed electronically. There were several levels of rights to be implemented. The secretariat records the invoices (scanning or file upload) and assigns them to the departments. There they can be examined and electronically verified. When registering, the management sees all the approved items and authorises them for accounting, with which the invoices are released for payment.

Sounds simple, and it is. With the following code ArchivistaDMS is extended by a complete creditor management:

```
#!/usr/bin/perl

use lib qw(/home/cvs/archivista/jobs);
use AVJobs;
my $host = shift;
my $db = shift;
my $user = shift;
my $pw = shift;
my $doc = shift;
my $dbh=MySQLOpen($host,$db,$user,$pw);
checkChange($dbh,$doc,$user);
$dbh->disconnect();

sub checkChange {
  my ($dbh,$doc,$user) = @_;
  my $docit = "where Laufnummer=$doc ".
    "and DokTyp2Nr=4 and (Gesperrt='' or Gesperrt is null)";
  my $sql = "select Laufnummer from archiv $docit";
  my ($lnr)=$dbh->selectrow_array($sql);
```

```
  if ($lnr>0) {
    if ($user eq "buha") {
      $sql = "select SachbearbeiterHidden,".
        "GeschaeftsleitungVisum from archiv $docit";
      my ($sach,$hidden,$visum) =
$dbh->selectrow_array($sql);
      if ($visum==0) {
        if ($sach ne $hidden) {
          $sql = "update $db.archiv set
Sachbearbeiter=".$dbh->quote($sach).
            ",SachbearbeiterHidden=".$dbh->quote($sach).
            ",SachbearbeiterVisum=0,SachbearbeiterDatum=Null
".
            "where Laufnummer=$doc";
          $dbh->do($sql);
        }
      } else {
        if ($sach ne $hidden) {
          $sql="update archiv set
Sachbearbeiter=".$dbh->quote($hidden)." $docit";
          $dbh->do($sql);
        }
      }
      $sql = "select BuchhaltungVisum,BuchhaltungDatum,".
        "GeschaeftsleitungVisum from archiv $docit";
      my ($vis,$dat,$gl) = $dbh->selectrow_array($sql);
      if ($gl==1) {
        if ($vis==1 && $dat==0) {
          $sql = "update archiv set
BuchhaltungDatum='".SQLStamp1().
            "' where Laufnummer=$doc";
          $dbh->do($sql);
        } elsif ($vis==0 && $dat>0) {
          $sql = "update archiv set BuchhaltungVisum=1 where
Laufnummer=$doc";
          $dbh->do($sql);
        }
      } else {
        if ($vis==1) {
          $sql = "update archiv set BuchhaltungVisum=0 where
Laufnummer=$doc";
          $dbh->do($sql);
        }
      }
    } elsif ($user eq "gl") {
      my $sql = "select GeschaeftsleitungDatum from archiv
$docit".
        " and GeschaeftsleitungVisum=1 and
BuchhaltungVisum=0";
      my @row = $dbh->selectrow_array($sql);
      if (int($row[0])==0) {
        $sql = "update archiv set
GeschaeftsleitungDatum='".SQLStamp1().
          "' where Laufnummer=$doc";
        $dbh->do($sql);
      }
    } else { # Sachbearbeiter
```

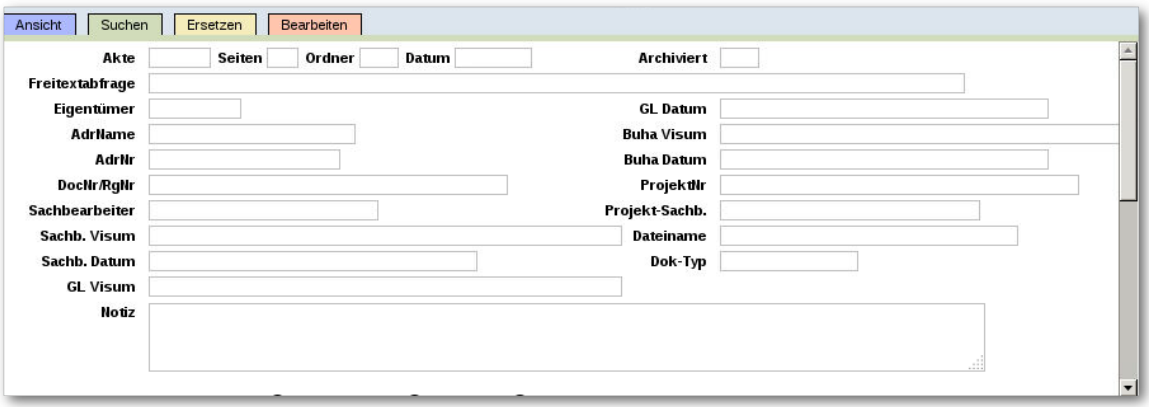
```
      my $sql = "select Sachbearbeiter from archiv $docit";
      my @row = $dbh->selectrow_array($sql);
      if ($row[0] eq "$user") {
        my $sql = "select
SachbearbeiterVisum,SachbearbeiterDatum ".
          "from archiv $docit and GeschaeftsleitungVisum=0
and ".
          "BuchhaltungVisum=0";
        my ($vis,$dat) = $dbh->selectrow_array($sql);
        if ($vis==1 && $dat==0) {
          $sql = "update archiv set
SachbearbeiterDatum='".SQLStamp1().
            "' where Laufnummer=$doc";
          $dbh->do($sql);
        } elsif ($vis==0 && $dat>0) {
          $sql="update archiv set SachbearbeiterVisum=1
where Laufnummer=$doc";
          $dbh->do($sql);
        }
      } else {
        $sql="update archiv set SachbearbeiterVisum=0 where
Laufnummer=$doc";
        $dbh->do($sql);
      }
    }
  }
}

sub SQLStamp1 {
  my @t = localtime( time() );
  my ( $stamp, $y, $m, $d, $h, $mi, $s );
  $y = $t[5] + 1900;
  $m = $t[4] + 1;
  $m = sprintf( "%02d", $m );
  $d = sprintf( "%02d", $t[3] );
  $stamp = $y . "-" . $m . "-" . $d . " 00:00:00";
  return $stamp;
}
```

The script can be activated under the name of the database (e.g. archivista.pl) in WebAdmin under ‘Administrating Jobs’.

Step 2: Customize mask

In order for the script to work, plus/minus has to be created the following mask:



Sorry: *The mask was created for a customer in Switzerland, hence the German*

names. The translated labels are: BuHa = Accounting, Sachbearbeiter = Clerk

Step 3: Creating and Assigning SQL Definitions

In order for individual users to find the correct documents when entering ArchivistaDMS, corresponding SQL definitions must be entered in WebAdmin:

Secretary/Accounting: DocTyp2Nr=4 and AccountingVisum=0 and (((Clerk is zero or Clerk=”) or (ManagementVisum=1 and AccountingVisum=0)) order by Date desc, Serial number desc

Clerk: DocTyp2Nr=4 and ClerkVisum=0 and ManagementVisum=0 and Clerk='[user}’ order by Date asc

Management: DocTyp2Nr=4 and ((clerk<>”””” and ManagementVisum=0)) order by Datum asc

The corresponding SQL definitions are to be assigned to the users — and ArchivistaDMS is already extended by a full-fledged accounts payable management.

Conclusion: Automation made easy

ArchivistaDMS can be easily automated with ‘Manage Jobs’ via WebAdmin. In the above example, the Perl programming language was used. The script languages PHP and Python can also be used. Other languages can also be integrated optionally.

If the above lines seem ‘Spanish’ to you, you are welcome to contact us, corresponding modules will be implemented for customers on every ArchivistaBox.