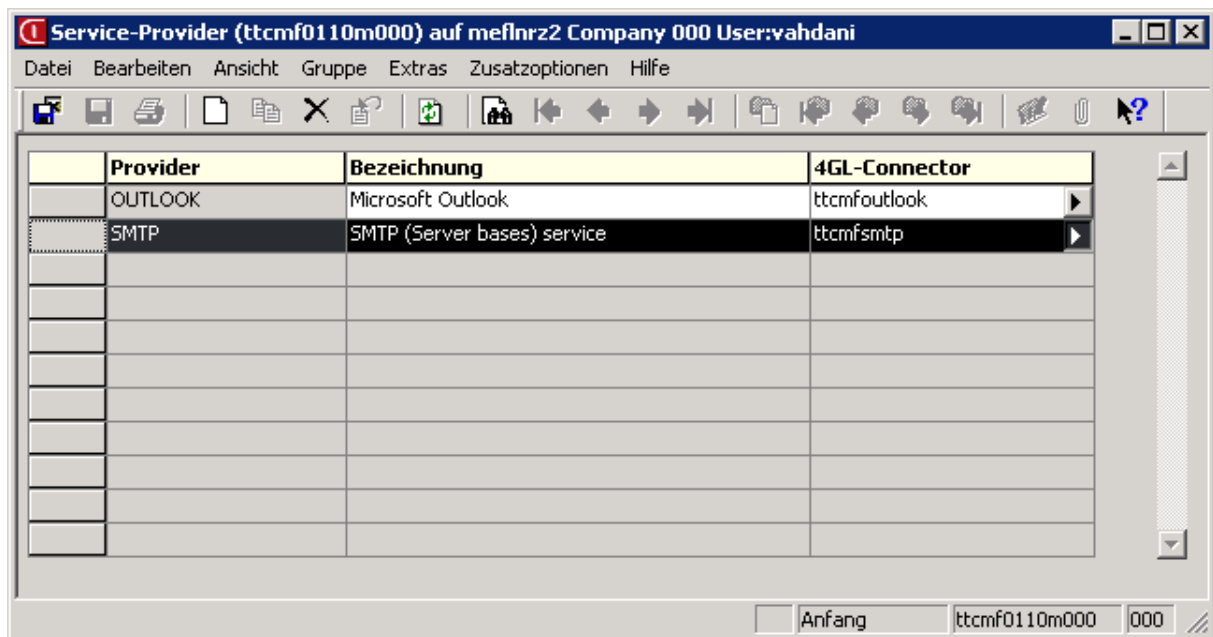
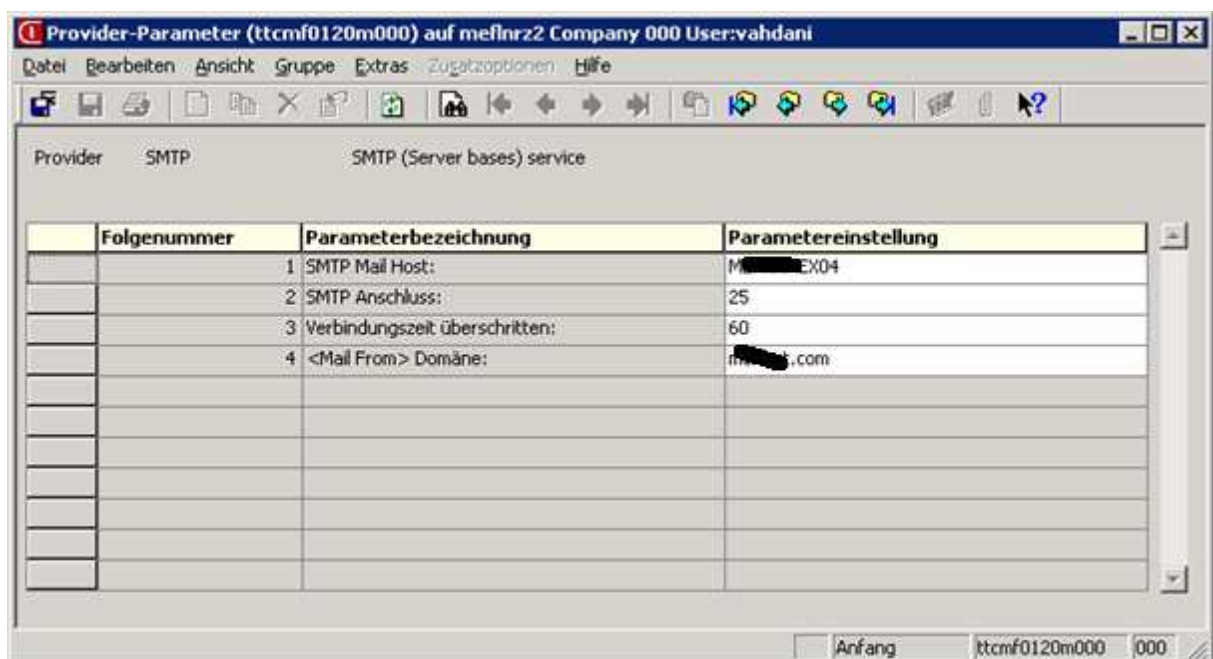


1- Define Service Provider



2- Set Service Provider Parameter:



| Service-Name | Bezeichnung | Provider | Aktiv | Logging | Speicherpfad Meldung | Dateityp |
|--------------|-----------------------------|----------|-------------------------------------|-------------------------------------|----------------------|------------------|
| OUTLOOK | Outlook Service | OUTLOOK | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | \${BSE}/cmf/outlook | Klartext Unicode |
| SMTP | SMTP (Server Based) Service | SMTP | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | \${BSE}/cmf/smtp | Klartext Unicode |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Anfang ttcmf0130m000 000

C Adressart nach Service (ttcmf0140m000) auf meflnrz2 Company 000 User:vahdani

Datei Bearbeiten Ansicht Gruppe Extras Zusatzoptionen Hilfe

| | Adressart | UI erforderlich | Adresserkennung | Service-Name | Papierart |
|--|-----------|-------------------------------------|-------------------------------------|------------------------------------|-----------|
| | SMTP | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | OUTLOOK ▶ Outlook Service | A4 ▶ |
| | SMTP | <input type="checkbox"/> | <input checked="" type="checkbox"/> | SMTP ▶ SMTP (Server Based) Service | A4 ▶ |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Anfang ttcmf0140m000 000

Send Mail Method 1: print to SMTP device:

SMTP-Device:

Daten Ausgabegerät (ttaad3100s000) auf meflnrz2 Company 000 User:vahdani

Datei Ansicht Extras Zusatzoptionen Hilfe

Papierart nach ... Systeme

Ausgabegerät: SMTP

Bezeichnung: CMF SMTP Print Device

Art des Ausgabegeräts: Datei neu schreiben

Locale: ISO-8859-1 Western Europe - 8 bit

☐ Datei im Austauschformat XML

Drucker

Treiber:

Druckerwarteschlange:

Papierart:

Linker Rand: 0 ☐ Vorschub pro Seite

Datei

Treiber:

Shell-Befehl:

Konvertierungsprogramm: ttcmfprint

Argument: smtp

Pfad: \${BSE}/tmp/cmfsmtpprint

☒ Ändern der Ausgabedatei zulässig.

Seitenlänge: 66

Ändern ttaad3100s000 000

Print:

Drucker auswählen (ttstpspopen) auf meflnrz2 Company 000 User:vahdani

Drucker Abfrage Nachricht senden Datei

Ausgabegerät: SMTP CMF SMTP Print Device

Optionen

☐ Vorschau

Ausgabedatei: homayoun.vahdani@...de

Vorlage:

Weiter

Abbrechen

voreinst. verwenden

voreinst. speichern

sprache auswählen...

Hilfe

Bericht ttaad340001000 Daten Ausgabegerät

ttstpspopen 000

EMAIL will be sent with report as rtf-attachment!

Send Mail Method 2: Send per Program:

Step 1 register Mail in Mails Table:

```
*****
* tcxtldll0001 0 VRC B61C a mefl
* EMail functions
* vahdani
* 19.10.10 [10:19]
*****
* Script Type: Library
*****
        table    ttcxtl010          | EMail Requests

function extern tcxtldll0001.register.email.request(
                                domain tcnama      i.fnam,
                                domain tcmcs.str50  i.fadr,
                                domain tcnama      i.tnam,
                                domain tcmcs.str50  i.tadr,
                                domain tcmcs.str50  i.subj,
                                ref    string      i.body)
{
dllusage

enddllusage
        db.set.to.default(ttcxtl010)
        tcxtl010.guid = uuid.generate$()
        tcxtl010.fnam = i.fnam
        tcxtl010.fadr = i.fadr
        tcxtl010.tnam = i.tnam
        tcxtl010.tadr = i.tadr
        tcxtl010.subj = i.subj
        tcxtl010.body = i.body
        tcxtl010._comprnr = 0
        db.insert(ttcxtl010, db.retry)
}

function extern tcxtldll0001.process.email.requests()
{
        activate("otcxtlsendmail")
}
***** end of source *****
```

Step 2: Send all registered mails:

```
*****
* tcxtlsendmail 0 VRC B61C a mefl
* Mail sender
* vahdani
* 19.10.10 [09:37]
*****
* Script Type: 0
*****

*****
* tdzzzmailtest 0 VRC B61C a mefl
* Test CMF
* vahdani
* 18.10.10 [16:56]
*****
* Script Type: 0
*****

*****
* CMF Sample Code
*****
* Script Type: 0
*****
```

dllusage

Desc: This code sample demonstrates the use of some of the Infor
ERP LN eMessage Connector commands
described in the CMF 4GL Functional Design. It accepts a
filename (which points to an RTF
document), Infor ERP LN eMessage Connector service name,
recipient name and recipient address

(SMTP only) as input arguments. It then uses the Infor ERP LN eMessage Connector commands to

construct a message then send it to the indicated recipients.

Pre: None.

Post: The file is sent to the appropriate Infor ERP LN eMessage Connector service.

In: The four program arguments accepted are:

sourcefile: The name of the file (RTF format) to be

sent.

service: This is the name of the service that will be used to send the message.

This must be name of a valid Infor ERP LN eMessage Connector service

name: This is the (human readable) name of the recipient.

address: This the address where the file should be sent. This could be an email address, a fax number, etc.

Out: None.

enddllusage

| | |
|------------------------------|--------------------------------------|
| table ttcxtl010 | Mail Requests |
| string recipient.address(80) | Recipient address |
| string recipient.name(80) | Human readable name of the recipient |
| string sourcefile(256) | Filename of the source file |

#include <bic_cmf>

function main()

{

suspend(2000)
|Switch to company 0
comprn.check(0)

db.retry.point()

select tcxtl010.*
from tcxtl010 for update
selectdo
 send.mail()
 db.delete(ttcxtl010, db.retry)
 commit.transaction()
endselect

commit.transaction()

}

function send.mail()

{

domain ttcmf.prov service |Infor ERP LN eMessageConnector service for transporting messages

long start.stat
long stop.stat
long send.stat
long mid |CMF message id
long fid

|Get program arguments
sourcefile = creat.tmp.file\$(bse.tmp.dir\$())
fid = seq.open(sourcefile, "w")
seq.puts(tcxtl010.body, fid)
seq.close(fid)

service = "SMTP"

|Build message XML header
mid = create.xml.header()

|Send message
if mid <>0 then
 start.stat = cmf.startService(service,2)
 if start.stat = 0 then
 |Infor ERPLN eMessage Connector service started
 send.stat = cmf.send(mid,service)
 stop.stat = cmf.stopService(service,2)
 endif

 if send.stat = 0 then
 |Message was sent
 endif

```

endif

file.rm(sourcefile)
}

function long create.xml.header()
{
    long    message.ID
    long    from.ID
    long    to.ID
    long    attachment.ID
    long    ret

    message.ID = cmf.create()

    ret = cmf.setClass(message.ID, "CMF.NOTE")
    ret = cmf.setSubject(message.ID, tcxtl010.subj)

    |Set FROM recipient
    from.ID = cmf.createRecipient(message.ID, ttcmf.role.from)
    ret = cmf.setRecipientName(from.ID, tcxtl010.fnam)
    ret = cmf.setRecipientAddress(from.ID, tcxtl010.fadr)
    ret = cmf.setRecipientType(from.ID, "SMTP")
    |In reality of course the from address could be filled from the
    |Address book.

    |Set TO recipient
    to.ID = cmf.createRecipient(message.ID, ttcmf.role.to)
    ret = cmf.setRecipientName(to.ID, tcxtl010.tnam)
    ret = cmf.setRecipientAddress(to.ID, tcxtl010.tadr)
    ret = cmf.setRecipientType(to.ID, "SMTP")
    ret = cmf.setRecipientResponsibility(to.ID, "TRUE")

    |Set attachment
    attachment.ID = cmf.createAttachment(message.ID)
    |ret = cmf.setAttachmentBody(attachment.ID, "TRUE")
    ret = cmf.setAttachmentBody(attachment.ID, ttyeno.yes )
    |ret = cmf.setAttachmentMIME(attachment.ID, "application/rtf")
    ret = cmf.setAttachmentMIME(attachment.ID, "text/plain")
    ret = cmf.setAttachmentFileName(attachment.ID, sourcefile, "Attachment")

    return(message.ID)
}
|***** end of source *****

```