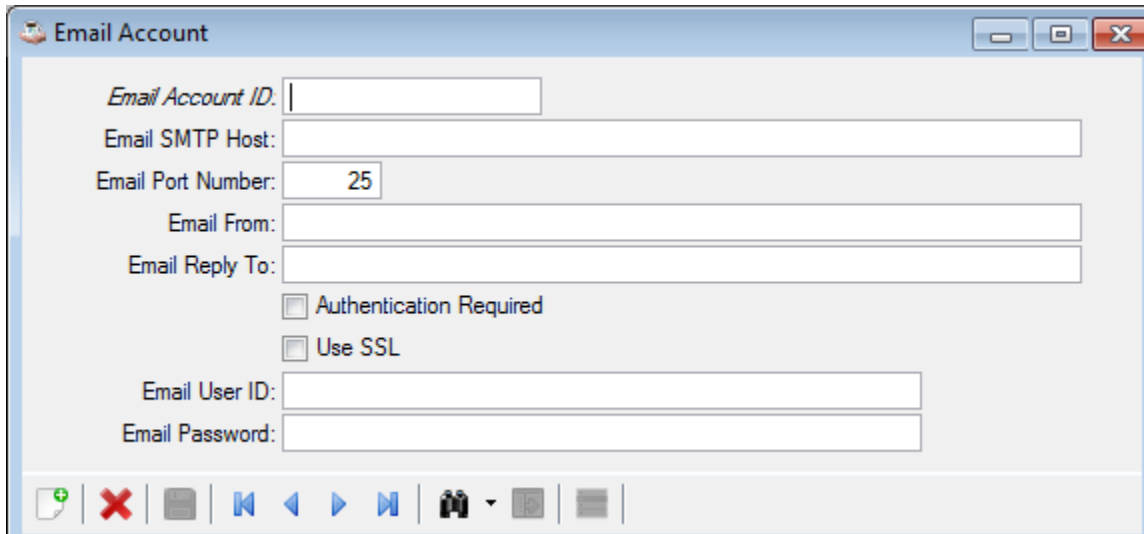# Sending Email via Barista Callpoint

This should bring up the client's default mail client:

```
e$=callpoint!.getColumnData("MY_TABLE.MY_EMAIL")
if e$<>""
        use java.awt.Desktop
        use java.net.URI
        s$="Developer Email"
        n!=null()
        m!=new URI@("mailto",e$+"?subject="+s$,n!)
        d!=Desktop@.getDesktop()
        d!.mail(m!)
endif
```
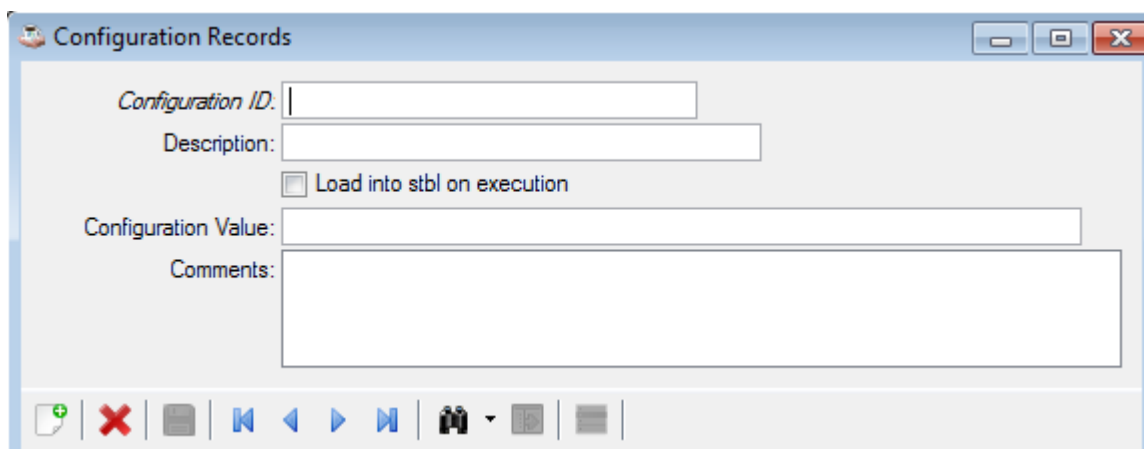
There is also a Barista public program that the Barista Document Management System uses, but could work for you as well. It can be called like this:

```
call stbl("+DIR_SYP")+"bac_sendmail.bbj",
: mail_account$,
: mail_from$,
: mail_replyto$,
: mail_to$,
: mail_cc$,
: mail_bcc$,
: mail_subject$,
: mail_message$,
: mail_files$,
: mail_status,
: mail_status$
```

If mail_account$ is not empty, then it is used to lookup the record in the ADM_EMAIL_ACCT table, otherwise it tries to get the value from stbl("+MAIL_ACCT") to do the lookup. The email account record can be configured via Barista Administration/Document Management/Email Accounts:

If no email account record is found, then the SMTP mail host, and SMTP user and password, if required, can be set via configuration records (aka STBLs) under Barista Administration/Maintenance/ Configuration Records, but the port and SSL flag are set to 25 and 0 (no SSL), respectively.

Multiple email addresses can be separated by commas or semicolons, but multiple file attachments have to be separated by semicolons.

The program uses a copy of the Email class found in <basis>/utils/email/email.bbj, which of course is also at your disposal with or without Barista.
bac_sendmail.bbj will report send failures interactively in a message dialog and set:

```
mail_status=err; mail_status$=errmes(-1)
```