In thin client mode how do I run a local external application like the Windows Calculator or Microsoft Excel?

There are several ways that an application running on a server can launch an application on the client.
The three most common are :

1. CALL CLIENT "SYSTEM", "C$RUN" or "C$SYSTEM" (to open an application)
2. Use "C$EASYOPEN" or "C$DESKTOP" (to open a file with the associated application)
3. CALL CLIENT "MYPROG" (to run a COBOL program)

1. CALL CLIENT "SYSTEM", "C$RUN" or "C$SYSTEM"
Use the SYSTEM, C$RUN and C$SYSTEM library routines to run an operating system command line as if it was executed from a command prompt, shell, script, batch file, or shortcut.**
SYSTEM runs the command synchronously so that the program waits for the command to complete before continuing.
C$RUN runs the command asynchronously so that the program continues to run in parallel with the command.
C$SYSTEM combines the functionality of SYSTEM and C$RUN and allows a command to be run synchronously or asynchronously depending on a "flags" parameter.
Add the CLIENT clause to the CALL statement in order to run the command on the client-side.
The isCOBOL framework ignores the CLIENT clause when not running in thin client mode.
So CALL CLIENT is safe to use in programs that run in both thin client and local modes.
For example, to launch the Windows calculator program:

```
call client "C$RUN" using "calc"
```

If Microsoft Excel is in the client-side PATH then you could launch it with:

```
call client "C$RUN" using "excel.exe"
```

However, usually excel.exe is not in the PATH. So you would need to specify a full path.
In order to identify whether excel.exe is installed and its disk location you could read the registry with the registry library routines such as DISPLAY_REG_OPEN_KEY and DISPLAY_REG_QUERY_VALUE.
However, if you have a specific file to open, it's easier to use method #2 below.

2. C$EASYOPEN or C$DESKTOP
To open a file on the client with the default associated application, you can use C$EASYOPEN

and include the name of the file to be opened.
If you want to specify the location of the file on the client, or you're using a version prior to 2015R2, you can copy the pdf file to the client with C$COPY, then call C$EASYOPEN using the CALL CLIENT syntax:

```
    call "c$copy" using "/myapp/reports/custlist.pdf"
                        "@[DISPLAY]:c: mpeportsc
ustlist.pdf"                giving retCode.   if retCode = 0      call client "c$easyopen" using
 "c: mpeportscustlist.pdf"                          giving retCode.   end-if
```

If it doesn't matter where the file is located on the client and you're running in version 2015R2 or later, you can pass a flag to C$EASYOPEN to tell it to copy the file to the client's temporary folder and open it there, without having to call C$COPY:

```
    call "c$easyopen" using "/myapp/reports/custlist.pdf"                    0
                giving retCode.
```

You could also use C$DESKTOP in place of C$EASYOPEN.
C$DESKTOP was introduced in 2017R2 and does multiple desktop operations, including open, edit, and print files using the associated applications.
When you use C$DESKTOP the runtime creates a temporary copy of the file on the client machine, then opens it.

```
   call "C$DESKTOP" using
 CDESKTOP-
OPEN                        "/myapp/reports/custlist.pdf"                1                giving retCode.
```

3. CALL CLIENT "MYPROG"
If a program named MYPROG.class is available in the client-side class path, then a program running on the server can call it using CALL CLIENT "MYPROG" and can pass it input, output or input-output parameters of any type by reference or by value just as if it was calling the program on the server.
For example, you could compile the following program and add it to your client-side jar file or have your installer put it on the client in the class path specified when your users launch the thin client:

```
    ID DIVISION.   PROGRAM-ID
. MYCALC.   PROCEDURE DIVISION.      call "C$RUN" using "calc".      goback.
```

You could call this program from your application with:

```
    call client "MYCALC"
```

Note:
There are differences in the behavior of CRUN/CSYSTEM between isCOBOL and ACUCOBOL-GT.
For example, calls relying on UNIX/Linux shell syntax for redirection of standard streams should be made using the SYSTEM routine.
See "isCOBOL implementation of CALL SYSTEM and the iscobol.system.exec property" for more information and workarounds for CRUN/CSYSTEM if you need platform specific functionality.