NOPEN$ - Open Windows Directory

The NOPEN$ routine is used to open a directory on the host operating system (normally Windows) so that the NLIST$ routine can then be used to list the directory contents.

1. Invocation

To open the directory code:

```
CALL NOPEN$ USING flag file area
```

where flag is a PIC 9(4) COMP field or literal which is 0 if file is the basic direct FD for the directory schema file produced using $SFMAIN; or 1 if file is a template filename string, terminated with LOW-VALUES. The area parameter is a PIC X (140) work-area used by NOPEN$, NLIST$ and NCLOS$.

2. STOP Codes and Exception Conditions

No STOP codes are returned by NOPEN$.

The following EXIT codes may be returned by NOPEN$:

EXIT code	$$COND	Description
24001	01	NOPEN$ has suffered an I/O error on the directory schema file.
24002	02	The directory schema file passed to NOPEN$ was not found or of the wrong type.
24003	03	NOPEN$ is not available on this host operating system.
24004	04	The directory passed to the NOPEN$ routine was invalid. The error code has been returned in $$CRES.

3. Programming Notes

The NOPEN$ routine must be used in conjunction with the NLIST$ and NCLOS$ routines.

The NOPEN$ routine has been modelled on the traditional OPEN$ routine. Note that NEOPN$ is also available as an extended version of NOPEN$.

The PIC X(140) work-area must not be used for any other routines apart from subsequent NLIST$ and NCLOS$ calls; until the final NCLOS$ routine has completed. In particular, the work-area must not be used for any nested NOPEN$ calls.

When used on GSM (Windows) the “template filename” may include wildcards. For example:

```
C:\notes\*.*   Subsequent NLIST$ calls will return all files
```

```
C:\notes\*.txt  Subsequent NLIST$ calls will only return .txt
                files

C:\notes\       Subsequent NLIST$ calls will fail and return an
                error!!!
```

If the flag value is 1 the maximum length of template filename string, including the terminator of LOW-VALUES, is 80 characters.

Important Note: Following a successful call of NOPEN$, the first call of NLIST$ must be **before** any subsequent calls to NOPEN$. In particular, if it is required to list 2, or more, different folders concurrently then this structure should be avoided:

```
CALL NOPEN$ USING 1 filename1 AREA1 * Open first folder
CALL NOPEN$ USING 1 filename2 AREA2 * Open second folder
CALL NLIST$ USING AREA1 DE1         * Get first file in first folder
CALL NLIST$ USING AREA2 DE2         * Get first file in second folder
```

Instead, code:

```
CALL NOPEN$ USING 1 filename1 AREA1 * Open first folder
CALL NLIST$ USING AREA1 DE1         * Get first file in first folder
CALL NOPEN$ USING 1 filename2 AREA2 * Open second folder
CALL NLIST$ USING AREA2 DE2         * Get first file in second folder
```

4. Examples

[EXAMPLES REQUIRED].

5. Copy-Books

None.

6. See Also

NLIST$	List Windows Directory
NLIS2$	List Windows Directory (Normalised File Type)
NCLOS$	Close Windows directory
NEOPN$	Extended Open Windows Directory
NELIS$	Extended List Windows Directory
NELI2$	Extended List Windows Directory (Normalised File Type)
NECLS$	Extended Close Windows Directory
NXOPN$	Specialised Open Windows Directory
NXLIS$	Specialised List Windows Directory
NXCLS$	Specialised Close Windows Directory
OPEN$	Open Global volume
LIST$	List Global volume
CLOSE$	Close Global volume