



Legac-E Education

Rationale for using the INCLUDE Statement

With the demise of physical punched card readers and the storage of Job Control Language (JCL) in Partitioned Data Sets (PDSs) on disk it begged the question; “Why use Catalogue Procedures?” After all the JCL could be modified directly using the installation standard editor, and there was little risk of JCL errors induced by physical damage to a card image.

One argument for the continued use of procedures was that it reduced disk storage space if multiple jobs used the same basic JCL sequence. A secondary argument might be that if the procedure is correctly structured, then the same basic JCL could be used for production and development which would make life easier when programs or applications are migrated to the production environment.

Even if procedures are used there is still the possibility that a given set of files (data sets) could be referenced in multiple jobs. This increases the maintenance effort if these files change for any reason; such as they need to be created under a new name or their attributes are amended when new versions are produced.

The contention might be that it would be easier to maintain JCL in such a situation if the files DD statements were held in one place rather than in multiple JCL sets. The use of JCL INCLUDE groups makes this possible whether procedures are being used or not. The syntax of the relevant JCL statement is:

```
// INCLUDE MEMBER=group_name
```

The set of JCL statements referenced by group_name must appear as a member of a PDS or PDSE within the procedure library search chain for the job which includes the statement.

Even before the Object Orientated premise of re-usability, programmers used COPY (Assembler and COBOL) or INCLUDE (PL/I) statements to reference code which may be common to multiple programs, so why not use similar techniques in JCL.