

ASAp/DCM

Product Integration

Transparent Borders.

A powerful feature of ASAp/DCM is the blurred delineation between product functions, also known as transparent borders. This concept allows the overlapping functions within ASAp/DCM to complement each other.

An example of this is the ability to manually submit a job with an operator command, as this would ordinarily be thought of as an ASAp function. Now consider automatically issuing the same operator command from DCM as the result of a message match. We have thus created a framework whereby a scheduled job's end is detected by the automated operations component, which then schedules the next job in the suite with a user-defined operator command.

Utility Programs.

Some of the utility programs supplied with ASAp/DCM lend further to this concept. Among the utility programs that can be placed in any job include:

- **#ASAPCMD:** Issue an operator command from within a job based upon condition codes.
- **#ASAPMSG:** Issue a message to the system log and operator's console.
- **#ASAPTMR:** Invoke an intentional wait within a job step of a user-specified duration.
- **#ASAPENQ:** Wait for a data set to be available for update before proceeding.

Use of these utility programs along with the complementing functions of ASAp/DCM can assist in creating a very flexible schedule. No single approach to scheduling jobs is required.

A linear job flow can be defined strictly using ASAp schedule records. It can also be accomplished by having each finishing job automatically submit the next job by using an imbedded submit command (#ASAPCMD).

Alternatively, each job can issue a message at completion indicating whether it was successful or not (#ASAPMSG). Message filters within DCM could then trap the appropriate messages and continue the schedule as required.

The point to be drawn from this is that ASAp/DCM's flexible design means your scheduling options may only be limited by the bounds of imagination.