

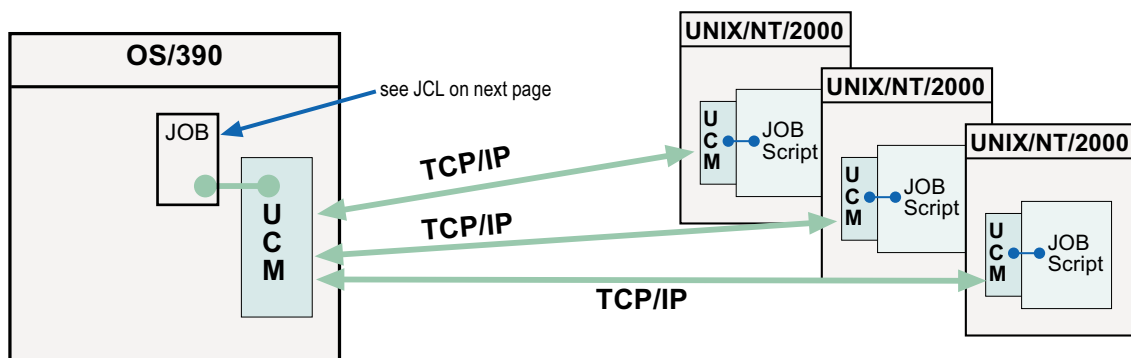
# Organisation-Wide Job Control with UCM

You are already working with a job scheduler under OS/390. You now intend to integrate other platforms. Perhaps you intend to execute scripts under Unix, tasks under Windows/NT/2000 or a job under AS/400? With other words, you want transmit data from OS/390 to other systems/platforms and backward, you intend to start activities on other systems, and you want to receive information about results of those activities.

UCM will enable you to start any activity on a system connected via TCP/IP. Whoever is able to submit a OS/390 job can also perform activities on other systems. It is only required to start UCM within a OS/390 job (step) and to enter the names of the target system and the job to be executed there. UCM will do everything else for you: it will contact the distant system, start the job, wait for results, send them back and file them in the respective DD cards of the OS/390 job/step.

To prevent tampering by unauthorized operators, UCM will interrogate the SAF interface. This means that, for example RACF, will control which operators/users are authorized to perform tasks on "remote" systems. UCM can perform tasks that already exist on the target system, or it can transfer the scripts to be executed before execution takes place.

As described above, UCM can be activated via a batch job under OS/390. This OS/390 job serves as a representative or deputy of the remote activity. It is executing under OS/390, or better, it is waiting as long as the remote task is active. When the remote task ends, UCM delivers the information about the execution on the remote system back to the deputy job. This information (stdout, stderr) will be filed under DD statements before the deputy job finishes.



## Simple solution to a well-known problem

UCM will make your OS/390 job scheduler a "cross-platform control system", be it OPC, CA-7 or any other scheduler. You can control processes on distributed target systems from a single place. This will enable your production control team to plan and control scattered processes without the need of additional specialized training.

- To embed a remote activity into the workflow of the mainframe scheduler, a representative job needs to be created under OS/390 that will evoke UCM.
- A representative or deputy can be scheduled as any other job on the mainframe.
- A deputy job can transfer a script or just ask for the execution of an existing script.
- A deputy job may have predecessors, requirements, successors, etc.

After the representative job has been submitted and the UCM step has started, the script file will be transferred to the target systems, if required, and will then be executed.

An OS/390 representative job will wait until the task is finished at the target system and will receive the results.

UCM completes your existing mainframe scheduler by automatic routing and execution functions. This way your mainframe scheduler is enabled to integrate and schedule central and distributed activities. Your output management system will archive execution information without any further

steps being needed.

Mainframe users have made considerable investments to ensure reliable scheduling and control. UCM will make any OS/390 scheduler a cross-platform scheduler. OS/390 production control teams take over planing and scheduling of activities on other systems. All functions of the mainframe scheduler will still exist for remote jobs:

- Job monitoring: Through a query for the deputy
- Creation of schedules: A deputy can be scheduled as any other job
- Job Forecast: A deputy will be recognized as any other
- A deputy may have predecessors, requirements, successors, etc.
- Output filing: via the corresponding deputy.

UCM offers a flexible and easy to handle solution.

### Simple solution: all you need is piece of JCL

```
//RUNCLNT JOB ...
//,,,
//DATA DD *
JUST SOME TEST DATA
THIS IS THE LAST RECORD
//HCPCNTL DD *
TRANSMIT (DATA)
  TO (172.16.1.98)
  PORT (5555)
  FILE (C:\TEST.DAT)
  TYPE (CHAR);
EXECUTE (DIR C:\TEST.*)
  ON (172.16.1.98)
  PORT (5555);
RECEIVE (STOREIT.HERE)
  FROM (172.16.1.98)
  PORT (5555)
  FILE (C:\TEST.DAT)
  TYPE (CHAR);
//STOREIT.HERE DD SYSOUT=*,DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
```

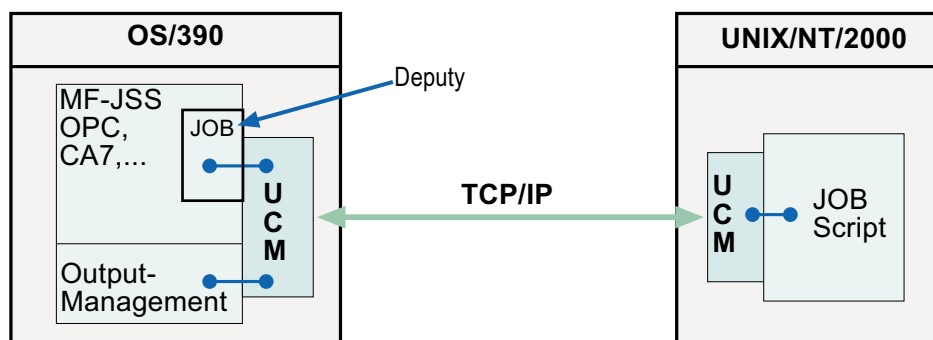
Could be any data set under OS/390

Transmit data from OS/390 to any other system

Execute a program at a remote computer

Fetch data from a remote system and store it under OS/390

### Seamless integration with existing environment



UNTERNEHMENSBERATUNG  
SOFTWARE SERVICE GMBH