

Comparison between MultiBatch and NetBatch - Plus



Contents

1. Introduction
2. Interface
3. Scheduler
4. Jobs
5. Parallelism
6. Calendar
7. Status Reporting
8. Parameter Database
9. Advantages
10. Quick Comparison Checkbox

Introduction

Both HP's NetBatch and Insider Technologies Ltd's MultiBatch products are designed to provide a Batch Scheduler for the HP Non-Stop server environment. Both products do accomplish this, however each uses a different methodology to achieve its aim, resulting in each product having quite different capabilities.

NetBatch and MultiBatch both consist of a PATHWAY interface to a scheduling engine, as well as a command line utility. The MultiBatch product also includes a GUI.

NetBatch requires the user to develop and maintain a JCL, written in NBEXEC or TACL, to control the sequencing of jobs. This requires knowledge of both these products and will incur a maintenance overhead. Although MultiBatch can use both TACL and NBEXEC as executors, natively there is no MultiBatch JCL. All job run parameters, sequencing and dependencies are maintained by the MultiBatch application. The user only requires knowledge of MultiBatch and the batch schedule.

Both products have a calendar facility, enabling automation to be controlled on a time/date basis. Whilst NetBatch applies this control to the batch executor, MultiBatch is the executor, and thus applies this control to the batch jobs themselves.

Although both products have been around for some time, MultiBatch is still being enhanced, with new facilities and features being developed. Insider Technologies Ltd have no plans for stopping development or withdrawing support. HP classes NetBatch as a 'Mature' product, hence no further development or enhancement will be undertaken.

Interface

Both NetBatch and MultiBatch come as standard with a command line interface (BATCHCOM for NetBatch and BCOM for MultiBatch) and a PATHWAY interface (at least in the case of MultiBatch and NetBatch-Plus). The major advantage of MultiBatch is its GUI interface. NetBatch is a mature product so it is unlikely that a GUI will be developed.

The MultiBatch GUI interface provides access to all the configuration and monitoring functions available from the PATHWAY interface, with the convenience of a Windows GUI. In addition, the MultiBatch GUI provides enhanced batch schedule configuration and status monitoring capabilities, with improved checking and validation. In addition to these standard features, MultiBatch is closely integrated with other products from Insider Technologies Ltd, such as Reflex 80:20 and Sentra/SIP, providing enhanced monitoring and reporting capabilities.

Scheduler

The HP NetBatch scheduler allows individual NetBatch jobs to be configured to run at a particular time, on a particular date and in a particular order. These jobs can have dependencies on preceding jobs, based on their successful completion. Any other checks that may be required, such as checks for the successful completion of individual jobs within the batch schedule, or execution of multiple jobs in a schedule in parallel, must be catered for in the JCL for each NetBatch schedule. The NetBatch application provides a batch-scheduling environment that allows the job executor, such as TACL or NBEXEC, to be controlled in this manner. It is the JCL that must perform the control and monitoring of individual jobs within the batch schedule.

It is possible to migrate a NetBatch schedule from one node to another, however this requires extensive manual administration using BATCHCOM ALTER commands as well as amendments to the schedule's base cold load file. Not only is this a resource intensive operation, but it also requires extensive knowledge of the schedule.

ITL's MultiBatch application is a batch executor in its own right. This means that its scheduling capabilities can be applied to the various jobs, and their individual components, within a schedule. There is no need for a complex JCL to be maintained when using MultiBatch. Each job within the schedule is configured within MultiBatch, together with any parameters that may be required, any dependencies on other jobs in the schedule, and any other relevant information. The MultiBatch scheduler will also allow for configuration and monitoring of parallel jobs without the need for complex checking within a JCL. All of this configuration information is held within the MultiBatch database and can be amended with ease using the PATHWAY or GUI. A further facility is the ability of MultiBatch to produce a virtual copy of the batch schedule. The schedule may be prototyped to ensure that any conditional steps execute as expected.

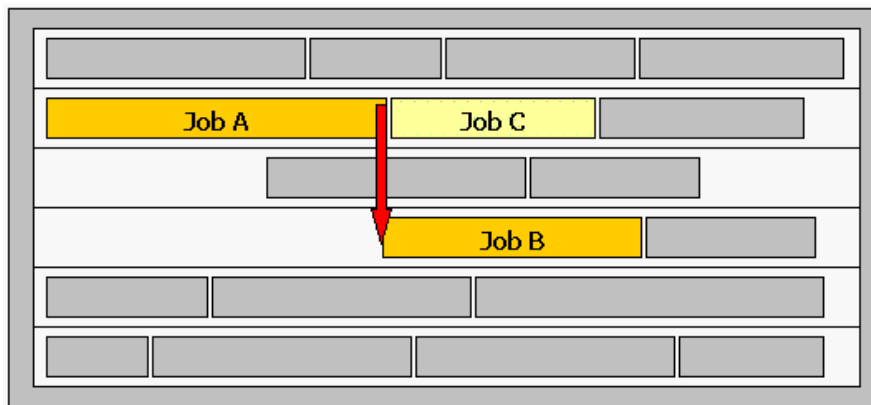
MultiBatch has the facility to migrate a batch schedule from one node to another. A batch schedule may be developed on a test node and, once completed, may be migrated to the production system. This is accomplished using an automated procedure that is built using simple substitution rules, for example change \TEST to \LIVE, \$TESTDAT to \$LIVEDAT etc.

Jobs

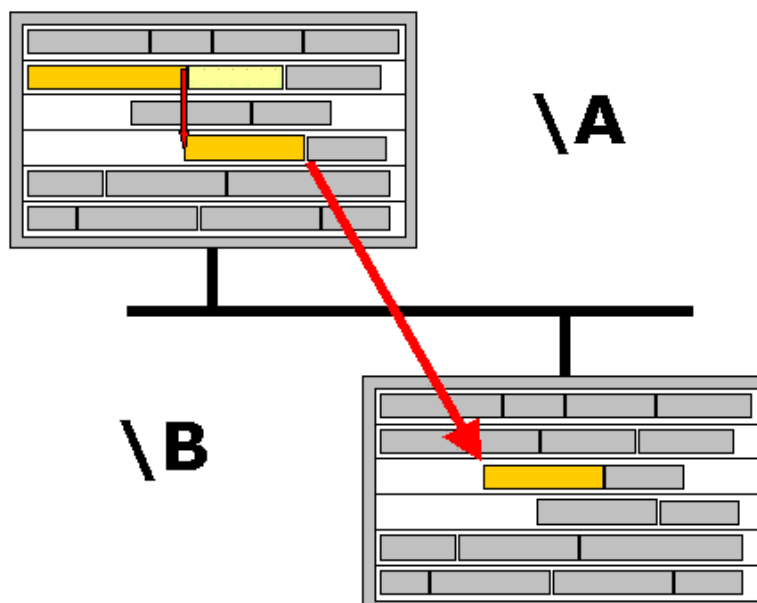
The NetBatch scheduler allows control of a single entity or job and this can be configured to run at certain times, dependent on the outcome of preceding jobs and each job can be configured to be dependent on a maximum of 20 previous jobs. The NetBatch scheduler cannot control any sub-steps within a job, or other requirements such as parallel processing, the JCL for each job must do this. Although NetBatch is not particularly well suited to large, monolithic batch schedules, it is ideally suited to 'bulk submit' and ad-hoc batch processing, such as program compilations and report generation.

If a NetBatch schedule fails it can be difficult to re-start it from the point of failure without further complicating the JCL. This is because the JCL controls the execution of the individual steps within the schedule; NetBatch just controls the execution of the JCL. Should a job fail mid-schedule it will either have to be detected by the JCL and the appropriate action taken, or the JCL will have to be amended to resume at the failed job, and the schedule re-started.

The MultiBatch scheduler allows individual jobs to be split into multiple parts, Jobs, Segments, Units and Steps. A job can be split into multiple segments; both of these are capable of having 10 dependencies on other Jobs and Segments within the batch schedule.



In the above example Job B will commence the moment Job A completes. Job C will also start running alongside Job B in a separate CPU. These dependencies may also be configured across schedules and Tandem nodes, allowing multiple jobs and segments to be linked.

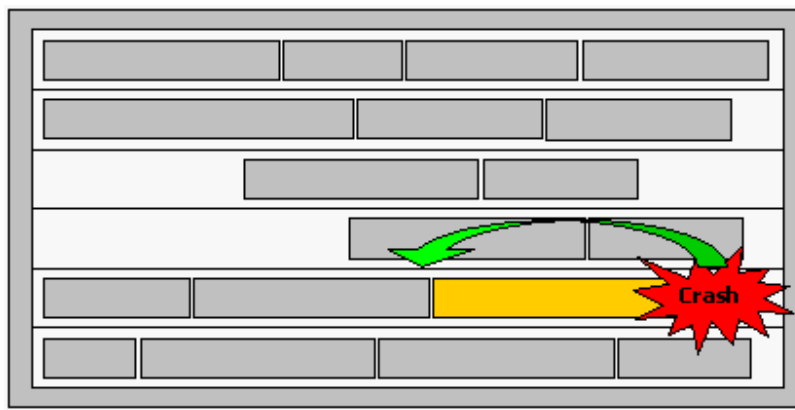


Jobs

In this example, a Job on Node \B only starts when the specified Job on Node \A completes.

Each segment is further split into Units and Steps, these representing the function of the batch job; the actual task that will be performed or program that will be executed. It is possible for each unit to have two steps, allowing for the running of two inter-related tasks; for example an EMS distributor may be started that is used as a feed for a statistical analysis application. MultiBatch is ideal for controlling large, monolithic processing runs, such as accounting month-end processing, where there is little change to the structure of a batch schedule.

MultiBatch jobs are split down to step and unit level, it is therefore possible to restart failed schedules from the point of failure, and not have to re-run the entire schedule, or amend any complex JCL. By tightly integrating a MultiBatch schedule with TMF, a fail-safe, checkpoint on restart schedule can be configured.



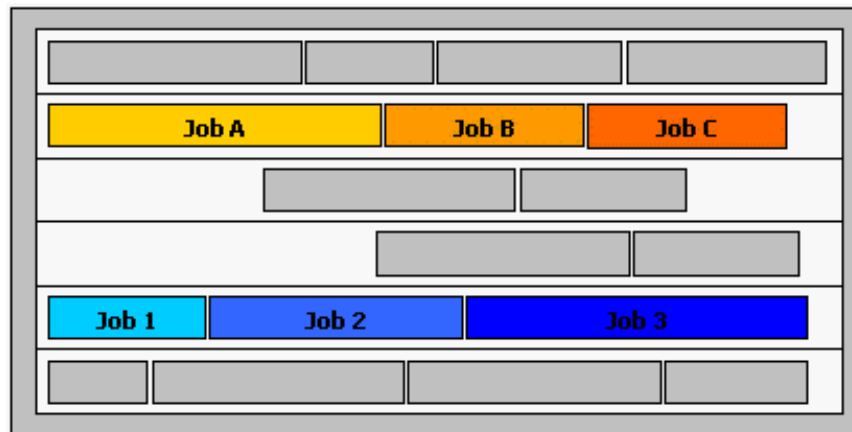
In this example, the MultiBatch job has crashed. It can be configured to automatically resume from the beginning without the need to amend any JCL.

MultiBatch also has the capability of monitoring jobs that are over-running. It is possible for each job in the schedule to be configured with an expected run time. This enables a virtual overview of the schedule to be defined, indicating what job is expected to be running at a particular point in time. MultiBatch can compare the actual position of the schedule against the expected position and alert the operator to any jobs that are taking longer to complete than normal.

Parallelism

NetBatch schedules can be configured with Job level dependencies, however these are usually run one after the other. This makes it difficult to spread the processing load in parallel over multiple CPU's without having to first write complex JCL macros.

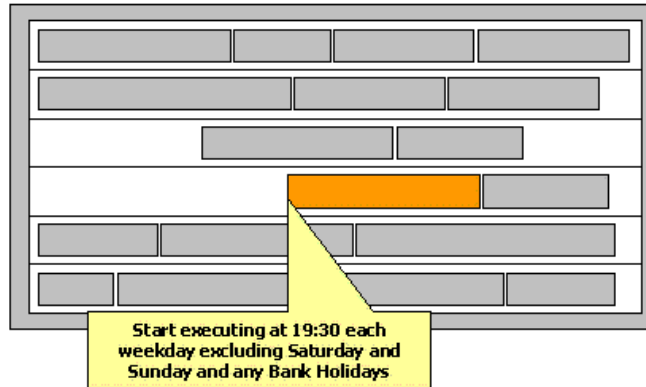
The advantage of the multi-part structure of batch jobs within the MultiBatch scheduler is the capability it has for parallel processing of jobs. A schedule can be configured in MultiBatch that allows a number of jobs to execute at once, spread over multiple CPU's, that have inter-related dependencies. This allows maximum use to be made of the available processing resource, and can significantly reduce the amount of time taken for a batch schedule to complete.



In this example, Job A, Job B and Job C run one after the other in CPU 1 whilst at the same time Job1, Job 2 and Job 3 run concurrently in CPU 4.

Calendar

Both products have a calendar facility. This allows certain key dates that may affect the running of a batch schedule to be configured in advance, and maintained with little or no impact on the actual batch schedule. NetBatch schedules that are made up of an edit file that is then compiled into an unstructured calendar file. The calendar file cannot be viewed, and as its source is changed using the editor, any amendments cannot be tracked. The MultiBatch calendar is held in the MultiBatch database and is amended by means of an application screen. All changes to the calendar are subject to security clearance, and will be audited.



In this example a Job has been configured to execute at a specified time on specified days, taking into consideration any holidays that have been configured in the MultiBatch calendar.

Status Reporting

Using the NetBatch command utility BATCHCOM, or the NetBatch-Plus PATHWAY, it is possible to see the status of each batch schedule. The status information will show the current state of the schedule including its next run date, if it is running or has failed, and history information. It is not possible to view the status of the individual jobs that are contained in the JCL, and thus the actual failure point. NetBatch does write out status information, however this is mostly written to a log file or the spooler, making monitoring of a schedule by a 3rd party enterprise manager or operations console product rather difficult.

Because each MultiBatch schedule is split down to the actual jobs (units and steps) and these are under the control of MultiBatch and not a JCL, it is possible to get status information right down to unit level (step). This enables the status of each individual task to be viewed, giving a more accurate overview of the batch schedules progress. MultiBatch writes job status and progress messages to the Tandem EMS subsystem. This allows each MultiBatch schedule to be closely monitored by any 3rd party product that is capable of acting upon EMS events. It is also closely integrated with other products from ITL, such as Reflex 80:20 and Sentra, allowing further alerting and automation, and the production of detailed MI reports.

Parameter Database

Both products make use of a database for storing run time parameters and assignments. This greatly reduces the amount of maintenance required when amending a schedule. Groups of parameters and assignments that are used by several batch jobs only need to be declared once and can be attached to jobs many times over. MultiBatch is shipped with various migration utilities that allow a parameter database that may have been created on a development system to be exported to a production system in a single step, vastly reducing the maintenance overhead.

Advantages

Basically, there are a number of advantages to using MultiBatch in preference to NetBatch and these are summarised below:

Maintenance & Development

NetBatch is a mature product and will not be subject to further development. New facilities and functionality are still being developed for MultiBatch, driven by customer requirements.

Interface

NetBatch has either a command line or Pathway interface. MultiBatch has both of these and a GUI.

Scheduler

NetBatch controls the execution of a batch schedule and relies heavily on a JCL to control the actual jobs within a schedule. MultiBatch controls the execution of each element of a batch schedule. Individual jobs and their elements are more visible within MultiBatch. The attributes of each job within the schedule can be viewed and altered using MultiBatch application screens and all changes are subject to the appropriate security clearance and are recorded in an audit log. NetBatch jobs can be amended using an editor and are thus not subject to any security vetting or auditing.

Failures

NetBatch again relies on a complex JCL to detect failures in the elements of a schedule. Because MultiBatch controls each element of a batch schedule the detection of failures and the restarting of a schedule at the correct point is a simple matter.

Parallel Processing

Once again, NetBatch relies on the JCL for parallel processing and monitoring whilst it is an easy task to configure a MultiBatch schedule to handle this.

Reporting

NetBatch status information is written to a log file or spooler, making it difficult to integrate with a 3rd party enterprise manager. MultiBatch writes status information in the form of EMS events, allowing monitoring by 3rd party tools. Additionally, it is closely integrated with other products from ITL for more in-depth monitoring.

In addition to the above, a migration tool is available that will allow a NetBatch schedule to be converted to a MultiBatch schedule.

Quick Comparison Checkbox

Competitive Points		NetBatch	MultiBatch
Single point of operations control across Expand network		N	Y
Requires no command-line syntax knowledge to configure/initiate jobs		N	Y
GUI for operations control		N	Y
GUI for job configuration		N	Y
GANTT style job status/progress GUI		N	Y
GUI for user configuration management, and security control		N	Y
Calendar facility, includes holiday, etc. specification		Y	Y
GUI for calendar set-up and management		N	Y
Job Control Integrity check (pre-run)		N	Y
Application/Job API to start/stop control jobs by application		N	Y
Job Dependency checks (job release control)		Y	Y
Ditto, re device dependence checking		Y	N
Designed and Coded for NonStop (Fault Tolerant running)		Y	Y
TMF transaction protected "safe auto-recovery from fail" facility	Jobs may be safely auto-restarted after failure of CPU	N	Y
Able to launch work to be managed by TACL control program/file		Y	Y
Job streams may be run in parallel		Y	Y
Job steps within streams may be run in parallel		N	Y
Able to launch OSS programs		Y	Y
Able to accept OSS filenames of full-length		Y	Y
Does not require Shell or Shell knowledge to launch OSS jobs		N	Y
Security audit log of changes		N	Y
Optional block-mode green-screen supplied as standard for fast bulk entry		N	Y
Ability to prepare a Virtual Copy of Schedule		N	Y



Insider Technologies is a UK-based software and services company quality certificated to ISO 9001:2000 and TickIT. Operating in the Financial and Messaging markets, it provides Service Management, Tracking, Bespoke Software and Information Mediation solutions. A cross section of our customers would include Banking and Financial Services, Telecommunications Providers and Government and Military Institutions.

For details about the full range of products and services available from Insider Technologies Limited, please contact our Product Development Centre at:

Insider Technologies Limited

Spinnaker Court
Chandlers Point
Broadway
Salford Quays
MANCHESTER, M50 2YR
United Kingdom
Tel: +44 (0)161 876 6606
Fax: +44 (0)161 868 6666
E-Mail: support@insidertech.co.uk
Web Site: <http://www.insidertech.co.uk>

business partner

