



MultiBatch Systems Management White Paper



Table of Contents

Introduction.....	3
MultiBatch Coldload.....	4
MultiBatch Housekeeping	7
MultiBatch Site-Swap.....	10
MultiBatch Migration Strategy	12
MultiBatch Application Processing.....	14
Summary	15

Introduction

The document complements the “MultiBatch White Paper” document (which provides a technical overview of the product). If you are unfamiliar with MultiBatch then it is recommended that the “MultiBatch White Paper” is read first.

What this document provides

This paper provides an overview on how MultiBatch can be used in various NonStop systems management roles and for facilitating the rollout of MultiBatch schedules, via the Migrator utility.

The following topics will be covered:

- Coldload
- Housekeeping
- Site-Swap
- MultiBatch Migration
- Application Processing

Who should read this document?

The document is aimed at people who are familiar with NonStop systems and application management functions and who wish to automate these procedures via MultiBatch.

It provides examples on how to use MultiBatch in the above roles, with the potential to reduce down-time for your NonStop systems and in the process, **improve SLAs** (Service Level Agreement) for your customers.

This document also provides examples on how to use the MultiBatch ‘Migrator’ utility to roll-out new or amended schedules to other MultiBatch environments, e.g. Production, Backup, Testing, or Trialling.

MultiBatch Coldload

This section describes the benefits MultiBatch can provide for coldloading, warm-starting or performing a shutdown of a NonStop system.

It provides an example of a 'system coldload' MultiBatch schedule. Similar principles can be applied for configuring a schedule to warm-start or shutdown the NonStop system, e.g. Shutdown RDF, Expand, TMF, run online dumps, etc.

Traditionally, the coldload of a NonStop system invokes TACL based 'coldload' files. Contents of these files include commands to start: TMF, DISKS, TCP/IP, LANS, EXPAND, RDF, Spoolers, etc. with each command line being invoked sequentially.

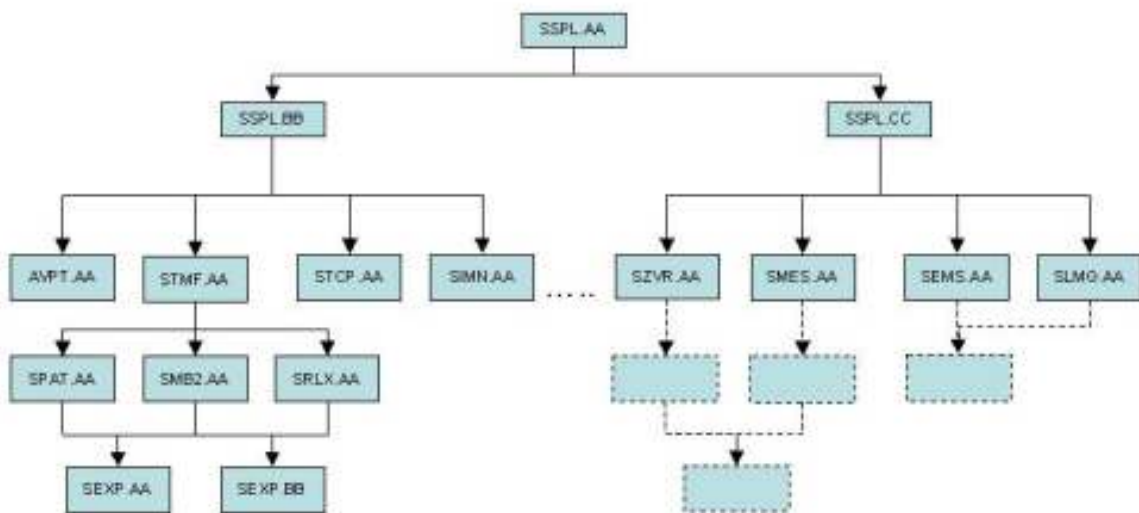
Although each command line in the startup files can be invoked sequentially and in turn load up the NonStop system, the parallelism of MultiBatch can speed up the coldload operation. This is especially beneficial when, following a site-swap, the 'new' Production node needs to be started up as soon as possible.

Batch flow

A flowchart example of how MultiBatch can be used in a coldload situation is provided below. A table containing the individual segment names and associated functions is provided on the next page. The dotted lines and boxes represent additional segments which can be incorporated as required.

The underlying units can execute the required site-swap commands using different user ids, e.g. the unit for STMF.AA (Start TMF) is run as 255,255, whereas the units for starting the various Pathways may be run under different 'manager' ids.

This is just a small example of the benefits of using MultiBatch to perform a coldload. As can be seen, the parallel approach of MultiBatch will facilitate the rapid coldload of a NonStop system, thereby reducing downtime and improving SLAs for customers.



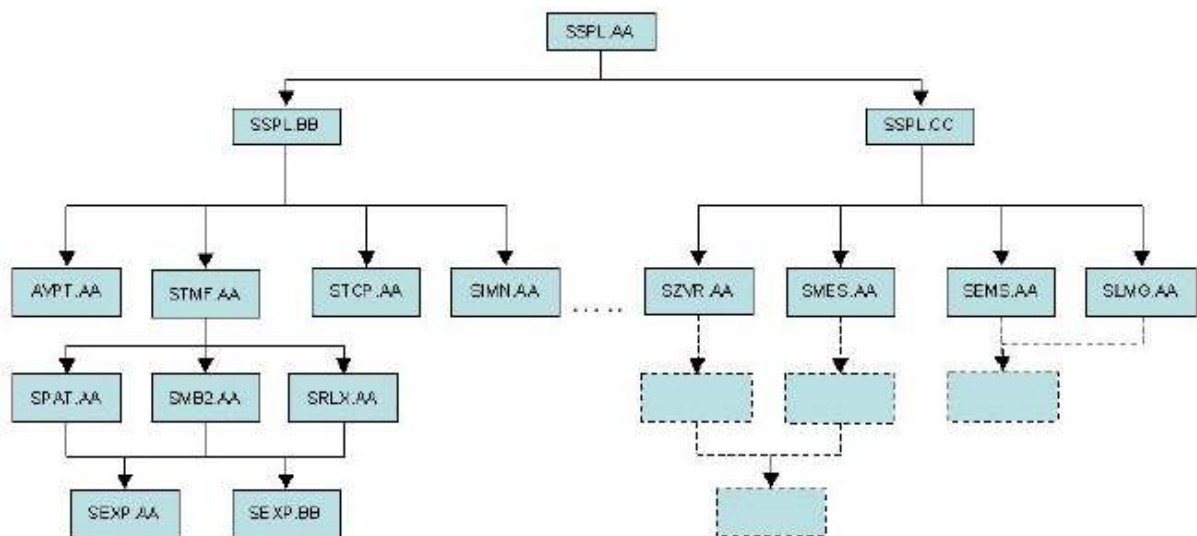
Segment Description

This is just a small example of a MultiBatch schedule for performing a coldload of a NonStop system. Other jobs and segments have not been specified, e.g. to: Start RDF, TMDS, OSS, Alternate Collectors, etc. but these can easily be incorporated into additional segments and units as required.

The 'user id' is configured within the MultiBatch step configuration, allowing individual units to execute under different users.

If a user id is not specified, then the units will run as the owner of the BMON when it is coldloaded, e.g. 255,255.

SEGMENT ID	FUNCTION	EXECUTING USER ID
SSPL.AA	Start spooler \$SPLS	255,255 (SUPER.SUPER)
SSPL.BB	Start spooler \$BPLM	255,222 (SUPER.OPERATOR)
SSPL.CC	Start spooler \$CPLM	255,222 (SUPER.OPERATOR)
STMF.AA	Start TMF	255,255
SVPT.AA	Start Viewpoint	100,255 (VWPT.MANAGER)
STCP.AA	Start TCP/IP	255,255
SIMN.AA	Start IMON	255,255
SZVR.AA	Start ZSERVER	255,255
SMES.AA	Start Measure	255,255
SEMS.AA	Configure EMSCTRL	255,255
SLMG.AA	Start LAN Managers	255,255
SPAT.AA	Start Application Pathway	125,255 (APPL.MANAGER)
SMB2.AA	Start MultiBatch Pathway	250,250 (MBAT.OWNER)
SRFX.AA	Start Reflex 80:20 Pathway	080,020 (REFLEX.OWNER)
SEXP.AA	Start Expand Line \$EXPA	255,255
SEXP.BB	Start Expand Line \$EXPB	255,255
Other jobs.segments	----	----



UNIT USER ID

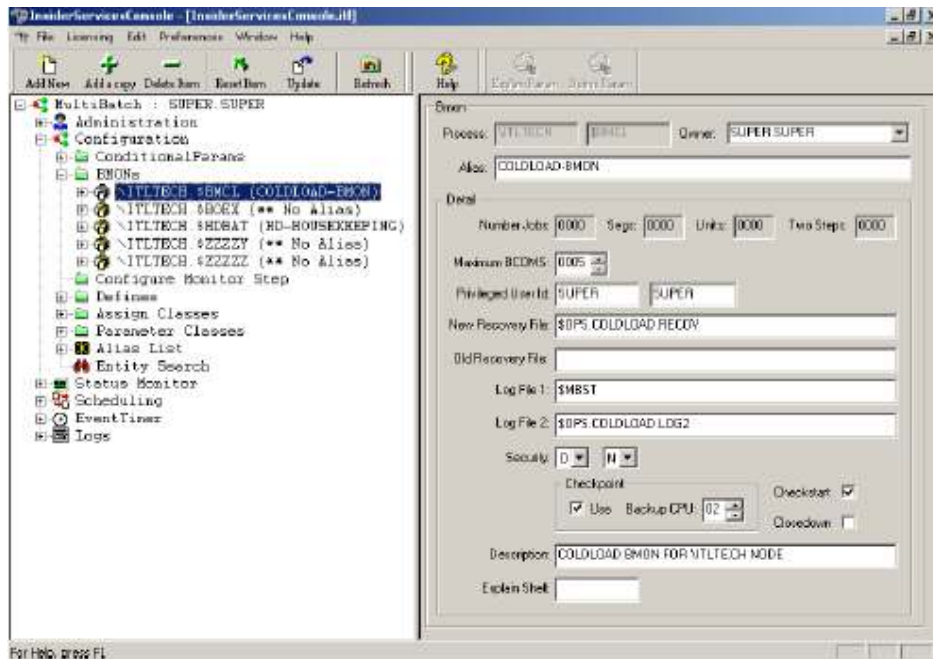
To enable units to run using different user ids, the BMON program uses a privileged procedure call which requires the BMON program to be licensed. Some users may be wary about installing third party licensed code, therefore ITL ship an unlicensed BMON program without this user id capability.

If a user id is not specified, then the units will run as the owner of the BMON when it is coldloaded, e.g. 255,255.

BMON Startup

Once a schedule has been configured, the MultiBatch BMON is then coldloaded via the system 'startup' file. The results from the BMON Build process need to be logged to a disk file (Type E, Rec 150), as the Spooler will not have been started at this point.

An example is provided below, demonstrating the use of a disk 'log' file (Log File 2) for recording the results of the BMON Build process.



After setting the system clock and adding a transition to the daylight-saving-time (DST) table, the BMON is then loaded, followed by a BCOM command to start the first job in the batch flow:

```
BCOM $BMCL; START SEGMENT SSPL.AA.
```

Note: The above procedures do not require TMF to be started at this point; TMF will be started during the batch schedule.

An initial MultiBatch Prepare will have been run once the configuration has been built; therefore, no further Prepares are required, unless the configuration changes, e.g. the addition or deletion of jobs, segments, or units.

However, if a Prepare is required during a system coldload, then ensure TMF is running and that the Prepare results are not output to the spooler; it is recommended that results are output to a disk file, based on a <record-length> of 80 and the <file-type> is Entry Sequenced.

MultiBatch Housekeeping

Manual repetitive tasks such as running Ad-Hoc, Daily and Weekly Backups, TMF Online Dumps, Spooler 'clean-up', System Checks (TMF, RDF, Devices, CPUs...), etc. can all be incorporated into a MultiBatch schedule.

Various MultiBatch scheduling facilities can be utilised to automate the running of these jobs, i.e. Event-Timer, Calendar Files, Job Start times, Job or Segment Dependencies.

An example of a MultiBatch configuration for running a Weekly Backup is provided below.

Weekly Backup FBK1 – Step Configuration

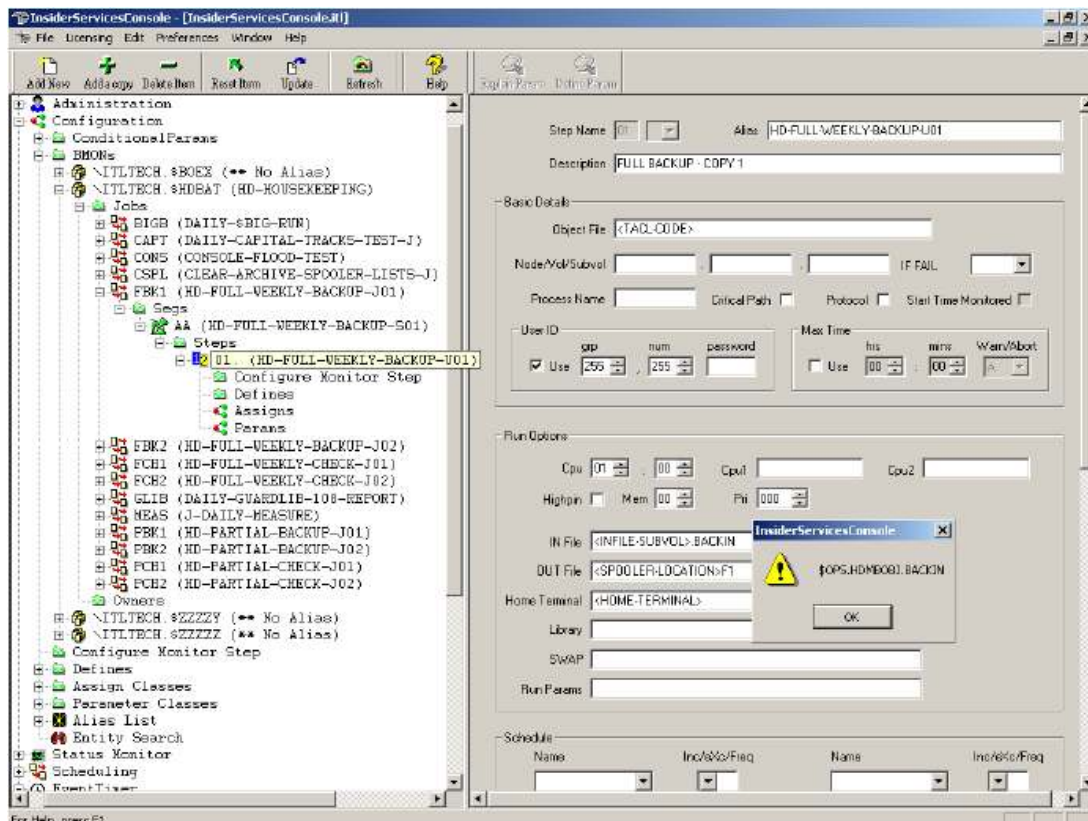
The MultiBatch step for job FBK1 executes a TACL 'INFILE' macro, which utilises various Assigns and Parameters.

The INFILE is a 'common' TACL file and can be used in other backup jobs. Therefore, a MultiBatch Conditional Parameter is used, as identified by the < > symbols.

By using Conditional Parameters, the user only needs to store the value in one area of the database enabling the <parameter> to be used in many steps.

If the underlying 'true' value needs to change, then it only needs to be amended in one area of the MultiBatch database: Conditional Parameters. This saves having to make repetitive changes in the various steps which use this value. See the "MultiBatch White Paper" document for further details on the use of Conditional Parameters.

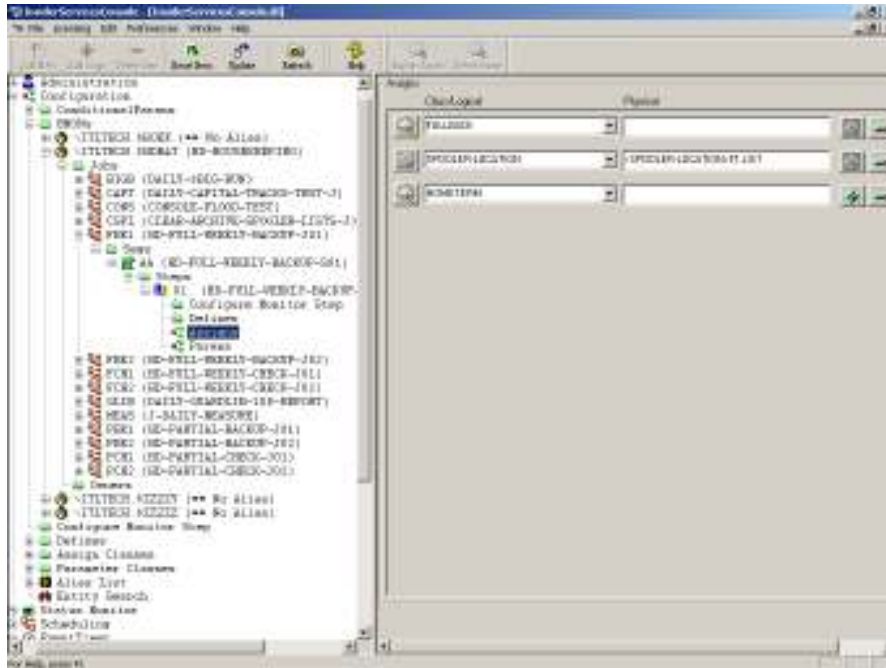
When the 'Explain Param' function is used (magnifying glass), the 'true' value is displayed as: \$OPS.HDMBOBJ.



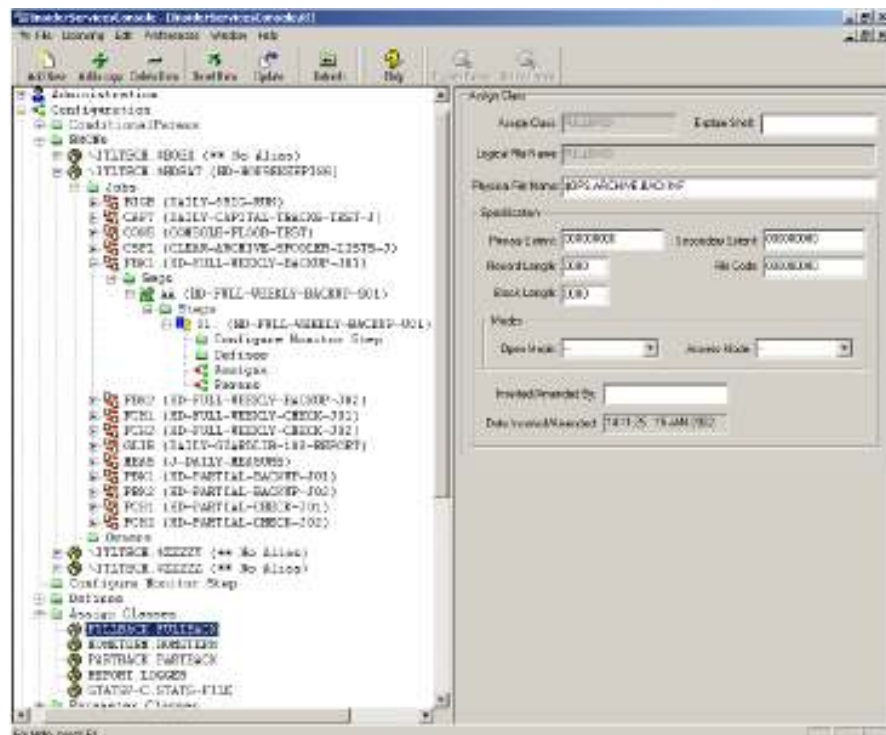
Weekly Backup FBK1 – Assigns

The TACL 'INFILE' macro for unit FBK1.AA.01 utilises Assigns and in the case of FULLBACK and HOMETERM, Assign 'Classes' are used.

This operates in a similar way to Conditional Parameters in that a 'local' value can be stored, allowing the Assigns Class names: FULLBACK and HOMETERM to be used in other steps.

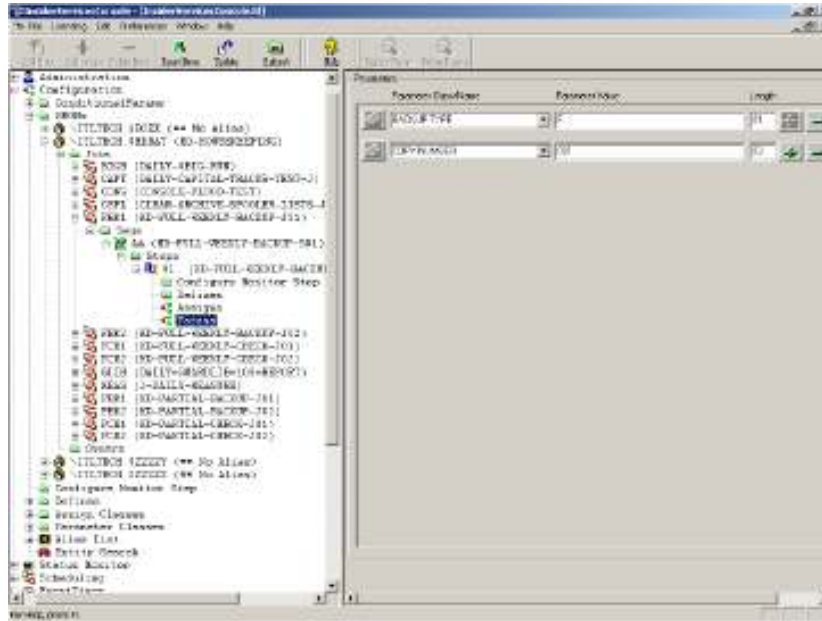


When the 'Explain Param' function is used, the underlying Assign value is then displayed:



Weekly Backup FBK1 – Parameters

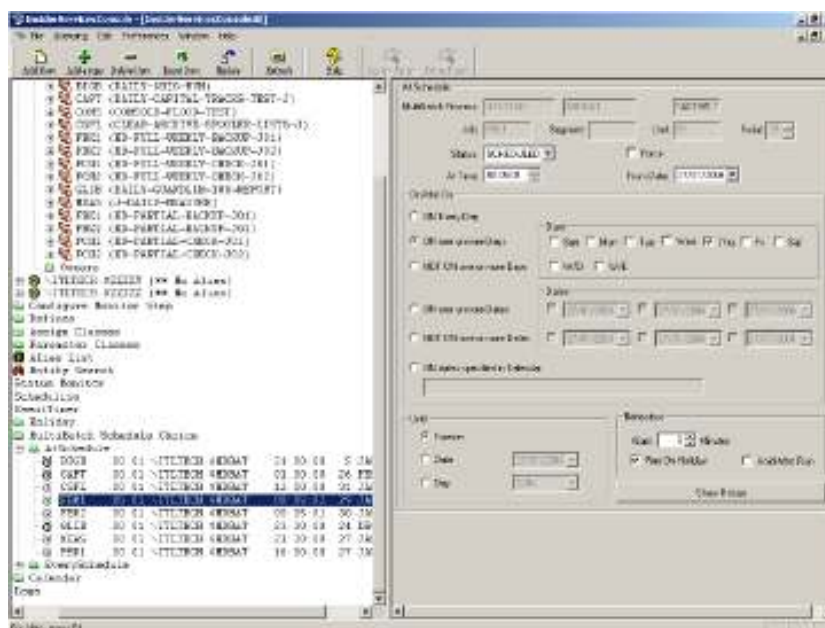
The TACL ‘INFILE’ macro also utilises two parameters: BACKUP-TYPE and COPY-NUMBER with the values of ‘F’ and ‘2’ respectively. Similar to Assign Classes, Parameter ‘Classes’ can also be configured for commonly used parameters, allowing them to be used in other steps. In this example, ‘unique’ values are specified for each parameter.



Weekly Backup FBK1 - Event-Timer Schedule

Displayed below, is an example of how to use MultiBatch Event-Timer to schedule a full weekly backup job.

In this example, job **FBK1** runs “EVERY Thursday at 00:05:01”.



MultiBatch Site-Swap

This section describes how by using MultiBatch, the turnaround time for a site-swap during a Disaster Recovery or Contingency situation can be improved.

As documented in the Coldload section, the *parallelism* of MultiBatch can provide benefits during a site-swap; for example: reduced down-time.

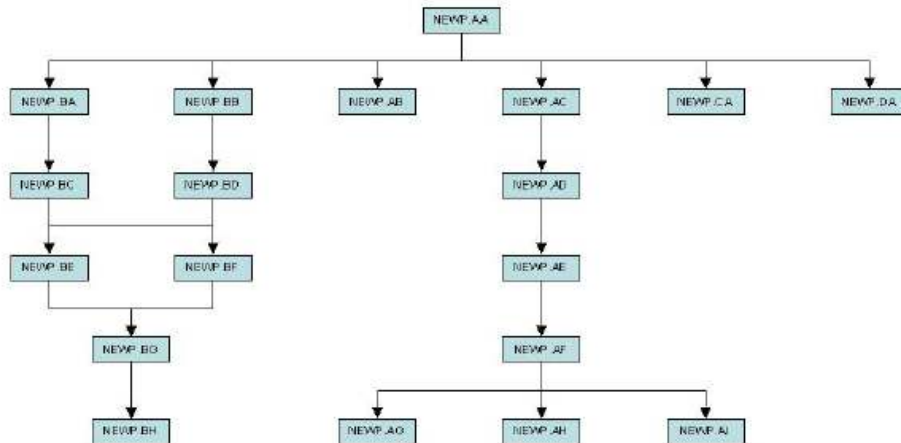
This document provides an example of a batch schedule which executes on the 'new' Production node.

Batch flow

A flowchart example of how MultiBatch can be used in a site-swap situation is provided below, where the batch schedule executes on the new Production node.

A table containing the individual job names and function is provided on the next page.

As with a Coldload MultiBatch schedule, the units can run using different user ids.



SEGMENT ID	FUNCTION
NEWP.AA	Set units in BMON \$NEWP 'ready to run'
NEWP.AB	Down Expand Lines
NEWP.AC	Check RDF Takeover has completed successfully
NEWP.AD	Stop, Start TMF and set Next Audittrail MAT
NEWP.AE	Set Audit flags on database. This segment may not be required, depending on the version of RDF used (or other remote database duplication software used, e.g. 'Goldengate')
NEWP.AF	Check Audit Flags
NEWP.AG, AH, AI...	Run Online Dumps for the various databases
NEWP.BA	Start EMS Alternate Collectors and Distributors
NEWP.BB	Start 'Operations' Viewpoint
NEWP.BC	Start Application Pathway
NEWP.BD	Start Application Pathway
NEWP.BE	Start Application processes
NEWP.BF	Start Application Pathway Servers
NEWP.BG	Start Networks (TCP/IP, X.25, etc.)
NEWP.BH	Start External Networks
NEWP.CA	Down Testing Environments
NEWP.DA	Down Training Environments

Segment Description

The MultiBatch 'site-swap' BMON (\$NEWP) resides on each node, with all units placed in a 'holdoff' state, i.e. skipped. This ensures the jobs will not be executed until required.

When they are required to run, it is achieved by the first MultiBatch unit – NEWP.AA.01 – executing a TACL 'INFILE', to change all units from a 'skipped' state to a 'runon' state (unit is ready to run). Contents of the TACL 'INFILE' contain commands such as:

```
BCOM $NEWP; ALTER STEP NEWP.AB.01 RUNON
```

```
BCOM $NEWP; ALTER STEP NEWP.AC.01 RUNON
```

```
BCOM $NEWP; ALTER STEP NEWP.AD.01 RUNON
```

...

Note: Comprehensive multi-tiered security within MultiBatch ensures jobs are only run as required and by the correct user!

Although all of the units are modified to a 'runon' state, the in-built segment dependencies control the flow and execution of the jobs.

Multiple units can be configured under each segment, e.g. NEWP.AB.01; NEWP.AB.02; NEWP.AB.03 may be configured to down three Expand lines.

This example Disaster Recovery BMON schedule, demonstrates one way of how to utilise MultiBatch in activating the 'new' Production node.

During a site-swap, various takeover modes can be incorporated into the MultiBatch steps (via user supplied TACL scripts) to prompt Operations, e.g. "Are you performing a Contingency [C] or Disaster [D] site-swap?"

Based on the response, certain MultiBatch units therefore may or may not run. For example if 'C' was input, then the units for downing the Expand lines will not be run but instead will be skipped.

MultiBatch Migration Strategy

The control and maintenance of MultiBatch schedules in Production, Backup, Testing and Trialling environments can all be easily maintained via the MultiBatch Migrator utility.

The Migrator utility enables batch schedules to be 'extracted' from a control (or release) database and rolled-out to the various environments, resident in the Testing, Trialling, Production and Backup nodes.

The 'control' database is used to maintain MultiBatch schedules and any changes to a schedule, such as additional jobs, changes to parameters, defines, are performed in this environment. Once changes have been made (ideally via a Change Control system), then they are rolled-out to the MultiBatch Testing or Trialling environments.

Following a successful Test and/or Trial of the new batch schedule, the schedule is again 'extracted' from the control database and rolled-out to the Production and Backup environments. Obviously, any changes required following Testing and Trialling are fed back into the control database.

Conditional Parameters are maintained within the various databases to reflect their 'unique' requirements; therefore, when a BMON is inserted into the Trialling environment, the \TRIAL Conditional Parameter database will contain 'unique' values, e.g. <disk> equates to \$DEV and <P> (process prefix) equates to \$T.

The MultiBatch 'MIGDEFS' file on \TRIAL is also utilised and during the Migrator Insert process, the 'old' node references are replaced with the 'new' node references; for example, the BMON for \CTRL.\$BM01 becomes \TRIAL.BM01 in the Trial database.

An example of 'releasing' BMON \$BM01 into the Trialling environment is provided on the next page.

Batch Schedule

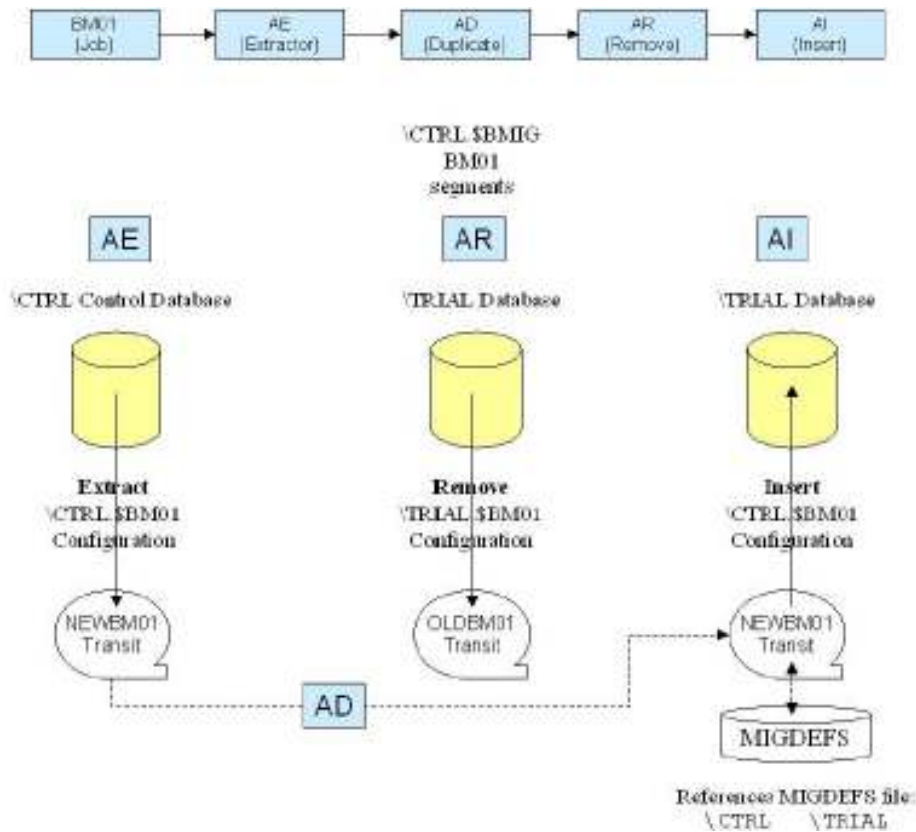
The individual Migrator requirements have been configured into MultiBatch segments, utilising dependencies at segment level.

The underlying units contain simple TACL scripts to execute the various Migrator commands. Based on the user response issued from segment AE, all of the segments will then execute Migrator for the appropriate environment.

BMON	Job	Segment	Description
\$BMIG	BM01	AE	1. Prompt issued: "Which environment are you updating?" 2. Response: Trialling 3. Extract copy of BMON \$BM01 into TRANSIT file: NEWBM01.
		AD	Duplicate MultiBatch Transit NEWBM01 file to the \TRIAL node.
		AR	Remove 'old' copy of \TRIAL.\$BM01 into Transit file: OLDBM01.
		AI	Insert 'new' copy of \CTRL.\$BM01 into the \TRIAL database, using transit file: NEWBM01. (MIGDEFS file will be referenced to change node prefix: \CTRL to \TRIAL).

Batch flow

Segment dependencies are configured. Therefore, if segment AD fails (perhaps due to network security problems), then the remaining batch will stop until the 'AD' issues have been resolved.



MultiBatch Application Processing

MultiBatch can be utilised to run all of your batch applications, e.g.: 'Online-Day', 'Overnight Processing' and 'End-Of-Day'.

Before the application is activated for customer access, manual tasks such as checking TMF,CPU, DISKS, EXPAND, TCP/IP, Audit Flags can all be incorporated into a MultiBatch schedule.

Following the successful completion of these 'system' checks, the MultiBatch schedule then starts the 'Online-Day' application for customer access.

Likewise, the 'Overnight' batch processing, incorporating the running of online dumps, generation of reports, archiving of transactions, transfer of audit logs to archiving, etc., can be built into a MultiBatch schedule.

The structure of the MultiBatch application schedules can be as simple or as complex as the user decides.

It is not possible to demonstrate the complexities of a MultiBatch Application schedule, as each application is unique and ITL clients regularly execute several hundred batch jobs in a single schedule.

Summary

The implementation of MultiBatch parallel processing can significantly reduce the time needed to complete tasks that were previously a serial exercise.

The benefits of parallel processing will therefore be seen in faster system startup times and improved SLAs for your customers.

Amendments to any of the MultiBatch schedules described in this document are easily performed, without the need to maintain lengthy JCL scripts (TACL or NBEXEC). All maintenance is performed via the MultiBatch Graphical User Interface.

The maintenance and release of MultiBatch schedules is easily achieved via means of the 'Migrator' utility, enabling new versions of a batch schedule to be rolled-out to other environments.



Insider Technologies is a UK-based software and services company quality certificated to ISO 9001:2008 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 in Salford Quays (home to MediaCityUK), 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
Website: <http://www.insidertech.co.uk>



ISV/Software Solutions